

# digiKam Developer Documentation

Generated on Sun Jan 5 2025 05:11:14 for digiKam Developer Documentation by Doxygen  
1.9.8

Sun Jan 5 2025 05:11:14



---

<b>1 digiKam project API reference.</b>	<b>1</b>
1.1 Source Code Directories	1
1.2 External Dependencies	6
1.2.1 Dependencies To Checkout All Source Code	6
1.2.2 Dependencies To Process Translations Files (optional)	6
1.2.3 Dependencies To Compile And Link Source Code	6
1.3 Get Source Code	11
1.3.1 Software Components	11
1.4 Development Environment	11
1.5 Cmake Configuration Options	12
1.5.1 Top Level Configuration	12
1.5.2 Core Configuration	12
1.6 Setup Local Compilation and Run-Time	13
1.7 Debug Traces At Run-Time	13
1.7.1 Logging Using an Environment Variable	13
1.7.2 Logging Categories in digiKam	14
1.7.3 Further Reading	15
1.8 Cmake compilation rules	15
1.8.1 Introduction	15
1.8.2 CMake Implementation Details	16
1.8.2.1 Include Directories	16
1.8.2.2 Shared Libraries	16
1.8.2.3 Static Libraries	16
1.8.2.4 Object Libraries	17
1.9 Contribute To The Code	17
1.9.1 Starting With Open-Source	17
1.9.2 Source Code Formatting	17
1.9.2.1 Indentation length	17
1.9.2.2 Tabs vs Spaces	18
1.9.2.3 Line length	18
1.9.2.4 Bracketing	18
1.9.2.5 Positioning of Access modifiers	18
1.9.3 Class, file and Variable names	18
1.9.3.1 Class and filenames	18
1.9.3.2 Protected Member variables	19
1.9.3.3 Non-Member variables	19
1.9.3.4 Private Member variables	19
1.9.4 Comments and Whitespace	19
1.9.5 Header Files	19
1.9.6 Automatic source code formatting	20
1.9.7 General recommendations	20
1.9.8 GDB Backtrace	21

---

1.9.9 Memory Leak	21
1.9.10 Profiling With Cachegrind	21
1.9.11 Unit Testing / Automated Testing	21
1.9.12 Checking For Corrupt Qt Signal Slot Connection	22
1.9.13 Finding Duplicated Code	22
1.9.14 API Documentation Validation, User Documentation Validation, Source Code Checking	22
1.9.15 Usability Issues	22
1.9.16 Generate API Documentation	22
1.9.17 Speed Up The Code-Compile-Test Cycle	23
1.9.18 Working With Branches From Git Repository	23
1.9.19 Sync a Branch With Master From Git Repository	23
<b>2 Namespace Index</b>	<b>25</b>
2.1 Namespace List	25
<b>3 Hierarchical Index</b>	<b>27</b>
3.1 Class Hierarchy	27
<b>4 Class Index</b>	<b>59</b>
4.1 Class List	59
<b>5 Namespace Documentation</b>	<b>89</b>
5.1 Digikam Namespace Reference	89
5.1.1 Detailed Description	123
5.1.2 Typedef Documentation	123
5.1.2.1 ActionJobCollection	123
5.1.2.2 BatchSetList	124
5.1.2.3 BatchToolSettings	124
5.1.2.4 BatchToolsList	124
5.1.2.5 DateRange	124
5.1.2.6 DateRangeList	124
5.1.2.7 DImgLoaderPrms	124
5.1.2.8 DItemsListIsLessThanHandler	124
5.1.2.9 QueuePoolItemsList	124
5.1.3 Enumeration Type Documentation	124
5.1.3.1 DetectorModel	124
5.1.3.2 DetectorNNModel	125
5.1.3.3 FullScreenOptions	125
5.1.3.4 GeoGroupStateEnum	125
5.1.3.5 HistogramRenderingType	126
5.1.3.6 HistogramScale	126
5.1.3.7 HudSide	126
5.1.3.8 MeaningOfDirection	126
5.1.3.9 OperationType	126

5.1.3.10 YoloVersions	127
5.1.4 Function Documentation	127
5.1.4.1 adjustedEnvironmentForApplImage()	127
5.1.4.2 asDateTimeLocal()	127
5.1.4.3 asDateTimeUTC()	127
5.1.4.4 coordinatesToClipboard()	127
5.1.4.5 DatabaseImageMetadataFieldsToMetadataInfoField()	128
5.1.4.6 DatabaseVideoMetadataFieldsToMetadataInfoField()	128
5.1.4.7 defineShortcut()	128
5.1.4.8 DNotificationWrapper()	128
5.1.4.9 GeolfaceHelperParseLatLonString()	128
5.1.4.10 getComponentValue()	129
5.1.4.11 image2Mat()	129
5.1.4.12 image2Mat_shared()	129
5.1.4.13 installQtTranslationFiles()	130
5.1.4.14 isReadableImageFile()	130
5.1.4.15 isRunningInApplImageBundle()	130
5.1.4.16 isRunningOnNativeKDE()	130
5.1.4.17 layoutMargin()	130
5.1.4.18 layoutSpacing()	130
5.1.4.19 loadEcmQtTranslationFiles()	131
5.1.4.20 loadStdQtTranslationFiles()	131
5.1.4.21 macOSBundlePrefix()	131
5.1.4.22 mat2Image()	131
5.1.4.23 mat2Image_shared()	131
5.1.4.24 openOnlineDocumentation()	131
5.1.4.25 operator<<()	132
5.1.4.26 operator"   ()	132
5.1.4.27 qHash()	132
5.1.4.28 QPointSquareDistance()	132
5.1.4.29 s_inlineTranslateString()	132
5.1.4.30 s_rawFileExtensionsdWithDesc()	133
5.1.4.31 s_rawFileExtensionsVersion()	133
5.1.4.32 s_setXmpTagStringFromEntry()	133
5.1.4.33 setExifXmpTagDataVariant()	134
5.1.4.34 setMacOSEnvironment()	134
5.1.4.35 setOpenCLEnvironment()	134
5.1.4.36 setWindowsEnvironment()	134
5.1.4.37 showRawCameraList()	134
5.1.4.38 startOfDay()	134
5.1.4.39 supportedImageMimeTypes()	134
5.1.4.40 toolButtonStyleSheet()	135

---

5.1.4.41 unloadQtTranslationFiles()	135
5.1.5 Variable Documentation	135
5.1.5.1 accessCol	135
5.1.5.2 accessRow	135
5.1.5.3 CR_basis	135
5.1.5.4 faceenum2size	136
5.1.5.5 GeolfaceMinMarkerGroupingRadius	136
5.1.5.6 s_metaEngineMutex	136
5.1.5.7 s_metaEngineSupportBmff	136
5.1.5.8 s_metaEngineWarnOrError	136
5.1.5.9 s_rfc3066ForXMP	136
5.1.5.10 s_stage	137
5.2 Digikam::Matrix Namespace Reference	137
5.2.1 Detailed Description	137
<b>6 Class Documentation</b>	<b>139</b>
6.1 CoreDbWatchAdaptor Class Reference	139
6.2 Digikam::AbstractAlbumModel Class Reference	140
6.2.1 Member Enumeration Documentation	142
6.2.1.1 AlbumDataRole	142
6.2.1.2 RootAlbumBehavior	142
6.2.2 Constructor & Destructor Documentation	143
6.2.2.1 AbstractAlbumModel()	143
6.2.3 Member Function Documentation	143
6.2.3.1 albumCleared()	143
6.2.3.2 albumData()	143
6.2.3.3 albumForIndex()	143
6.2.3.4 albumType()	143
6.2.3.5 allAlbumsCleared()	144
6.2.3.6 columnHeader()	144
6.2.3.7 decorationRoleData()	144
6.2.3.8 dragDropHandler()	144
6.2.3.9 filterAlbum()	144
6.2.3.10 fontRoleData()	144
6.2.3.11 indexForAlbum()	144
6.2.3.12 isFaceTagModel()	145
6.2.3.13 retrieveAlbum()	145
6.2.3.14 rootAlbumAvailable	145
6.2.3.15 rootAlbumBehavior()	145
6.2.3.16 rootAlbumIndex()	145
6.2.3.17 setDragDropHandler()	145
6.2.3.18 setDropIndex()	145

6.2.3.19 setEnableDrag()	146
6.2.3.20 sortRoleData()	146
6.3 Digikam::AbstractAlbumTreeView Class Reference	147
6.3.1 Detailed Description	150
6.3.2 Member Enumeration Documentation	150
6.3.2.1 Flag	150
6.3.3 Constructor & Destructor Documentation	150
6.3.3.1 AbstractAlbumTreeView()	150
6.3.4 Member Function Documentation	150
6.3.4.1 adaptColumnsToContent	150
6.3.4.2 addCustomContextMenuActions()	151
6.3.4.3 contextMenuIcon()	152
6.3.4.4 contextMenuTitle()	152
6.3.4.5 currentAlbumChanged	152
6.3.4.6 doLoadState()	152
6.3.4.7 doSaveState()	153
6.3.4.8 expandEverything	153
6.3.4.9 expandMatches()	153
6.3.4.10 handleCustomContextMenuAction()	153
6.3.4.11 indexVisuallyAt()	154
6.3.4.12 pixmapForDrag()	154
6.3.4.13 scrollToSelectedAlbum	154
6.3.4.14 selectedAlbumsChanged	154
6.3.4.15 setAlbumManagerCurrentAlbum()	154
6.3.4.16 setContextMenuIcon()	154
6.3.4.17 setCurrentAlbums	154
6.3.4.18 setEnableContextMenu()	155
6.3.4.19 setExpandNewCurrentItem()	155
6.3.4.20 setExpandOnSingleClick()	155
6.3.4.21 setSelectAlbumOnClick()	155
6.3.4.22 setSelectOnContextMenu()	156
6.3.4.23 showContextMenuAt()	156
6.3.4.24 slotRootAlbumAvailable	156
6.3.4.25 viewportEvent()	156
6.4 Digikam::AbstractAlbumTreeView::ContextMenuElement Class Reference	157
6.4.1 Detailed Description	157
6.4.2 Member Function Documentation	157
6.4.2.1 addActions()	157
6.5 Digikam::AbstractAlbumTreeView::Private Class Reference	158
6.6 Digikam::AbstractAlbumTreeViewSelectComboBox Class Reference	159
6.6.1 Constructor & Destructor Documentation	161
6.6.1.1 AbstractAlbumTreeViewSelectComboBox()	161

---

6.6.2 Member Function Documentation	162
6.6.2.1 addCheckUncheckContextMenuActions()	162
6.6.2.2 installView()	162
6.6.2.3 sendViewportEventToView()	162
6.6.2.4 setTreeView()	162
6.7 Digikam::AbstractCheckableAlbumModel Class Reference	163
6.7.1 Constructor & Destructor Documentation	168
6.7.1.1 AbstractCheckableAlbumModel()	168
6.7.2 Member Function Documentation	168
6.7.2.1 albumCleared()	168
6.7.2.2 albumData()	168
6.7.2.3 allAlbumsCleared()	168
6.7.2.4 checkStateChanged	169
6.7.2.5 prepareAddExcludeDecoration()	169
6.7.2.6 setAddExcludeTristate()	169
6.7.2.7 setData()	169
6.7.2.8 setRecursive()	169
6.7.2.9 setRootCheckable()	169
6.7.2.10 setTristate()	170
6.8 Digikam::AbstractCheckableAlbumTreeView Class Reference	171
6.8.1 Constructor & Destructor Documentation	174
6.8.1.1 AbstractCheckableAlbumTreeView()	174
6.8.2 Member Function Documentation	175
6.8.2.1 albumModel()	175
6.8.2.2 doLoadState()	175
6.8.2.3 doSaveState()	175
6.8.2.4 isRestoreCheckState()	175
6.8.2.5 middleButtonPressed()	175
6.8.2.6 setCheckOnMiddleClick()	176
6.8.2.7 setRestoreCheckState()	176
6.9 Digikam::AbstractCountingAlbumModel Class Reference	177
6.9.1 Member Function Documentation	180
6.9.1.1 albumCleared()	180
6.9.1.2 albumCount()	180
6.9.1.3 albumData()	181
6.9.1.4 albumForId()	181
6.9.1.5 albumName()	181
6.9.1.6 allAlbumsCleared()	181
6.9.1.7 excludeChildrenCount	181
6.9.1.8 includeChildrenCount	181
6.9.1.9 setCountHash	182
6.9.1.10 setup()	182



6.10 Digikam::AbstractCountingAlbumTreeView Class Reference	183
6.11 Digikam::AbstractDetector Class Reference	186
6.11.1 Member Function Documentation	187
6.11.1.1 prepareForDetection()	187
6.12 Digikam::AbstractItemDragDropHandler Class Reference	187
6.12.1 Member Function Documentation	188
6.12.1.1 accepts()	188
6.12.1.2 acceptsMimeData()	188
6.12.1.3 createMimeData()	188
6.12.1.4 dropEvent()	188
6.12.1.5 mimeTypes()	189
6.13 Digikam::AbstractMarkerTiler Class Reference	190
6.13.1 Member Function Documentation	191
6.13.1.1 bestRepresentativeIndexFromList()	191
6.13.1.2 getTile()	192
6.13.1.3 getTileGroupState()	192
6.13.1.4 getTileRepresentativeMarker()	192
6.13.1.5 indicesEqual()	192
6.13.1.6 onIndicesClicked()	192
6.13.1.7 pixmapFromRepresentativeIndex()	192
6.13.1.8 prepareTiles()	193
6.13.1.9 setActive()	193
6.13.1.10 tilerFlags()	193
6.14 Digikam::AbstractMarkerTiler::ClickInfo Class Reference	193
6.15 Digikam::AbstractMarkerTiler::NonEmptyIterator Class Reference	193
6.16 Digikam::AbstractMarkerTiler::Tile Class Reference	194
6.17 Digikam::AbstractSearchGroupContainer Class Reference	195
6.17.1 Constructor & Destructor Documentation	196
6.17.1.1 AbstractSearchGroupContainer()	196
6.17.2 Member Function Documentation	196
6.17.2.1 addGroupToLayout()	196
6.17.2.2 createSearchGroup()	197
6.18 Digikam::AbstractSpecificAlbumModel Class Reference	197
6.18.1 Member Function Documentation	200
6.18.1.1 columnHeader()	200
6.19 Digikam::AbstractWidgetDelegateOverlay Class Reference	200
6.19.1 Constructor & Destructor Documentation	202
6.19.1.1 AbstractWidgetDelegateOverlay()	202
6.19.2 Member Function Documentation	202
6.19.2.1 checkIndex()	202
6.19.2.2 checkIndexOnEnter()	202
6.19.2.3 createWidget()	202

---

6.19.2.4	hide()	202
6.19.2.5	parentWidget()	203
6.19.2.6	setActive()	203
6.19.2.7	slotEntered	203
6.19.2.8	slotReset	203
6.19.2.9	viewportLeaveEvent()	203
6.19.2.10	widgetEnterEvent()	204
6.19.2.11	widgetEnterNotifyMultiple()	204
6.20	Digikam::ActionCategorizedView Class Reference	205
6.21	Digikam::ActionData Class Reference	207
6.22	Digikam::ActionItemModel Class Reference	208
6.22.1	Member Enumeration Documentation	209
6.22.1.1	MenuCategoryFlag	209
6.22.2	Constructor & Destructor Documentation	210
6.22.2.1	ActionItemModel()	210
6.22.3	Member Function Documentation	210
6.22.3.1	actionForIndex()	210
6.22.3.2	createFilterModel()	210
6.22.3.3	hover	210
6.22.3.4	itemForAction()	210
6.23	Digikam::ActionJob Class Reference	211
6.23.1	Constructor & Destructor Documentation	211
6.23.1.1	ActionJob()	211
6.23.1.2	~ActionJob()	212
6.23.2	Member Function Documentation	212
6.23.2.1	cancel	212
6.23.2.2	signalDone	212
6.23.2.3	signalProgress	212
6.23.2.4	signalStarted	212
6.23.3	Member Data Documentation	212
6.23.3.1	m_cancel	212
6.23.3.2	m_timer	212
6.24	Digikam::ActionSortFilterProxyModel Class Reference	213
6.25	Digikam::ActionTask Class Reference	215
6.26	Digikam::ActionThread Class Reference	217
6.26.1	Member Function Documentation	218
6.26.1.1	signalCancelActionTask	218
6.26.1.2	signalFinished	218
6.26.1.3	signalQueueProcessed	218
6.26.1.4	signalStarting	219
6.27	Digikam::ActionThreadBase Class Reference	219
6.27.1	Member Function Documentation	220

6.27.1.1	appendJobs()	220
6.27.1.2	cancel()	220
6.27.1.3	isEmpty()	220
6.27.1.4	maximumNumberOfThreads()	220
6.27.1.5	pendingCount()	220
6.27.1.6	run()	220
6.27.1.7	setDefaultMaximumNumberOfThreads()	221
6.27.1.8	setMaximumNumberOfThreads()	221
6.28	Digikam::ActionVersionsOverlay Class Reference	222
6.28.1	Member Function Documentation	225
6.28.1.1	checkIndex()	225
6.28.1.2	createButton()	225
6.28.1.3	setActive()	225
6.28.1.4	updateButton()	225
6.29	Digikam::AddBookmarkDialog Class Reference	226
6.30	Digikam::AddBookmarkProxyModel Class Reference	226
6.30.1	Detailed Description	227
6.31	Digikam::AddTagsComboBox Class Reference	228
6.31.1	Member Function Documentation	231
6.31.1.1	currentTaggingAction()	231
6.31.1.2	setAlbumModels()	231
6.31.1.3	setCurrentTag()	232
6.31.1.4	setParentTag	232
6.31.1.5	taggingActionActivated	232
6.31.1.6	taggingActionSelected	232
6.32	Digikam::AddTagsLineEdit Class Reference	233
6.32.1	Member Function Documentation	234
6.32.1.1	setAlbumModels()	234
6.32.1.2	setCurrentTag()	234
6.32.1.3	setFilterModel()	234
6.32.1.4	setParentTag	235
6.32.1.5	setSupportingTagModel()	235
6.32.1.6	setTagTreeView()	235
6.32.1.7	slotReturnPressed	235
6.32.1.8	taggingActionActivated	235
6.32.1.9	taggingActionSelected	235
6.33	Digikam::AdvancedMetadataTab Class Reference	236
6.33.1	Constructor & Destructor Documentation	236
6.33.1.1	AdvancedMetadataTab()	236
6.34	Digikam::AdvancedRenameDialog Class Reference	237
6.35	Digikam::AdvancedRenameInput Class Reference	238
6.36	Digikam::AdvancedRenameLineEdit Class Reference	239

---

6.37 Digikam::AdvancedRenameListItem Class Reference . . . . .	240
6.38 Digikam::AdvancedRenameManager Class Reference . . . . .	241
6.39 Digikam::AdvancedRenameProcessDialog Class Reference . . . . .	243
6.40 Digikam::AdvancedRenameWidget Class Reference . . . . .	245
6.40.1 Member Function Documentation . . . . .	246
6.40.1.1 clear() . . . . .	246
6.40.1.2 clearParseString() . . . . .	246
6.40.1.3 focusLineEdit() . . . . .	246
6.40.1.4 highlightLineEdit() [1/2] . . . . .	246
6.40.1.5 highlightLineEdit() [2/2] . . . . .	246
6.40.1.6 parse() . . . . .	246
6.40.1.7 parser() . . . . .	247
6.40.1.8 parseString() . . . . .	247
6.40.1.9 setControlWidgets() . . . . .	247
6.40.1.10 setLayoutStyle() . . . . .	247
6.40.1.11 setParser() . . . . .	248
6.40.1.12 setParseString() . . . . .	248
6.41 Digikam::AdvancedSettings Class Reference . . . . .	249
6.42 Digikam::AestheticDetector Class Reference . . . . .	250
6.42.1 Member Function Documentation . . . . .	251
6.42.1.1 detect() . . . . .	251
6.43 Digikam::Akonadiface Class Reference . . . . .	251
6.44 Digikam::Album Class Reference . . . . .	252
6.44.1 Detailed Description . . . . .	253
6.44.2 Member Enumeration Documentation . . . . .	254
6.44.2.1 Type . . . . .	254
6.44.3 Constructor & Destructor Documentation . . . . .	255
6.44.3.1 Album() . . . . .	255
6.44.3.2 ~Album() . . . . .	255
6.44.4 Member Function Documentation . . . . .	255
6.44.4.1 childAlbumIds() . . . . .	255
6.44.4.2 childAlbums() . . . . .	256
6.44.4.3 childAtRow() . . . . .	256
6.44.4.4 childCount() . . . . .	256
6.44.4.5 clear() . . . . .	256
6.44.4.6 databaseUrl() . . . . .	256
6.44.4.7 extraData() . . . . .	256
6.44.4.8 firstChild() . . . . .	257
6.44.4.9 globalID() [1/2] . . . . .	257
6.44.4.10 globalID() [2/2] . . . . .	257
6.44.4.11 id() . . . . .	258
6.44.4.12 isAncestorOf() . . . . .	258

6.44.4.13 isRoot()	258
6.44.4.14 isTrashAlbum()	259
6.44.4.15 isUsedByLabelsTree()	259
6.44.4.16 lastChild()	259
6.44.4.17 next()	259
6.44.4.18 parent()	259
6.44.4.19 prepareForDeletion()	260
6.44.4.20 prev()	260
6.44.4.21 removeExtraData()	260
6.44.4.22 rowFromAlbum()	260
6.44.4.23 setExtraData()	261
6.44.4.24 setUsedByLabelsTree()	261
6.44.4.25 title()	261
6.44.4.26 type()	262
6.45 Digikam::AlbumChangeset Class Reference	262
6.46 Digikam::AlbumCopyMoveHint Class Reference	262
6.46.1 Constructor & Destructor Documentation	263
6.46.1.1 AlbumCopyMoveHint()	263
6.47 Digikam::AlbumCustomizer Class Reference	263
6.48 Digikam::AlbumDragDropHandler Class Reference	264
6.48.1 Member Function Documentation	265
6.48.1.1 accepts()	265
6.48.1.2 createMimeData()	265
6.48.1.3 dropEvent()	265
6.48.1.4 mimeTypes()	266
6.49 Digikam::AlbumFilterModel Class Reference	267
6.49.1 Member Enumeration Documentation	269
6.49.1.1 FilterBehavior	269
6.49.1.2 MatchResult	269
6.49.2 Member Function Documentation	269
6.49.2.1 compareByOrder()	269
6.49.2.2 compareValue()	270
6.49.2.3 hasSearchResult	270
6.49.2.4 isFiltering()	270
6.49.2.5 lessThan()	270
6.49.2.6 matches()	270
6.49.2.7 matchResult() [1/2]	271
6.49.2.8 matchResult() [2/2]	271
6.49.2.9 searchTextSettings()	271
6.49.2.10 searchTextSettingsAboutToChange	271
6.49.2.11 searchTextSettingsChanged	272
6.49.2.12 setFilterBehavior()	272

---

6.49.2.13 setSearchTextSettings . . . . .	272
6.49.2.14 setSourceAlbumModel() . . . . .	272
6.49.2.15 setSourceFilterModel() . . . . .	272
6.49.2.16 setSourceModel() . . . . .	273
6.49.2.17 signalFilterChanged . . . . .	273
6.49.2.18 updateFilter() . . . . .	273
6.50 Digikam::AlbumFolderViewSideBarWidget Class Reference . . . . .	274
6.50.1 Member Function Documentation . . . . .	276
6.50.1.1 applySettings() . . . . .	276
6.50.1.2 changeAlbumFromHistory() . . . . .	276
6.50.1.3 doLoadState() . . . . .	276
6.50.1.4 doSaveState() . . . . .	276
6.50.1.5 getCaption() . . . . .	276
6.50.1.6 getIcon() . . . . .	277
6.50.1.7 setActive() . . . . .	277
6.51 Digikam::AlbumHistory Class Reference . . . . .	278
6.51.1 Detailed Description . . . . .	279
6.51.2 Member Function Documentation . . . . .	279
6.51.2.1 addAlbums() . . . . .	279
6.52 Digikam::AlbumInfo Class Reference . . . . .	280
6.52.1 Detailed Description . . . . .	280
6.52.2 Member Function Documentation . . . . .	280
6.52.2.1 operator<() . . . . .	280
6.53 Digikam::AlbumIterator Class Reference . . . . .	280
6.53.1 Detailed Description . . . . .	281
6.54 Digikam::AlbumLabelsSearchHandler Class Reference . . . . .	281
6.54.1 Member Function Documentation . . . . .	282
6.54.1.1 albumForSelectedItems() . . . . .	282
6.54.1.2 generatedName() . . . . .	282
6.54.1.3 imagesUrls() . . . . .	282
6.54.1.4 isRestoringSelectionFromHistory() . . . . .	282
6.54.1.5 restoreSelectionFromHistory() . . . . .	282
6.55 Digikam::AlbumManager Class Reference . . . . .	283
6.55.1 Detailed Description . . . . .	286
6.55.2 Member Function Documentation . . . . .	286
6.55.2.1 albumTitles() . . . . .	286
6.55.2.2 allDAAlbums() . . . . .	287
6.55.2.3 allPAAlbums() . . . . .	287
6.55.2.4 allSAAlbums() . . . . .	287
6.55.2.5 allTAAlbums() . . . . .	287
6.55.2.6 changeDatabase() . . . . .	287
6.55.2.7 checkDatabaseDirsAfterFirstRun() . . . . .	287

---

6.55.2.8 cleanUp()	288
6.55.2.9 clearCurrentAlbums()	288
6.55.2.10 createPALbum() [1/3]	288
6.55.2.11 createPALbum() [2/3]	288
6.55.2.12 createPALbum() [3/3]	288
6.55.2.13 createSAlbum()	289
6.55.2.14 createTAlbum()	290
6.55.2.15 currentAlbums()	290
6.55.2.16 currentPALbum()	290
6.55.2.17 currentTAlbums()	291
6.55.2.18 databaseEqual()	291
6.55.2.19 deleteSAlbum()	291
6.55.2.20 deleteTAlbum()	291
6.55.2.21 findAlbum() [1/2]	292
6.55.2.22 findAlbum() [2/2]	292
6.55.2.23 findDAlbum()	292
6.55.2.24 findOrCreateTAlbums()	293
6.55.2.25 findPALbum() [1/2]	293
6.55.2.26 findPALbum() [2/2]	294
6.55.2.27 findSAlbum() [1/2]	294
6.55.2.28 findSAlbum() [2/2]	294
6.55.2.29 findSAlbumsBySearchType()	294
6.55.2.30 findTagsWithProperty()	296
6.55.2.31 findTAlbum() [1/2]	296
6.55.2.32 findTAlbum() [2/2]	296
6.55.2.33 getDAlbumsCount()	296
6.55.2.34 getFaceCount()	297
6.55.2.35 getItemFromAlbum()	297
6.55.2.36 getPAlbumsCount()	297
6.55.2.37 getRecentlyAssignedTags()	297
6.55.2.38 getTAlbumsCount()	298
6.55.2.39 getUnconfirmedFaceCount()	298
6.55.2.40 instance()	298
6.55.2.41 isMovingAlbum()	298
6.55.2.42 mergeTAlbum()	298
6.55.2.43 moveTAlbum()	299
6.55.2.44 prepareItemCounts()	299
6.55.2.45 refresh()	299
6.55.2.46 renamePALbum()	300
6.55.2.47 renameTAlbum()	300
6.55.2.48 setCurrentAlbums()	300
6.55.2.49 setDatabase()	301

---

6.55.2.50	signalAlbumAboutToBeAdded	301
6.55.2.51	signalAlbumAboutToBeDeleted	301
6.55.2.52	signalAlbumAboutToBeMoved	301
6.55.2.53	signalAlbumAdded	301
6.55.2.54	signalAlbumDeleted	301
6.55.2.55	signalAlbumHasBeenDeleted	302
6.55.2.56	signalAlbumMoved	302
6.55.2.57	signalShowOnlyAvailableAlbumsChanged	302
6.55.2.58	startScan()	302
6.55.2.59	subTags()	302
6.55.2.60	tagNames() [1/2]	302
6.55.2.61	tagNames() [2/2]	303
6.55.2.62	tagPaths() [1/2]	303
6.55.2.63	tagPaths() [2/2]	303
6.55.2.64	updatePAlbumIcon()	304
6.55.2.65	updateSAlbum()	304
6.55.2.66	updateTAlbumIcon()	304
6.56	Digikam::AlbumManager::Private Class Reference	305
6.56.1	Member Data Documentation	306
6.56.1.1	currentAlbums	306
6.57	Digikam::AlbumManagerCreator Class Reference	306
6.58	Digikam::AlbumModel Class Reference	307
6.58.1	Constructor & Destructor Documentation	312
6.58.1.1	AlbumModel()	312
6.58.2	Member Function Documentation	312
6.58.2.1	albumData()	312
6.58.2.2	albumForId()	312
6.58.2.3	decorationRoleData()	312
6.59	Digikam::AlbumModelDragDropHandler Class Reference	313
6.59.1	Member Function Documentation	314
6.59.1.1	accepts()	314
6.59.1.2	acceptsMimeData()	314
6.59.1.3	createMimeData()	314
6.59.1.4	dropEvent()	314
6.59.1.5	mimeTypes()	314
6.60	Digikam::AlbumModificationHelper Class Reference	315
6.60.1	Detailed Description	316
6.60.2	Constructor & Destructor Documentation	316
6.60.2.1	AlbumModificationHelper()	316
6.60.2.2	~AlbumModificationHelper()	316
6.60.3	Member Function Documentation	316
6.60.3.1	bindAlbum()	316



6.60.3.2 boundAlbum()	317
6.60.3.3 slotAlbumDelete	317
6.60.3.4 slotAlbumEdit	317
6.60.3.5 slotAlbumNew	317
6.60.3.6 slotAlbumRename	318
6.61 Digikam::AlbumParser Class Reference	319
6.61.1 Constructor & Destructor Documentation	321
6.61.1.1 AlbumParser() [1/2]	321
6.61.1.2 AlbumParser() [2/2]	321
6.62 Digikam::AlbumPointer< T > Class Template Reference	321
6.62.1 Detailed Description	322
6.63 Digikam::AlbumPointerList< T > Class Template Reference	322
6.64 Digikam::AlbumPropsEdit Class Reference	323
6.65 Digikam::AlbumRootChangeset Class Reference	324
6.66 Digikam::AlbumRootInfo Class Reference	324
6.67 Digikam::AlbumRootLocation Class Reference	325
6.68 Digikam::AlbumsDBJobInfo Class Reference	327
6.69 Digikam::AlbumsDBJobsThread Class Reference	328
6.69.1 Member Function Documentation	329
6.69.1.1 albumsListing()	329
6.70 Digikam::AlbumSelectComboBox Class Reference	330
6.70.1 Member Function Documentation	332
6.70.1.1 filterModel()	332
6.70.1.2 installView()	332
6.70.1.3 model()	333
6.70.1.4 setAllSelectedText()	333
6.70.1.5 setCheckable()	333
6.70.1.6 setCloseOnActivate()	333
6.70.1.7 setDefaultAlbumModel()	333
6.70.1.8 setNoSelectionText()	333
6.70.1.9 setRecursive()	333
6.70.1.10 setShowCheckStateSummary()	334
6.70.1.11 updateText	334
6.71 Digikam::AlbumSelectDialog Class Reference	334
6.72 Digikam::AlbumSelectionTreeView Class Reference	335
6.72.1 Detailed Description	339
6.72.2 Member Function Documentation	339
6.72.2.1 setEnableToolTips()	339
6.72.2.2 signalFindDuplicates	339
6.73 Digikam::AlbumSelectors Class Reference	340
6.73.1 Constructor & Destructor Documentation	341
6.73.1.1 AlbumSelectors()	341

---

6.73.2 Member Function Documentation	341
6.73.2.1 loadState	341
6.73.2.2 resetPAlbumSelection()	341
6.73.2.3 resetSelection()	342
6.73.2.4 resetTAlbumSelection()	342
6.73.2.5 saveState	342
6.73.2.6 selectedAlbumIds()	342
6.73.2.7 selectedAlbums()	342
6.73.2.8 selectedAlbumsAndTags()	342
6.73.2.9 selectedTagIds()	342
6.73.2.10 selectedTags()	342
6.73.2.11 setAlbumSelected()	343
6.73.2.12 setTagSelected()	343
6.73.2.13 setTypeSelection()	343
6.73.2.14 typeSelection()	343
6.73.2.15 wholeAlbumsChecked()	343
6.73.2.16 wholeTagsChecked()	343
6.74 Digikam::AlbumSelectTabs Class Reference	344
6.75 Digikam::AlbumSelectTreeView Class Reference	345
6.75.1 Detailed Description	349
6.75.2 Constructor & Destructor Documentation	349
6.75.2.1 AlbumSelectTreeView()	349
6.75.2.2 ~AlbumSelectTreeView()	349
6.75.3 Member Function Documentation	349
6.75.3.1 addCustomContextMenuActions()	349
6.75.3.2 handleCustomContextMenuAction()	350
6.75.3.3 slotNewAlbum	350
6.76 Digikam::AlbumSelectWidget Class Reference	351
6.77 Digikam::AlbumShortInfo Class Reference	351
6.78 Digikam::AlbumSimplified Class Reference	352
6.78.1 Detailed Description	352
6.79 Digikam::AlbumsJob Class Reference	353
6.80 Digikam::AlbumThumbnailLoader Class Reference	355
6.80.1 Member Enumeration Documentation	356
6.80.1.1 RelativeSize	356
6.80.2 Member Function Documentation	357
6.80.2.1 getAlbumThumbnail()	357
6.80.2.2 getAlbumThumbnailDirectly()	357
6.80.2.3 getFaceThumbnailDirectly()	357
6.80.2.4 getStandardTagIcon()	357
6.80.2.5 getTagThumbnail()	357
6.80.2.6 getTagThumbnailDirectly()	358

6.80.2.7 instance()	358
6.80.2.8 setThumbnailSize()	358
6.80.2.9 signalDispatchThumbnailInternal	358
6.80.2.10 signalFailed	358
6.80.2.11 signalReloadThumbnails	358
6.80.2.12 signalThumbnail	359
6.80.2.13 thumbnailSize()	359
6.81 Digikam::AlbumTreeView Class Reference	360
6.82 Digikam::AlbumTreeViewDelegate Class Reference	364
6.83 Digikam::AlbumTreeViewSelectComboBox Class Reference	365
6.84 Digikam::AlbumWatch Class Reference	368
6.85 Digikam::AltLangStrEdit Class Reference	369
6.85.1 Constructor & Destructor Documentation	371
6.85.1.1 AltLangStrEdit()	371
6.85.2 Member Function Documentation	371
6.85.2.1 addCurrent()	371
6.85.2.2 allLanguagesRFC3066()	371
6.85.2.3 languageNameRFC3066()	371
6.85.2.4 reset()	371
6.85.2.5 setLinesVisible()	371
6.85.2.6 setTitle()	372
6.85.2.7 setTitleWidget()	372
6.85.2.8 signalModified	372
6.85.2.9 signalSelectionChanged	372
6.85.2.10 signalValueAdded	372
6.85.2.11 signalValueDeleted	372
6.85.2.12 slotEnabledInternalWidgets	373
6.85.2.13 titleWidget()	373
6.86 Digikam::AltLangStrEdit::Private Class Reference	373
6.87 Digikam::AnimatedClearButton Class Reference	374
6.87.1 Member Function Documentation	375
6.87.1.1 setShallBeShown()	375
6.87.1.2 stayVisibleWhenAnimatedOut()	375
6.88 Digikam::AnimatedVisibility Class Reference	376
6.88.1 Constructor & Destructor Documentation	377
6.88.1.1 AnimatedVisibility()	377
6.89 Digikam::AntiVignettingContainer Class Reference	377
6.90 Digikam::AntiVignettingFilter Class Reference	378
6.90.1 Member Function Documentation	381
6.90.1.1 filterAction()	381
6.90.1.2 filterIdentifier()	381
6.90.1.3 readParameters()	381

---

6.91 Digikam::AntiVignettingSettings Class Reference	381
6.92 Digikam::ApplicationSettings Class Reference	382
6.92.1 Member Enumeration Documentation	388
6.92.1.1 StringComparisonType	388
6.92.2 Member Function Documentation	388
6.92.2.1 askGroupingOperateOnAll()	388
6.92.2.2 getGroupingOperateOnAll()	388
6.92.2.3 getIconShowOverlays()	389
6.92.2.4 getStringComparisonType()	389
6.92.2.5 operationTypeExplanation()	389
6.92.2.6 operationTypeTitle()	389
6.92.2.7 readMsgBoxShouldBeShown()	390
6.92.2.8 saveMsgBoxShouldBeShown()	390
6.92.2.9 setGroupingOperateOnAll()	390
6.92.2.10 setIconShowOverlays()	390
6.92.2.11 setImageSorting()	391
6.92.2.12 setStringComparisonType()	391
6.93 Digikam::ApplicationSettings::Private Class Reference	391
6.94 Digikam::AssignedBatchTools Class Reference	396
6.94.1 Detailed Description	397
6.95 Digikam::AssignedListView Class Reference	397
6.96 Digikam::AssignedListViewItem Class Reference	398
6.97 Digikam::AssignNameOverlay Class Reference	400
6.97.1 Member Function Documentation	403
6.97.1.1 checkIndex()	403
6.97.1.2 createWidget()	403
6.97.1.3 setActive()	403
6.97.1.4 setFocusOnWidget()	403
6.97.1.5 showOnIndex()	403
6.97.1.6 updateFace()	403
6.97.1.7 viewportLeaveEvent()	404
6.97.1.8 visualChange()	404
6.97.1.9 widgetEnterEvent()	404
6.97.1.10 widgetLeaveEvent()	404
6.98 Digikam::AssignNameWidget Class Reference	405
6.98.1 Constructor & Destructor Documentation	407
6.98.1.1 AssignNameWidget()	407
6.98.2 Member Function Documentation	407
6.98.2.1 assigned	407
6.98.2.2 ignoredClicked	407
6.98.2.3 labelClicked	408
6.98.2.4 rejected	408

6.98.2.5 selected	408
6.98.2.6 setAlbumModels()	408
6.98.2.7 setCurrentTag	408
6.98.2.8 setMode()	408
6.98.2.9 setParentTag	409
6.98.2.10 setUserData	409
6.99 Digikam::AssignNameWidget::Private Class Reference	409
6.99.1 Member Function Documentation	410
6.99.1.1 updateRejectButton()	410
6.100 Digikam::AssignNameWidgetStates Class Reference	411
6.101 Digikam::AudPlayerWdg Class Reference	414
6.102 Digikam::AutoCrop Class Reference	415
6.102.1 Constructor & Destructor Documentation	418
6.102.1.1 AutoCrop()	418
6.102.2 Member Function Documentation	418
6.102.2.1 autoInnerCrop()	418
6.102.2.2 startAnalyse()	418
6.103 Digikam::AutoExpoFilter Class Reference	419
6.103.1 Member Function Documentation	422
6.103.1.1 filterAction()	422
6.103.1.2 filterIdentifier()	422
6.103.1.3 readParameters()	422
6.104 Digikam::AutoLevelsFilter Class Reference	423
6.104.1 Member Function Documentation	426
6.104.1.1 filterAction()	426
6.104.1.2 filterIdentifier()	426
6.104.1.3 readParameters()	426
6.105 Digikam::AutoTagsAssign Class Reference	426
6.105.1 Member Function Documentation	427
6.105.1.1 generateTagsList()	427
6.106 Digikam::AutotagsAssignment Class Reference	428
6.106.1 Member Enumeration Documentation	430
6.106.1.1 AutotagsAssignmentScanMode	430
6.106.2 Constructor & Destructor Documentation	431
6.106.2.1 AutotagsAssignment()	431
6.106.3 Member Function Documentation	431
6.106.3.1 setUseMultiCoreCPU()	431
6.107 Digikam::AutotagsAssignmentTask Class Reference	432
6.108 Digikam::BackendGeonamesRG Class Reference	433
6.108.1 Constructor & Destructor Documentation	435
6.108.1.1 BackendGeonamesRG()	435
6.108.1.2 ~BackendGeonamesRG()	435

---

6.108.2 Member Function Documentation	435
6.108.2.1 backendName()	435
6.108.2.2 callRGBBackend()	435
6.108.2.3 cancelRequests()	436
6.108.2.4 getErrorMessage()	436
6.108.2.5 makeQMapFromXML()	436
6.109 Digikam::BackendGeonamesUSRG Class Reference	436
6.109.1 Constructor & Destructor Documentation	438
6.109.1.1 BackendGeonamesUSRG()	438
6.109.1.2 ~BackendGeonamesUSRG()	438
6.109.2 Member Function Documentation	438
6.109.2.1 backendName()	438
6.109.2.2 callRGBBackend()	438
6.109.2.3 cancelRequests()	439
6.109.2.4 getErrorMessage()	439
6.109.2.5 makeQMapFromXML()	439
6.110 Digikam::BackendGoogleMaps Class Reference	440
6.110.1 Constructor & Destructor Documentation	442
6.110.1.1 ~BackendGoogleMaps()	442
6.110.2 Member Function Documentation	442
6.110.2.1 addActionToConfigurationMenu()	442
6.110.2.2 backendHumanName()	443
6.110.2.3 backendName()	443
6.110.2.4 geoCoordinates()	443
6.110.2.5 getCenter()	443
6.110.2.6 getMarkerModelLevel()	443
6.110.2.7 getNormalizedBounds()	443
6.110.2.8 getZoom()	443
6.110.2.9 isReady()	444
6.110.2.10 mapSize()	444
6.110.2.11 mapWidget()	444
6.110.2.12 mapWidgetDocked()	444
6.110.2.13 mouseModeChanged()	444
6.110.2.14 readSettingsFromGroup()	444
6.110.2.15 regionSelectionChanged()	444
6.110.2.16 releaseWidget()	445
6.110.2.17 reload()	445
6.110.2.18 saveSettingsToGroup()	445
6.110.2.19 screenCoordinates()	445
6.110.2.20 setActive()	445
6.110.2.21 setCenter()	445
6.110.2.22 setMarkerPixmap()	445

6.110.2.23	setZoom()	446
6.110.2.24	updateActionAvailability()	446
6.110.2.25	updateClusters()	446
6.110.2.26	updateMarkers()	446
6.110.2.27	zoomIn()	446
6.110.2.28	zoomOut()	446
6.111	Digikam::BackendMarble Class Reference	447
6.111.1	Constructor & Destructor Documentation	450
6.111.1.1	~BackendMarble()	450
6.111.2	Member Function Documentation	450
6.111.2.1	addActionToConfigurationMenu()	450
6.111.2.2	applyCacheToWidget()	450
6.111.2.3	backendHumanName()	450
6.111.2.4	backendName()	450
6.111.2.5	centerOn()	450
6.111.2.6	eventFilter()	450
6.111.2.7	geoCoordinates()	451
6.111.2.8	GeoPainter_drawPixmapAtCoordinates()	451
6.111.2.9	getCenter()	451
6.111.2.10	getMarkerModelLevel()	451
6.111.2.11	getNormalizedBounds()	451
6.111.2.12	getProjection()	451
6.111.2.13	getZoom()	452
6.111.2.14	isReady()	452
6.111.2.15	mapSize()	452
6.111.2.16	mapWidget()	452
6.111.2.17	mapWidgetDocked()	452
6.111.2.18	marbleCustomPaint()	452
6.111.2.19	mouseModeChanged()	452
6.111.2.20	readSettingsFromGroup()	452
6.111.2.21	regionSelectionChanged()	453
6.111.2.22	releaseWidget()	453
6.111.2.23	reload()	453
6.111.2.24	saveSettingsToGroup()	453
6.111.2.25	screenCoordinates()	453
6.111.2.26	setActive()	453
6.111.2.27	setCenter()	453
6.111.2.28	setZoom()	454
6.111.2.29	slotScheduleUpdate	454
6.111.2.30	updateActionAvailability()	454
6.111.2.31	updateClusters()	454
6.111.2.32	updateMarkers()	454

---

6.111.2.33 zoomIn()	454
6.111.2.34 zoomOut()	454
6.112 Digikam::BackendMarbleLayer Class Reference	455
6.113 Digikam::BackendOsmRG Class Reference	455
6.113.1 Constructor & Destructor Documentation	457
6.113.1.1 BackendOsmRG()	457
6.113.1.2 ~BackendOsmRG()	457
6.113.2 Member Function Documentation	457
6.113.2.1 backendName()	457
6.113.2.2 callRGBBackend()	457
6.113.2.3 cancelRequests()	458
6.113.2.4 getErrorMessage()	458
6.113.2.5 makeQMapFromXML()	458
6.114 Digikam::BalooInfo Class Reference	458
6.115 Digikam::BalooWrap Class Reference	459
6.115.1 Detailed Description	460
6.115.2 Member Function Documentation	460
6.115.2.1 getSemanticInfo()	460
6.115.2.2 setSemanticInfo()	460
6.116 Digikam::BasicDImgFilterGenerator< T > Class Template Reference	461
6.116.1 Constructor & Destructor Documentation	462
6.116.1.1 BasicDImgFilterGenerator()	462
6.116.2 Member Function Documentation	462
6.116.2.1 createFilter()	462
6.116.2.2 displayableName()	462
6.116.2.3 supportedFilters()	462
6.116.2.4 supportedVersions()	462
6.117 Digikam::BatchTool Class Reference	463
6.117.1 Member Enumeration Documentation	465
6.117.1.1 BatchToolGroup	465
6.117.2 Constructor & Destructor Documentation	466
6.117.2.1 BatchTool()	466
6.117.3 Member Function Documentation	466
6.117.3.1 apply()	466
6.117.3.2 applyFilter()	466
6.117.3.3 cancel()	466
6.117.3.4 clone()	466
6.117.3.5 defaultSettings()	466
6.117.3.6 deleteSettingsWidget()	467
6.117.3.7 errorDescription()	467
6.117.3.8 getNeedResetExifOrientation()	467
6.117.3.9 getResetExifOrientationAllowed()	467



6.117.3.10 image()	467
6.117.3.11 ioFileSettings()	467
6.117.3.12 isCancelled()	467
6.117.3.13 isRawFile()	467
6.117.3.14 loadToDImg()	468
6.117.3.15 outputSuffix()	468
6.117.3.16 rawDecodingSettings()	468
6.117.3.17 registerSettingsWidget()	468
6.117.3.18 savefromDImg()	468
6.117.3.19 setBranchHistory()	468
6.117.3.20 setDRawDecoderSettings()	469
6.117.3.21 setErrorDescription()	469
6.117.3.22 setImageData()	469
6.117.3.23 setInputUrl()	469
6.117.3.24 setIOFileSettings()	469
6.117.3.25 setItemInfo()	469
6.117.3.26 setLastChainedTool()	469
6.117.3.27 setNeedResetExifOrientation()	470
6.117.3.28 setOutputUrl()	470
6.117.3.29 setOutputUrlFromInputUrl()	470
6.117.3.30 setRawLoadingRules()	470
6.117.3.31 setResetExifOrientationAllowed()	470
6.117.3.32 setSaveAsNewVersion()	470
6.117.3.33 setSettings()	470
6.117.3.34 settingsWidget()	471
6.117.3.35 setToolDescription()	471
6.117.3.36 setToolIconName()	471
6.117.3.37 setToolTitle()	471
6.117.3.38 setWorkingUrl()	471
6.117.3.39 signalAssignSettings2Widget	471
6.117.3.40 slotAssignSettings2Widget	471
6.117.3.41 toolGroup()	472
6.117.3.42 toolGroupToString()	472
6.117.3.43 toolOperations()	472
6.117.3.44 toolVersion()	472
6.117.4 Member Data Documentation	472
6.117.4.1 m_settingsWidget	472
6.118 Digikam::BatchToolSet Class Reference	472
6.118.1 Detailed Description	473
6.118.2 Member Function Documentation	473
6.118.2.1 operator==()	473
6.119 Digikam::BatchToolsFactory Class Reference	473

---

6.120 Digikam::BCGContainer Class Reference	474
6.121 Digikam::BCGFilter Class Reference	475
6.121.1 Member Function Documentation	478
6.121.1.1 filterAction()	478
6.121.1.2 filterIdentifier()	478
6.121.1.3 readParameters()	478
6.122 Digikam::BCGSettings Class Reference	478
6.123 Digikam::BdEngineBackend Class Reference	479
6.123.1 Member Enumeration Documentation	482
6.123.1.1 QueryStateEnum	482
6.123.1.2 Status	482
6.123.2 Constructor & Destructor Documentation	482
6.123.2.1 BdEngineBackend()	482
6.123.3 Member Function Documentation	482
6.123.3.1 asDBDateTime()	482
6.123.3.2 beginTransaction()	483
6.123.3.3 checkOrSetWALMode()	483
6.123.3.4 close()	483
6.123.3.5 commitTransaction()	483
6.123.3.6 configElement()	483
6.123.3.7 connectionErrorHandling()	483
6.123.3.8 copyQuery()	483
6.123.3.9 databaseType()	484
6.123.3.10 exec()	484
6.123.3.11 execDBAction() [1/2]	484
6.123.3.12 execDBAction() [2/2]	484
6.123.3.13 execDBActionQuery()	484
6.123.3.14 execDirectSql()	484
6.123.3.15 execDirectSqlWithResult()	485
6.123.3.16 execQuery() [1/3]	485
6.123.3.17 execQuery() [2/3]	485
6.123.3.18 execQuery() [3/3]	485
6.123.3.19 execSql() [1/2]	485
6.123.3.20 execSql() [2/2]	486
6.123.3.21 execUpsertDBAction()	486
6.123.3.22 getDBAction()	486
6.123.3.23 getQuery()	486
6.123.3.24 handleQueryResult()	486
6.123.3.25 isCompatible()	487
6.123.3.26 isInTransaction()	487
6.123.3.27 lastError()	487
6.123.3.28 lastSQLException()	487

6.123.3.29 maximumBoundValues()	487
6.123.3.30 open()	487
6.123.3.31 prepareQuery()	488
6.123.3.32 queryErrorHandling()	488
6.123.3.33 readToList()	488
6.123.3.34 rollbackTransaction()	488
6.123.3.35 setDbEngineErrorHandler()	488
6.123.3.36 setForeignKeyChecks()	488
6.123.3.37 status()	489
6.123.3.38 tables()	489
6.124 Digikam::BdEngineBackend::QueryState Class Reference	489
6.125 Digikam::BdEngineBackendPrivate Class Reference	490
6.125.1 Member Function Documentation	492
6.125.1.1 connectionErrorAbortQueries()	492
6.125.1.2 connectionErrorContinueQueries()	492
6.125.1.3 databaseForThread()	492
6.125.1.4 handleWithErrorHandler()	492
6.125.1.5 queryOperationWakeAll()	492
6.125.1.6 setQueryOperationFlag()	492
6.125.2 Member Data Documentation	493
6.125.2.1 currentValidity	493
6.126 Digikam::BdEngineBackendPrivate::AbstractUnlocker Class Reference	493
6.127 Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker Class Reference	494
6.128 Digikam::BdEngineBackendPrivate::BusyWaiter Class Reference	495
6.129 Digikam::BdEngineBackendPrivate::ErrorLocker Class Reference	497
6.129.1 Member Function Documentation	498
6.129.1.1 wait()	498
6.130 Digikam::BlackFrameListView Class Reference	498
6.131 Digikam::BlackFrameListViewItem Class Reference	499
6.132 Digikam::BlackFrameParser Class Reference	500
6.133 Digikam::BlackFrameToolTip Class Reference	501
6.133.1 Member Function Documentation	502
6.133.1.1 repositionRect()	502
6.133.1.2 tipContents()	502
6.134 Digikam::BlurDetector Class Reference	503
6.134.1 Member Function Documentation	504
6.134.1.1 detect()	504
6.135 Digikam::BlurFilter Class Reference	505
6.135.1 Constructor & Destructor Documentation	508
6.135.1.1 BlurFilter()	508
6.135.2 Member Function Documentation	508
6.135.2.1 filterAction()	508

---

6.135.2.2 filterIdentifier()	508
6.135.2.3 readParameters()	508
6.136 Digikam::BlurFXFilter Class Reference	509
6.136.1 Member Function Documentation	512
6.136.1.1 filterAction()	512
6.136.1.2 filterIdentifier()	512
6.136.1.3 readParameters()	512
6.137 Digikam::BookmarkNode Class Reference	513
6.138 Digikam::BookmarksDialog Class Reference	514
6.139 Digikam::BookmarksManager Class Reference	515
6.139.1 Detailed Description	516
6.140 Digikam::BookmarksMenu Class Reference	517
6.140.1 Detailed Description	518
6.140.2 Member Function Documentation	518
6.140.2.1 prePopulated()	518
6.141 Digikam::BookmarksModel Class Reference	519
6.141.1 Detailed Description	520
6.142 Digikam::BorderContainer Class Reference	520
6.143 Digikam::BorderFilter Class Reference	522
6.143.1 Constructor & Destructor Documentation	525
6.143.1.1 BorderFilter()	525
6.143.2 Member Function Documentation	525
6.143.2.1 filterAction()	525
6.143.2.2 filterIdentifier()	525
6.143.2.3 readParameters()	525
6.144 Digikam::BorderSettings Class Reference	526
6.145 Digikam::BqmInfoface Class Reference	527
6.145.1 Member Function Documentation	529
6.145.1.1 allItemInfoListFromCurrentQueue()	529
6.145.1.2 pendingItemInfoListFromCurrentQueue()	529
6.145.1.3 selectedItemInfoListFromCurrentQueue()	529
6.146 Digikam::BuildTrashCountersJob Class Reference	530
6.147 Digikam::BWSepiaContainer Class Reference	531
6.147.1 Member Enumeration Documentation	532
6.147.1.1 BlackWhiteConversionType	532
6.148 Digikam::BWSepiaFilter Class Reference	533
6.148.1 Member Function Documentation	536
6.148.1.1 filterAction()	536
6.148.1.2 filterIdentifier()	536
6.148.1.3 readParameters()	536
6.149 Digikam::BWSepiaSettings Class Reference	537
6.150 Digikam::CachedPixmapKey Class Reference	538

---

6.151 Digikam::CachedPixmap Class Reference . . . . .	538
6.152 Digikam::CameraAutoDetectThread Class Reference . . . . .	539
6.153 Digikam::CameraController Class Reference . . . . .	540
6.153.1 Member Function Documentation . . . . .	542
6.153.1.1 getThumbsInfo() . . . . .	542
6.154 Digikam::CameraFolderDialog Class Reference . . . . .	542
6.155 Digikam::CameraFolderItem Class Reference . . . . .	543
6.156 Digikam::CameraFolderView Class Reference . . . . .	544
6.157 Digikam::CameraHistoryUpdater Class Reference . . . . .	545
6.158 Digikam::CameraInfoDialog Class Reference . . . . .	546
6.159 Digikam::CameraItem Class Reference . . . . .	547
6.160 Digikam::CameraItemList Class Reference . . . . .	548
6.161 Digikam::CameraList Class Reference . . . . .	549
6.162 Digikam::CameraMessageBox Class Reference . . . . .	550
6.162.1 Member Function Documentation . . . . .	550
6.162.1.1 informationList() . . . . .	550
6.162.1.2 warningContinueCancelList() . . . . .	550
6.163 Digikam::CameraNameHelper Class Reference . . . . .	550
6.164 Digikam::CameraNameOption Class Reference . . . . .	551
6.164.1 Member Function Documentation . . . . .	552
6.164.1.1 parseOperation() . . . . .	552
6.165 Digikam::CameraSelection Class Reference . . . . .	553
6.166 Digikam::CameraThumbsCtrl Class Reference . . . . .	554
6.166.1 Member Function Documentation . . . . .	554
6.166.1.1 cameraController() . . . . .	554
6.166.1.2 getThumbInfo() . . . . .	555
6.166.1.3 updateThumbInfoFromCache() . . . . .	555
6.167 Digikam::CameraType Class Reference . . . . .	555
6.168 Digikam::CamItemInfo Class Reference . . . . .	555
6.168.1 Member Enumeration Documentation . . . . .	556
6.168.1.1 DownloadStatus . . . . .	556
6.168.2 Member Function Documentation . . . . .	557
6.168.2.1 isNull() . . . . .	557
6.168.2.2 operator"!="() . . . . .	557
6.168.2.3 operator=="() . . . . .	557
6.168.2.4 url() . . . . .	557
6.168.3 Member Data Documentation . . . . .	557
6.168.3.1 downloaded . . . . .	557
6.168.3.2 size . . . . .	558
6.169 Digikam::CamItemSortSettings Class Reference . . . . .	558
6.169.1 Member Enumeration Documentation . . . . .	559
6.169.1.1 SortOrder . . . . .	559

---

6.169.2 Member Function Documentation	559
6.169.2.1 compare()	559
6.169.2.2 compareByOrder()	559
6.169.2.3 compareCategories()	560
6.169.2.4 compareValue()	560
6.169.2.5 lessThan() [1/2]	560
6.169.2.6 lessThan() [2/2]	560
6.169.2.7 lessThanByOrder()	560
6.169.2.8 naturalCompare()	561
6.170 Digikam::Canvas Class Reference	562
6.170.1 Member Function Documentation	565
6.170.1.1 applyTransform()	565
6.170.1.2 currentImage()	565
6.170.1.3 currentImageFileFormat()	565
6.170.1.4 currentImageFilePath()	566
6.170.1.5 exifRotated()	566
6.170.1.6 fitToSelect()	566
6.170.1.7 getSelectedArea()	566
6.170.1.8 imageHeight()	566
6.170.1.9 imageWidth()	566
6.170.1.10 interface()	566
6.170.1.11 isReadOnly()	566
6.170.1.12 setExifOrient()	567
6.170.1.13 setExposureSettings()	567
6.170.1.14 setICCSettings()	567
6.170.1.15 setSoftProofingEnabled()	567
6.171 Digikam::CaptionEdit Class Reference	568
6.172 Digikam::CaptionsMap Class Reference	570
6.172.1 Detailed Description	570
6.172.2 Member Function Documentation	571
6.172.2.1 setAuthorsList()	571
6.173 Digikam::CaptionValues Class Reference	571
6.174 Digikam::CaptureDlg Class Reference	571
6.175 Digikam::CaptureWidget Class Reference	572
6.176 Digikam::CaseModifier Class Reference	573
6.176.1 Member Function Documentation	574
6.176.1.1 parseOperation()	574
6.177 Digikam::CategorizedItemModel Class Reference	576
6.177.1 Member Enumeration Documentation	577
6.177.1.1 ExtraRoles	577
6.178 Digikam::CBContainer Class Reference	577
6.179 Digikam::CBFilter Class Reference	578

6.179.1 Member Function Documentation	581
6.179.1.1 filterAction()	581
6.179.1.2 filterIdentifier()	581
6.179.1.3 readParameters()	581
6.180 Digikam::CBSettings Class Reference	581
6.181 Digikam::ChangeBookmarkCommand Class Reference	582
6.182 Digikam::ChangeFaceRecognitionModelDig Class Reference	583
6.183 Digikam::ChangingDB Class Reference	583
6.184 Digikam::CharcoalFilter Class Reference	584
6.184.1 Member Function Documentation	587
6.184.1.1 filterAction()	587
6.184.1.2 filterIdentifier()	587
6.184.1.3 readParameters()	587
6.185 Digikam::CheckableAlbumFilterModel Class Reference	588
6.185.1 Detailed Description	590
6.185.2 Member Function Documentation	590
6.185.2.1 isFiltering()	590
6.185.2.2 matches()	591
6.186 Digikam::ChoiceSearchComboBox Class Reference	593
6.186.1 Constructor & Destructor Documentation	595
6.186.1.1 ChoiceSearchComboBox()	595
6.186.2 Member Function Documentation	595
6.186.2.1 installView()	595
6.186.2.2 setLabelText()	595
6.186.2.3 setSearchModel()	595
6.187 Digikam::ChoiceSearchModel Class Reference	596
6.187.1 Member Function Documentation	597
6.187.1.1 checkedDisplayTexts()	597
6.187.1.2 checkedKeys() [1/2]	597
6.187.1.3 checkedKeys() [2/2]	597
6.187.1.4 resetChecked()	598
6.187.1.5 setChecked() [1/3]	598
6.187.1.6 setChecked() [2/3]	598
6.187.1.7 setChecked() [3/3]	598
6.187.1.8 setChoice() [1/3]	598
6.187.1.9 setChoice() [2/3]	598
6.187.1.10 setChoice() [3/3]	599
6.188 Digikam::ChoiceSearchModel::Entry Class Reference	599
6.188.1 Member Function Documentation	599
6.188.1.1 operator==(())	599
6.189 Digikam::CIETongueWidget Class Reference	600
6.190 Digikam::ClickDragReleaseItem Class Reference	601

---

6.190.1 Member Function Documentation	602
6.190.1.1 mousePressEvent()	602
6.190.1.2 started	602
6.191 Digikam::ClockPhotoDialog Class Reference	603
6.191.1 Member Function Documentation	603
6.191.1.1 setImage()	603
6.192 Digikam::CMat Struct Reference	604
6.192.1 Detailed Description	604
6.192.2 Member Data Documentation	604
6.192.2.1 center	604
6.192.2.2 data	604
6.192.2.3 radius	604
6.192.2.4 row_stride	604
6.193 Digikam::CollectionImageChangeset Class Reference	605
6.193.1 Member Enumeration Documentation	605
6.193.1.1 Operation	605
6.193.2 Constructor & Destructor Documentation	606
6.193.2.1 CollectionImageChangeset()	606
6.193.3 Member Function Documentation	606
6.193.3.1 ids()	606
6.193.3.2 operator<<()	606
6.194 Digikam::CollectionLocation Class Reference	607
6.194.1 Member Enumeration Documentation	608
6.194.1.1 CaseSensitivity	608
6.194.1.2 Status	608
6.194.1.3 Type	609
6.194.2 Member Function Documentation	609
6.194.2.1 albumRootPath()	609
6.194.2.2 asQtCaseSensitivity()	609
6.194.2.3 caseSensitivity()	609
6.194.2.4 id()	610
6.194.2.5 label()	610
6.194.2.6 status()	610
6.194.2.7 type()	610
6.195 Digikam::CollectionManager Class Reference	611
6.195.1 Member Enumeration Documentation	613
6.195.1.1 LocationCheckResult	613
6.195.2 Member Function Documentation	613
6.195.2.1 addLocation()	613
6.195.2.2 album()	614
6.195.2.3 albumRoot()	614
6.195.2.4 albumRootLabel()	614



---

6.195.2.5 albumRootPath()	614
6.195.2.6 allAvailableAlbumRootPaths()	614
6.195.2.7 allAvailableLocations()	615
6.195.2.8 allLocations()	615
6.195.2.9 changeType()	615
6.195.2.10 checkHardWiredLocations()	615
6.195.2.11 checkLocation()	615
6.195.2.12 isAlbumRoot() [1/2]	615
6.195.2.13 isAlbumRoot() [2/2]	616
6.195.2.14 locationForAlbumRoot()	616
6.195.2.15 locationForAlbumRootId()	616
6.195.2.16 locationForUrl()	616
6.195.2.17 locationPropertiesChanged	616
6.195.2.18 locationStatusChanged	616
6.195.2.19 migrateToVolume()	617
6.195.2.20 migrationCandidates()	617
6.195.2.21 oneAlbumRoot()	617
6.195.2.22 refresh()	617
6.195.2.23 removeLocation()	617
6.195.2.24 setLabel()	617
6.195.2.25 setWatchDisabled()	618
6.196 Digikam::CollectionManager::Private Class Reference	618
6.196.1 Member Function Documentation	619
6.196.1.1 findVolumeForLocation()	619
6.196.1.2 findVolumeForUrl()	619
6.197 Digikam::CollectionPage Class Reference	620
6.198 Digikam::CollectionScanner Class Reference	622
6.198.1 Member Enumeration Documentation	624
6.198.1.1 FileScanMode	624
6.198.2 Member Function Documentation	625
6.198.2.1 cancelled	625
6.198.2.2 completeScan()	625
6.198.2.3 copyFileProperties()	625
6.198.2.4 createHintContainer()	625
6.198.2.5 databaseInitialScanDone()	625
6.198.2.6 finishCompleteScan()	625
6.198.2.7 finishedScanningAlbumRoot	626
6.198.2.8 getNewIdsList()	626
6.198.2.9 partialScan() [1/2]	626
6.198.2.10 partialScan() [2/2]	626
6.198.2.11 safelyRemoveAlbums()	626
6.198.2.12 scanFile() [1/3]	626

---

6.198.2.13 scanFile() [2/3]	627
6.198.2.14 scanFile() [3/3]	627
6.198.2.15 scannedFiles	627
6.198.2.16 setNeedFileCount()	627
6.198.2.17 setObserver()	627
6.198.2.18 setPerformFastScan()	627
6.198.2.19 setSignalsEnabled()	628
6.198.2.20 startScanningAlbumRoot	628
6.198.2.21 totalFilesToScan	628
6.199 Digikam::CollectionScanner::Private Class Reference	628
6.199.1 Member Function Documentation	629
6.199.1.1 finishScanner()	629
6.200 Digikam::CollectionScannerHintContainer Class Reference	630
6.201 Digikam::CollectionScannerHintContainerImplementation Class Reference	632
6.201.1 Member Function Documentation	633
6.201.1.1 clear()	633
6.201.1.2 recordHint()	633
6.201.1.3 recordHints() [1/3]	633
6.201.1.4 recordHints() [2/3]	634
6.201.1.5 recordHints() [3/3]	634
6.202 Digikam::CollectionScannerObserver Class Reference	635
6.203 Digikam::ColorCorrectionDlg Class Reference	636
6.204 Digikam::ColorFXContainer Class Reference	636
6.205 Digikam::ColorFXFilter Class Reference	637
6.205.1 Member Function Documentation	640
6.205.1.1 filterAction()	640
6.205.1.2 filterIdentifier()	640
6.205.1.3 readParameters()	640
6.206 Digikam::ColorFXSettings Class Reference	640
6.207 Digikam::ColorGradientWidget Class Reference	641
6.208 Digikam::ColorLabelFilter Class Reference	642
6.209 Digikam::ColorLabelMenuAction Class Reference	644
6.210 Digikam::ColorLabelSelector Class Reference	645
6.211 Digikam::ColorLabelWidget Class Reference	646
6.211.1 Member Function Documentation	647
6.211.1.1 colorLabels()	647
6.211.1.2 setButtonsExclusive()	648
6.211.1.3 setColorLabels()	648
6.211.1.4 setDescriptionBoxVisible()	648
6.212 Digikam::ComboBoxDelegate Class Reference	649
6.212.1 Member Function Documentation	650
6.212.1.1 paint()	650

6.212.1.2 startEditing()	650
6.213 Digikam::CommentInfo Class Reference	650
6.214 Digikam::CommonKeys Class Reference	651
6.214.1 Member Function Documentation	652
6.214.1.1 getDbValue()	652
6.215 Digikam::CompressionDetector Class Reference	653
6.215.1 Member Function Documentation	654
6.215.1.1 detect()	654
6.216 Digikam::ContentAwareContainer Class Reference	654
6.217 Digikam::ContentAwareFilter Class Reference	655
6.217.1 Member Function Documentation	658
6.217.1.1 filterAction()	658
6.217.1.2 filterIdentifier()	658
6.217.1.3 readParameters()	658
6.218 Digikam::ContextMenuHelper Class Reference	658
6.218.1 Detailed Description	661
6.218.2 Constructor & Destructor Documentation	661
6.218.2.1 ContextMenuHelper()	661
6.218.3 Member Function Documentation	661
6.218.3.1 addAction() [1/3]	661
6.218.3.2 addAction() [2/3]	662
6.218.3.3 addAction() [3/3]	662
6.218.3.4 addActionDeleteFaceTag()	662
6.218.3.5 addActionNewAlbum()	662
6.218.3.6 addActionNewTag()	663
6.218.3.7 addActionTagToFaceTag()	663
6.218.3.8 addAlbumCheckUncheckActions()	663
6.218.3.9 addAssignTagsMenu()	663
6.218.3.10 addCreateTagFromAddressbookMenu()	663
6.218.3.11 addExportMenu()	664
6.218.3.12 addGotoMenu()	664
6.218.3.13 addGroupMenu()	664
6.218.3.14 addImportMenu()	665
6.218.3.15 addIQSAction()	665
6.218.3.16 addLabelsAction()	665
6.218.3.17 addOpenAndNavigateActions()	665
6.218.3.18 addQueueManagerMenu()	666
6.218.3.19 addRemoveAllTags()	666
6.218.3.20 addRemoveTagsMenu()	666
6.218.3.21 addSeparator()	667
6.218.3.22 addServicesMenu()	667
6.218.3.23 addStandardActionCopy()	667

---

6.218.3.24 addStandardActionCut()	667
6.218.3.25 addStandardActionItemDelete()	668
6.218.3.26 addStandardActionLightTable()	668
6.218.3.27 addStandardActionPaste()	668
6.218.3.28 addStandardActionThumbnail()	668
6.218.3.29 addSubMenu()	669
6.218.3.30 exec()	669
6.218.3.31 setAlbumModel()	669
6.218.3.32 setItemFilterModel()	669
6.219 Digikam::ContextMenuHelper::Private Class Reference	670
6.220 Digikam::CoordinatesOverlayWidget Class Reference	670
6.221 Digikam::CopyOrMoveJob Class Reference	671
6.222 Digikam::CopyrightInfo Class Reference	672
6.223 Digikam::CoreDB Class Reference	673
6.223.1 Constructor & Destructor Documentation	678
6.223.1.1 CoreDB()	678
6.223.1.2 ~CoreDB()	678
6.223.2 Member Function Documentation	678
6.223.2.1 addAlbum()	678
6.223.2.2 addAlbumRoot()	679
6.223.2.3 addImageMetadata()	679
6.223.2.4 addImageRelation()	679
6.223.2.5 addImageRelations()	680
6.223.2.6 addImageTagProperty()	680
6.223.2.7 addItem()	680
6.223.2.8 addItemInformation()	680
6.223.2.9 addItemPosition()	681
6.223.2.10 addItemTag() [1/2]	681
6.223.2.11 addItemTag() [2/2]	681
6.223.2.12 addSearch()	682
6.223.2.13 addTag()	682
6.223.2.14 addTagProperty()	682
6.223.2.15 addTagsToItems()	683
6.223.2.16 addToDownloadHistory()	683
6.223.2.17 addVideoMetadata()	683
6.223.2.18 changeImageComment()	683
6.223.2.19 changeImageMetadata()	684
6.223.2.20 changeItemInformation()	684
6.223.2.21 changeItemPosition()	684
6.223.2.22 changeVideoMetadata()	684
6.223.2.23 copyAlbumProperties()	684
6.223.2.24 copyImageAttributes()	685

---

6.223.2.25 copyImageProperties()	685
6.223.2.26 copyImageTags()	685
6.223.2.27 copyItem()	685
6.223.2.28 databaseUuid()	685
6.223.2.29 deleteAlbum()	686
6.223.2.30 deleteAlbumRoot()	686
6.223.2.31 deleteItem() [1/2]	686
6.223.2.32 deleteItem() [2/2]	686
6.223.2.33 deleteObsoleteItem()	687
6.223.2.34 deleteRemovedItems()	687
6.223.2.35 deleteSearch()	687
6.223.2.36 deleteSearches()	687
6.223.2.37 deleteStaleAlbums()	687
6.223.2.38 deleteTag()	688
6.223.2.39 findByNameAndCreationDate()	688
6.223.2.40 findImageId()	688
6.223.2.41 findInDownloadHistory()	688
6.223.2.42 getAlbumAndSubalbumsForPath()	689
6.223.2.43 getAlbumAverageDate()	689
6.223.2.44 getAlbumForPath()	689
6.223.2.45 getAlbumHighestDate()	690
6.223.2.46 getAlbumLowestDate()	690
6.223.2.47 getAlbumModificationDate()	690
6.223.2.48 getAlbumModificationMap()	691
6.223.2.49 getAlbumRelativePath()	691
6.223.2.50 getAlbumRootId()	691
6.223.2.51 getAlbumRoots()	692
6.223.2.52 getAlbumShortInfos()	692
6.223.2.53 getAlbumsOnAlbumRoot()	692
6.223.2.54 getAllCreationDates()	692
6.223.2.55 getAllItems()	692
6.223.2.56 getAllItemsWithAlbum()	692
6.223.2.57 getDatabaseEncoding()	693
6.223.2.58 getDirtyOrMissingFacelImageUrls()	693
6.223.2.59 getFilterSettings()	693
6.223.2.60 getFirstItemWithFaceTag()	693
6.223.2.61 getFormatStatistics()	693
6.223.2.62 getIdenticalFiles()	693
6.223.2.63 getImageId()	693
6.223.2.64 getImageIds() [1/4]	694
6.223.2.65 getImageIds() [2/4]	694
6.223.2.66 getImageIds() [3/4]	694

6.223.2.67 <a href="#">getImageIds()</a> [ 4 / 4 ]	696
6.223.2.68 <a href="#">getImageMetadata()</a>	696
6.223.2.69 <a href="#">getImageProperty()</a>	696
6.223.2.70 <a href="#">getImagesFields()</a>	697
6.223.2.71 <a href="#">getImagesRelatedFrom()</a>	697
6.223.2.72 <a href="#">getImagesRelatingTo()</a>	697
6.223.2.73 <a href="#">getImagesWithImageTagProperty()</a>	697
6.223.2.74 <a href="#">getImagesWithProperty()</a>	697
6.223.2.75 <a href="#">getImageTagProperties()</a>	697
6.223.2.76 <a href="#">getImageUuid()</a>	698
6.223.2.77 <a href="#">getItemAlbum()</a>	698
6.223.2.78 <a href="#">getItemComments()</a>	698
6.223.2.79 <a href="#">getItemCommonTagIDs()</a>	698
6.223.2.80 <a href="#">getItemCopyright()</a>	698
6.223.2.81 <a href="#">getItemFromAlbum()</a>	699
6.223.2.82 <a href="#">getItemHistory()</a>	699
6.223.2.83 <a href="#">getItemIDsAndURLsInAlbum()</a>	699
6.223.2.84 <a href="#">getItemIDsInAlbum()</a>	700
6.223.2.85 <a href="#">getItemIDsInTag()</a>	700
6.223.2.86 <a href="#">getItemInformation()</a>	700
6.223.2.87 <a href="#">getItemName()</a>	700
6.223.2.88 <a href="#">getItemNamesInAlbum()</a>	701
6.223.2.89 <a href="#">getItemPosition()</a>	701
6.223.2.90 <a href="#">getItemScanInfo()</a>	701
6.223.2.91 <a href="#">getItemScanInfos()</a>	701
6.223.2.92 <a href="#">getItemForUuid()</a>	702
6.223.2.93 <a href="#">getItemShortInfo()</a> [ 1 / 2 ]	702
6.223.2.94 <a href="#">getItemShortInfo()</a> [ 2 / 2 ]	702
6.223.2.95 <a href="#">getItemTagIDs()</a>	702
6.223.2.96 <a href="#">getItemURLsWithTag()</a>	702
6.223.2.97 <a href="#">getItemTagIDs()</a>	702
6.223.2.98 <a href="#">getItemTagNames()</a>	703
6.223.2.99 <a href="#">getItemURLsInAlbum()</a>	703
6.223.2.100 <a href="#">getItemURLsInTag()</a>	703
6.223.2.101 <a href="#">getListFromImageMetadata()</a>	704
6.223.2.102 <a href="#">getNumberOfAllItemsAndAlbums()</a>	704
6.223.2.103 <a href="#">getNumberOfImagesInAlbums()</a>	704
6.223.2.104 <a href="#">getNumberOfImagesInTagProperties()</a> [ 1 / 2 ]	704
6.223.2.105 <a href="#">getNumberOfImagesInTagProperties()</a> [ 2 / 2 ]	705
6.223.2.106 <a href="#">getNumberOfImagesInTags()</a>	705
6.223.2.107 <a href="#">getNumberOfItemsInAlbum()</a>	705
6.223.2.108 <a href="#">getObsoleteItemIds()</a>	705

---

6.223.2.109	getOneRelatedImageEach()	705
6.223.2.110	getRecentlyAssignedTags()	705
6.223.2.111	getRelatedImagesToByType()	706
6.223.2.112	getRelationCloud()	706
6.223.2.113	getSearchInfo()	706
6.223.2.114	getSearchQuery()	706
6.223.2.115	getSetting()	706
6.223.2.116	getTagIdsWithProperties()	707
6.223.2.117	getTagProperties() [1/3]	707
6.223.2.118	getTagProperties() [2/3]	707
6.223.2.119	getTagProperties() [3/3]	707
6.223.2.120	getTagShortInfos()	707
6.223.2.121	getTagsWithProperty()	707
6.223.2.122	getUniqueHashVersion()	707
6.223.2.123	getUserFilterSettings()	708
6.223.2.124	getVideoMetadata()	708
6.223.2.125	hasImageHistory()	708
6.223.2.126	hasTags()	708
6.223.2.127	integrityCheck()	708
6.223.2.128	makeStaleAlbum()	709
6.223.2.129	migrateAlbumRoot()	709
6.223.2.130	moveItem()	709
6.223.2.131	removeAllImageComments()	709
6.223.2.132	removeAllItemCopyrightProperties()	709
6.223.2.133	removeImageComment()	710
6.223.2.134	removeImageRelation()	710
6.223.2.135	removeImageTagProperties()	710
6.223.2.136	removeItemAllTags()	710
6.223.2.137	removeItemCopyrightProperties()	711
6.223.2.138	removeItemPosition()	711
6.223.2.139	removeItemPositionAltitude()	711
6.223.2.140	removeItems()	711
6.223.2.141	removeItemsFromAlbum()	711
6.223.2.142	removeItemsPermanently()	712
6.223.2.143	removeItemTag()	712
6.223.2.144	removeTagProperties()	712
6.223.2.145	removeTagsFromItems()	713
6.223.2.146	renameAlbum()	713
6.223.2.147	renameItem()	713
6.223.2.148	scanAlbums()	713
6.223.2.149	scanSearches()	713
6.223.2.150	scanTags()	714

6.223.2.151 setAlbumCaption()	714
6.223.2.152 setAlbumCategory()	715
6.223.2.153 setAlbumDate()	715
6.223.2.154 setAlbumIcon()	715
6.223.2.155 setAlbumModificationDate()	716
6.223.2.156 setAlbumRootCaseSensitivity()	716
6.223.2.157 setAlbumRootLabel()	716
6.223.2.158 setAlbumRootPath()	716
6.223.2.159 setAlbumRootType()	717
6.223.2.160 setFilterSettings()	717
6.223.2.161 setImageComment()	717
6.223.2.162 setImageProperty()	717
6.223.2.163 setItemAlbum()	718
6.223.2.164 setItemCopyrightProperty()	718
6.223.2.165 setItemHistory()	718
6.223.2.166 setItemManualOrder()	718
6.223.2.167 setItemModificationDate()	718
6.223.2.168 setItemStatus()	719
6.223.2.169 setSetting()	719
6.223.2.170 setTagIcon()	719
6.223.2.171 setTagName()	719
6.223.2.172 setTagParentID()	720
6.223.2.173 setUserFilterSettings()	720
6.223.2.174 updateItem()	720
6.223.2.175 updateSearch()	721
6.223.2.176 vacuum()	721
6.224 Digikam::CoreDbAccess Class Reference	721
6.224.1 Detailed Description	722
6.224.2 Constructor & Destructor Documentation	722
6.224.2.1 CoreDbAccess()	722
6.224.3 Member Function Documentation	722
6.224.3.1 backend()	722
6.224.3.2 checkReadyForUse()	722
6.224.3.3 cleanUpDatabase()	723
6.224.3.4 databaseWatch()	723
6.224.3.5 db()	723
6.224.3.6 initDbEngineErrorHandler()	723
6.224.3.7 lastError()	723
6.224.3.8 parameters()	723
6.224.3.9 setLastError()	723
6.224.3.10 setParameters()	724
6.225 Digikam::CoreDbAccessUnlock Class Reference	724



6.225.1 Constructor & Destructor Documentation	724
6.225.1.1 CoreDbAccessUnlock()	724
6.226 Digikam::CoreDbBackend Class Reference	725
6.226.1 Member Function Documentation	728
6.226.1.1 initSchema()	728
6.226.1.2 recordChangeset()	728
6.226.1.3 setCoreDbWatch()	728
6.227 Digikam::CoreDbBackendPrivate Class Reference	729
6.227.1 Member Function Documentation	731
6.227.1.1 transactionFinished()	731
6.228 Digikam::CoreDbBackendPrivate::ChangesetContainer< T > Class Template Reference	731
6.229 Digikam::CoreDbCopyManager Class Reference	732
6.230 Digikam::CoreDbDownloadHistory Class Reference	732
6.230.1 Member Function Documentation	733
6.230.1.1 setDownloaded()	733
6.230.1.2 status()	733
6.231 Digikam::CoreDbNameFilter Class Reference	733
6.231.1 Constructor & Destructor Documentation	733
6.231.1.1 CoreDbNameFilter()	733
6.231.2 Member Function Documentation	734
6.231.2.1 matches()	734
6.232 Digikam::CoreDbOperationGroup Class Reference	734
6.232.1 Detailed Description	734
6.232.2 Constructor & Destructor Documentation	734
6.232.2.1 CoreDbOperationGroup() [1/2]	734
6.232.2.2 CoreDbOperationGroup() [2/2]	734
6.232.3 Member Function Documentation	734
6.232.3.1 allowLift()	734
6.232.3.2 lift()	735
6.232.3.3 resetTime()	735
6.233 Digikam::CoreDbPrivilegesChecker Class Reference	735
6.234 Digikam::CoreDbSchemaUpdater Class Reference	735
6.235 Digikam::CoreDbTransaction Class Reference	735
6.235.1 Detailed Description	736
6.235.2 Constructor & Destructor Documentation	736
6.235.2.1 CoreDbTransaction() [1/2]	736
6.235.2.2 CoreDbTransaction() [2/2]	736
6.236 Digikam::CoreDbUrl Class Reference	737
6.236.1 Constructor & Destructor Documentation	739
6.236.1.1 CoreDbUrl() [1/2]	739
6.236.1.2 CoreDbUrl() [2/2]	739
6.236.2 Member Function Documentation	739

---

6.236.2.1 album()	739
6.236.2.2 albumRoot()	739
6.236.2.3 albumRootId()	739
6.236.2.4 albumRootPath()	739
6.236.2.5 albumUrl()	740
6.236.2.6 areaCoordinates()	740
6.236.2.7 dateUrl()	740
6.236.2.8 endDate()	740
6.236.2.9 fileUrl()	740
6.236.2.10 fromAlbumAndName()	740
6.236.2.11 fromDateForMonth()	741
6.236.2.12 fromDateForYear()	741
6.236.2.13 fromDateRange()	741
6.236.2.14 fromFileUrl()	741
6.236.2.15 fromTagIds()	742
6.236.2.16 isAlbumUrl()	742
6.236.2.17 mapImagesUrl()	742
6.236.2.18 name()	742
6.236.2.19 parameters()	742
6.236.2.20 searchId()	742
6.236.2.21 searchUrl()	742
6.236.2.22 setParameters()	743
6.236.2.23 startDate()	743
6.236.2.24 tagId()	743
6.236.2.25 tagIds()	743
6.237 Digikam::CoreDbWatch Class Reference	744
6.237.1 Member Function Documentation	745
6.237.1.1 databaseChanged	745
6.237.1.2 imageChange	745
6.238 Digikam::CountrySelector Class Reference	746
6.239 Digikam::CtrlButton Class Reference	747
6.240 Digikam::CurvesBox Class Reference	748
6.241 Digikam::CurvesContainer Class Reference	749
6.241.1 Constructor & Destructor Documentation	749
6.241.1.1 CurvesContainer()	749
6.241.2 Member Function Documentation	750
6.241.2.1 initialize()	750
6.241.2.2 isEmpty()	750
6.241.2.3 isStoredLosslessly()	750
6.241.3 Member Data Documentation	750
6.241.3.1 curvesType	750
6.242 Digikam::CurvesFilter Class Reference	751

---

6.242.1 Member Function Documentation	754
6.242.1.1 filterAction()	754
6.242.1.2 filterIdentifier()	754
6.242.1.3 readParameters()	754
6.243 Digikam::CurvesSettings Class Reference	755
6.244 Digikam::CurvesWidget Class Reference	757
6.244.1 Member Function Documentation	758
6.244.1.1 reset()	758
6.244.1.2 resetUI()	758
6.244.1.3 restoreCurve()	758
6.244.1.4 saveCurve()	759
6.244.1.5 stopHistogramComputation()	759
6.244.1.6 updateData()	759
6.245 Digikam::CustomStepsDoubleSpinBox Class Reference	760
6.245.1 Constructor & Destructor Documentation	760
6.245.1.1 CustomStepsDoubleSpinBox()	760
6.245.2 Member Function Documentation	761
6.245.2.1 reset()	761
6.245.2.2 setSingleSteps()	761
6.245.2.3 setSuggestedInitialValue()	761
6.245.2.4 setSuggestedValues()	761
6.246 Digikam::CustomStepsIntSpinBox Class Reference	762
6.246.1 Constructor & Destructor Documentation	763
6.246.1.1 CustomStepsIntSpinBox()	763
6.246.2 Member Function Documentation	763
6.246.2.1 enableFractionMagic()	763
6.246.2.2 fractionMagicValue()	763
6.246.2.3 reset()	763
6.246.2.4 setSingleSteps()	763
6.246.2.5 setSuggestedInitialValue()	764
6.246.2.6 setSuggestedValues()	764
6.247 Digikam::DAboutData Class Reference	764
6.248 Digikam::DAbstractSliderSpinBox Class Reference	766
6.248.1 Member Function Documentation	767
6.248.1.1 setBlockUpdateSignalOnDrag()	767
6.248.1.2 setInternalValue()	768
6.249 Digikam::DActiveLabel Class Reference	768
6.249.1 Detailed Description	768
6.250 Digikam::DAdjustableLabel Class Reference	769
6.250.1 Detailed Description	769
6.251 Digikam::DAlbum Class Reference	770
6.251.1 Detailed Description	772

6.251.2 Member Function Documentation	772
6.251.2.1 databaseUrl()	772
6.252 Digikam::DAAlbumDrag Class Reference	772
6.252.1 Detailed Description	773
6.253 Digikam::DAAlbumInfo Class Reference	773
6.254 Digikam::DAArrowClickLabel Class Reference	774
6.255 Digikam::DatabaseBlob Class Reference	775
6.255.1 Detailed Description	775
6.255.2 Member Function Documentation	775
6.255.2.1 read()	775
6.256 Digikam::DatabaseCopyThread Class Reference	776
6.257 Digikam::DatabaseFields::DatabaseFieldsEnumIterator< FieldName > Class Template Reference	776
6.257.1 Detailed Description	777
6.258 Digikam::DatabaseFields::DatabaseFieldsEnumIteratorSetOnly< FieldName > Class Template Reference	777
6.258.1 Detailed Description	777
6.259 Digikam::DatabaseFields::FieldMetaInfo< FieldName > Class Template Reference	778
6.260 Digikam::DatabaseFields::Hash< T > Class Template Reference	778
6.260.1 Detailed Description	779
6.261 Digikam::DatabaseFields::Set Class Reference	779
6.261.1 Detailed Description	779
6.262 Digikam::DatabaseLoadSaveFileInfoProvider Class Reference	780
6.262.1 Member Function Documentation	780
6.262.1.1 dimensionsHint()	780
6.262.1.2 orientationHint()	780
6.263 Digikam::DatabaseMigrationDialog Class Reference	781
6.264 Digikam::DatabaseOption Class Reference	782
6.264.1 Member Function Documentation	783
6.264.1.1 parseOperation()	783
6.265 Digikam::DatabaseOptionDialog Class Reference	784
6.266 Digikam::DatabasePage Class Reference	786
6.267 Digikam::DatabaseServer Class Reference	787
6.267.1 Member Function Documentation	788
6.267.1.1 isRunning()	788
6.267.1.2 startDatabaseProcess()	788
6.267.1.3 stopDatabaseProcess()	788
6.268 Digikam::DatabaseServerError Class Reference	788
6.268.1 Member Enumeration Documentation	788
6.268.1.1 DatabaseServerErrorEnum	788
6.269 Digikam::DatabaseServerStarter Class Reference	789
6.269.1 Member Function Documentation	789
6.269.1.1 instance()	789

---

6.270 Digikam::DatabaseSettingsWidget Class Reference . . . . .	790
6.270.1 Member Function Documentation . . . . .	791
6.270.1.1 checkDatabaseSettings() . . . . .	791
6.271 Digikam::DatabaseSettingsWidget::Private Class Reference . . . . .	791
6.272 Digikam::DatabaseTask Class Reference . . . . .	792
6.272.1 Member Function Documentation . . . . .	793
6.272.1.1 signalAddItemsToProcess . . . . .	793
6.273 Digikam::DatabaseVersionManager Class Reference . . . . .	794
6.273.1 Member Function Documentation . . . . .	795
6.273.1.1 toplevelDirectory() . . . . .	795
6.274 Digikam::DatabaseWorkerInterface Class Reference . . . . .	796
6.275 Digikam::DatabaseWriter Class Reference . . . . .	799
6.276 Digikam::DateAlbumModel Class Reference . . . . .	802
6.276.1 Detailed Description . . . . .	806
6.276.2 Constructor & Destructor Documentation . . . . .	806
6.276.2.1 DateAlbumModel() . . . . .	806
6.276.3 Member Function Documentation . . . . .	807
6.276.3.1 albumForId() . . . . .	807
6.276.3.2 albumName() . . . . .	807
6.276.3.3 decorationRoleData() . . . . .	807
6.276.3.4 monthIndexForDate() . . . . .	807
6.276.3.5 setPixmaps() . . . . .	807
6.276.3.6 sortRoleData() . . . . .	808
6.277 Digikam::DateFolderView Class Reference . . . . .	809
6.277.1 Member Function Documentation . . . . .	811
6.277.1.1 doLoadState() . . . . .	811
6.277.1.2 doSaveState() . . . . .	811
6.277.1.3 setConfigGroup() . . . . .	811
6.278 Digikam::DateFolderViewSideBarWidget Class Reference . . . . .	812
6.278.1 Member Function Documentation . . . . .	813
6.278.1.1 applySettings() . . . . .	813
6.278.1.2 changeAlbumFromHistory() . . . . .	814
6.278.1.3 doLoadState() . . . . .	814
6.278.1.4 doSaveState() . . . . .	814
6.278.1.5 getCaption() . . . . .	814
6.278.1.6 getIcon() . . . . .	814
6.278.1.7 setActive() . . . . .	814
6.279 Digikam::DateFormat Class Reference . . . . .	815
6.280 Digikam::DateOption Class Reference . . . . .	816
6.280.1 Member Function Documentation . . . . .	817
6.280.1.1 parseOperation() . . . . .	817
6.281 Digikam::DateOptionDialog Class Reference . . . . .	818

6.282 Digikam::DatePickerValidator Class Reference . . . . .	819
6.283 Digikam::DatePickerYearSelector Class Reference . . . . .	820
6.283.1 Constructor & Destructor Documentation . . . . .	821
6.283.1.1 DatePickerYearSelector() . . . . .	821
6.284 Digikam::DatesDBJobInfo Class Reference . . . . .	821
6.285 Digikam::DatesDBJobsThread Class Reference . . . . .	823
6.285.1 Member Function Documentation . . . . .	824
6.285.1.1 datesListing() . . . . .	824
6.286 Digikam::DatesJob Class Reference . . . . .	825
6.287 Digikam::DateTreeView Class Reference . . . . .	827
6.288 Digikam::DbCleaner Class Reference . . . . .	831
6.288.1 Member Function Documentation . . . . .	833
6.288.1.1 setUseMultiCoreCPU() . . . . .	833
6.289 Digikam::DbEngineAccess Class Reference . . . . .	834
6.289.1 Detailed Description . . . . .	834
6.289.2 Member Function Documentation . . . . .	834
6.289.2.1 checkReadyForUse() . . . . .	834
6.290 Digikam::DbEngineAction Class Reference . . . . .	834
6.291 Digikam::DbEngineActionElement Class Reference . . . . .	834
6.292 Digikam::DbEngineActionType Class Reference . . . . .	834
6.292.1 Detailed Description . . . . .	835
6.292.2 Member Function Documentation . . . . .	835
6.292.2.1 getActionValue() . . . . .	835
6.292.2.2 isValue() . . . . .	835
6.292.2.3 setActionValue() . . . . .	835
6.292.2.4 setValue() . . . . .	835
6.293 Digikam::DbEngineConfig Class Reference . . . . .	835
6.294 Digikam::DbEngineConfigSettings Class Reference . . . . .	836
6.295 Digikam::DbEngineConfigSettingsLoader Class Reference . . . . .	836
6.296 Digikam::DbEngineConnectionChecker Class Reference . . . . .	837
6.297 Digikam::DbEngineErrorAnswer Class Reference . . . . .	838
6.297.1 Member Function Documentation . . . . .	839
6.297.1.1 connectionErrorContinueQueries() . . . . .	839
6.298 Digikam::DbEngineErrorHandler Class Reference . . . . .	839
6.298.1 Member Function Documentation . . . . .	840
6.298.1.1 connectionError . . . . .	840
6.298.1.2 consultUserForError . . . . .	840
6.299 Digikam::DbEngineGuiErrorHandler Class Reference . . . . .	841
6.300 Digikam::DbEngineLocking Class Reference . . . . .	842
6.301 Digikam::DbEngineParameters Class Reference . . . . .	842
6.301.1 Detailed Description . . . . .	843
6.301.2 Constructor & Destructor Documentation . . . . .	844

6.301.2.1 DbEngineParameters()	844
6.301.3 Member Function Documentation	844
6.301.3.1 defaultMysqlAdminCmd()	844
6.301.3.2 defaultMysqlInitCmd()	844
6.301.3.3 defaultMysqlServerCmd()	844
6.301.3.4 defaultMysqlUpgradeCmd()	844
6.301.3.5 defaultParameters()	844
6.301.3.6 faceParameters()	844
6.301.3.7 getCoreDatabaseNameOrDir()	845
6.301.3.8 hash()	845
6.301.3.9 isValid()	845
6.301.3.10 parametersForSQLite()	845
6.301.3.11 readFromConfig()	845
6.301.3.12 serverPrivatePath()	845
6.301.3.13 setCoreDatabasePath()	845
6.301.3.14 setInternalServerPath()	846
6.301.3.15 similarityParameters()	846
6.301.3.16 SQLiteDatabaseType()	846
6.301.3.17 thumbnailParameters()	846
6.301.4 Member Data Documentation	846
6.301.4.1 internalServerMysqlInitCmd	846
6.302 Digikam::DbEngineSqlQuery Class Reference	847
6.303 Digikam::DbEngineThreadData Class Reference	847
6.304 Digikam::DbHeaderListItem Class Reference	848
6.305 Digikam::DBinaryIface Class Reference	849
6.306 Digikam::DBinarySearch Class Reference	851
6.306.1 Detailed Description	852
6.307 Digikam::DBInfoIface Class Reference	852
6.307.1 Member Function Documentation	854
6.307.1.1 albumChooser()	854
6.307.1.2 albumChooserItems()	854
6.307.1.3 albumInfo()	854
6.307.1.4 albumItems()	854
6.307.1.5 albumsItems()	855
6.307.1.6 allAlbumItems()	855
6.307.1.7 currentAlbumItems()	855
6.307.1.8 currentSelectedItems()	855
6.307.1.9 defaultUploadUrl()	855
6.307.1.10 deleteImage()	855
6.307.1.11 itemInfo()	855
6.307.1.12 openSetupPage()	856
6.307.1.13 parseAlbumItemsRecursive()	856

---

6.307.1.14 passShortcutActionsToWidget()	856
6.307.1.15 setItemInfo()	856
6.307.1.16 supportAlbums()	856
6.307.1.17 tagFilterModel()	856
6.307.1.18 uploadUrl()	856
6.307.1.19 uploadWidget()	857
6.308 Digikam::DBJob Class Reference	857
6.309 Digikam::DBJobInfo Class Reference	858
6.310 Digikam::DBJobsManager Class Reference	859
6.310.1 Member Function Documentation	860
6.310.1.1 instance()	860
6.310.1.2 startAlbumsJobThread()	860
6.310.1.3 startDatesJobThread()	860
6.310.1.4 startGPSJobThread()	860
6.310.1.5 startSearchesJobThread()	861
6.310.1.6 startTagsJobThread()	861
6.311 Digikam::DBJobsThread Class Reference	862
6.311.1 Member Function Documentation	863
6.311.1.1 connectFinishAndErrorSignals()	863
6.311.1.2 error	863
6.311.1.3 errorsList()	863
6.311.1.4 hasErrors()	864
6.312 Digikam::DbKeysCollection Class Reference	864
6.312.1 Detailed Description	865
6.312.2 Constructor & Destructor Documentation	865
6.312.2.1 DbKeysCollection()	865
6.312.3 Member Function Documentation	865
6.312.3.1 addId()	865
6.312.3.2 collectionName()	865
6.312.3.3 getDbValue()	866
6.312.3.4 getValue()	867
6.312.3.5 ids()	867
6.313 Digikam::DbKeySelector Class Reference	868
6.314 Digikam::DbKeySelectorItem Class Reference	869
6.315 Digikam::DbKeySelectorView Class Reference	870
6.316 Digikam::DbShrinkDialog Class Reference	871
6.317 Digikam::DBStatDlg Class Reference	872
6.318 Digikam::DBusyDlg Class Reference	873
6.319 Digikam::DBusyThread Class Reference	874
6.320 Digikam::DCameraDragObject Class Reference	875
6.320.1 Detailed Description	875
6.321 Digikam::DCameraItemDrag Class Reference	876



6.321.1 Detailed Description	876
6.322 Digikam::DCategorizedSortFilterProxyModel Class Reference	877
6.322.1 Detailed Description	878
6.322.2 Member Enumeration Documentation	878
6.322.2.1 AdditionalRoles	878
6.322.3 Member Function Documentation	879
6.322.3.1 compareCategories()	879
6.322.3.2 isCategorizedModel()	879
6.322.3.3 lessThan()	880
6.322.3.4 setCategorizedModel()	880
6.322.3.5 setSortCategoriesByNaturalComparison()	880
6.322.3.6 sort()	880
6.322.3.7 sortCategoriesByNaturalComparison()	881
6.322.3.8 sortColumn()	881
6.322.3.9 sortOrder()	881
6.322.3.10 subSortLessThan()	881
6.323 Digikam::DCategorizedSortFilterProxyModel::Private Class Reference	882
6.324 Digikam::DCategorizedView Class Reference	882
6.324.1 Detailed Description	884
6.324.2 Member Function Documentation	884
6.324.2.1 categorizedIndexesIn()	884
6.324.2.2 categoryAt()	885
6.324.2.3 categoryRange()	885
6.324.2.4 categoryVisualRect()	885
6.324.2.5 setDrawDraggedItems()	885
6.325 Digikam::DCategorizedView::Private Class Reference	886
6.325.1 Member Function Documentation	887
6.325.1.1 cacheCategory()	887
6.325.1.2 cachedRectCategory()	887
6.325.1.3 cachedRectIndex()	887
6.325.1.4 cacheIndex()	887
6.325.1.5 categoryUpperBound()	888
6.325.1.6 categoryVisualRect()	888
6.325.1.7 contentsSize()	888
6.325.1.8 drawDraggedItems() [1/2]	888
6.325.1.9 drawDraggedItems() [2/2]	888
6.325.1.10 drawNewCategory()	888
6.325.1.11 intersectionSet()	889
6.325.1.12 selectionForRect()	889
6.325.1.13 updateScrollbars()	889
6.325.1.14 visualCategoryRectInViewport()	889
6.325.1.15 visualRect()	889

---

6.325.1.16 visualRectInViewport()	889
6.325.2 Member Data Documentation	890
6.325.2.1 elementsInfo	890
6.326 Digikam::DCategorizedView::Private::ElementInfo Class Reference	890
6.327 Digikam::DCategoryDrawer Class Reference	891
6.327.1 Detailed Description	892
6.327.2 Constructor & Destructor Documentation	892
6.327.2.1 DCategoryDrawer()	892
6.327.3 Member Function Documentation	892
6.327.3.1 actionRequested	892
6.327.3.2 categoryHeight()	893
6.327.3.3 collapseOrExpandClicked	893
6.327.3.4 drawCategory()	893
6.327.3.5 leftMargin()	893
6.327.3.6 mouseButtonDoubleClicked()	894
6.327.3.7 mouseButtonPressed()	894
6.327.3.8 mouseButtonReleased()	894
6.327.3.9 mouseLeft()	895
6.327.3.10 mouseMoved()	895
6.327.3.11 rightMargin()	895
6.327.3.12 view()	896
6.328 Digikam::DClickLabel Class Reference	896
6.329 Digikam::DColor Class Reference	897
6.329.1 Constructor & Destructor Documentation	898
6.329.1.1 DColor() [1/4]	898
6.329.1.2 DColor() [2/4]	898
6.329.1.3 DColor() [3/4]	898
6.329.1.4 DColor() [4/4]	898
6.329.2 Member Function Documentation	898
6.329.2.1 blendZero()	898
6.329.2.2 convertToSixteenBit()	898
6.329.2.3 getHSL()	899
6.329.2.4 getYCbCr()	899
6.329.2.5 premultiply()	899
6.329.2.6 setColor()	899
6.329.2.7 setHSL()	899
6.329.2.8 setPixel()	900
6.329.2.9 setYCbCr()	900
6.330 Digikam::DColorComposer Class Reference	900
6.330.1 Member Enumeration Documentation	901
6.330.1.1 CompositingOperation	901
6.330.2 Member Function Documentation	901

---

6.330.2.1 compose() [1/2]	901
6.330.2.2 compose() [2/2]	901
6.330.2.3 getComposer()	902
6.331 Digikam::DColorSelector Class Reference	902
6.331.1 Detailed Description	902
6.332 Digikam::DColorValueSelector Class Reference	903
6.332.1 Member Function Documentation	905
6.332.1.1 chooserMode()	905
6.332.1.2 colorValue()	905
6.332.1.3 drawContents()	905
6.332.1.4 drawPalette()	905
6.332.1.5 hue()	906
6.332.1.6 saturation()	906
6.332.1.7 setChooserMode()	906
6.332.1.8 setColorValue()	906
6.332.1.9 setHue()	906
6.332.1.10 setSaturation()	907
6.332.1.11 updateContents()	907
6.333 Digikam::DComboBox Class Reference	908
6.334 Digikam::DConfigDlg Class Reference	909
6.334.1 Detailed Description	911
6.334.2 Member Enumeration Documentation	911
6.334.2.1 FaceType	911
6.334.3 Constructor & Destructor Documentation	911
6.334.3.1 DConfigDlg() [1/2]	911
6.334.3.2 ~DConfigDlg()	911
6.334.3.3 DConfigDlg() [2/2]	911
6.334.4 Member Function Documentation	912
6.334.4.1 addActionButton()	912
6.334.4.2 addPage() [1/2]	912
6.334.4.3 addPage() [2/2]	912
6.334.4.4 addSubPage() [1/2]	913
6.334.4.5 addSubPage() [2/2]	913
6.334.4.6 button()	914
6.334.4.7 buttonBox() [1/2]	914
6.334.4.8 buttonBox() [2/2]	914
6.334.4.9 currentPage()	914
6.334.4.10 currentPageChanged	914
6.334.4.11 insertPage() [1/2]	915
6.334.4.12 insertPage() [2/2]	915
6.334.4.13 pageRemoved	916
6.334.4.14 pageWidget() [1/2]	916

---

6.334.4.15	<a href="#">pageWidget()</a> [2/2]	916
6.334.4.16	<a href="#">removePage()</a>	916
6.334.4.17	<a href="#">setButtonBox()</a>	917
6.334.4.18	<a href="#">setConfigGroup()</a>	917
6.334.4.19	<a href="#">setCurrentPage()</a>	917
6.334.4.20	<a href="#">setFaceType()</a>	917
6.334.4.21	<a href="#">setPageWidget()</a>	917
6.334.4.22	<a href="#">setStandardButtons()</a>	918
6.335	<a href="#">Digikam::DConfigDlgMngr Class Reference</a>	919
6.335.1	<a href="#">Detailed Description</a>	920
6.335.2	<a href="#">Constructor &amp; Destructor Documentation</a>	920
6.335.2.1	<a href="#">DConfigDlgMngr()</a>	920
6.335.2.2	<a href="#">~DConfigDlgMngr()</a>	921
6.335.3	<a href="#">Member Function Documentation</a>	921
6.335.3.1	<a href="#">addWidget()</a>	921
6.335.3.2	<a href="#">changedMap()</a>	921
6.335.3.3	<a href="#">getCustomProperty()</a>	921
6.335.3.4	<a href="#">getCustomPropertyChangedSignal()</a>	921
6.335.3.5	<a href="#">getUserProperty()</a>	922
6.335.3.6	<a href="#">getUserPropertyChangedSignal()</a>	922
6.335.3.7	<a href="#">hasChanged()</a>	922
6.335.3.8	<a href="#">init()</a>	922
6.335.3.9	<a href="#">initMaps()</a>	922
6.335.3.10	<a href="#">isDefault()</a>	922
6.335.3.11	<a href="#">parseChildren()</a>	923
6.335.3.12	<a href="#">property()</a>	923
6.335.3.13	<a href="#">propertyMap()</a>	923
6.335.3.14	<a href="#">setProperty()</a>	923
6.335.3.15	<a href="#">settingsChanged</a> [1/2]	923
6.335.3.16	<a href="#">settingsChanged</a> [2/2]	923
6.335.3.17	<a href="#">setupWidget()</a>	924
6.335.3.18	<a href="#">updateSettings</a>	924
6.335.3.19	<a href="#">updateWidgets</a>	924
6.335.3.20	<a href="#">updateWidgetsDefault</a>	924
6.335.3.21	<a href="#">widgetModified</a>	924
6.336	<a href="#">Digikam::DConfigDlgModel Class Reference</a>	925
6.336.1	<a href="#">Detailed Description</a>	926
6.336.2	<a href="#">Member Enumeration Documentation</a>	926
6.336.2.1	<a href="#">Role</a>	926
6.336.3	<a href="#">Constructor &amp; Destructor Documentation</a>	926
6.336.3.1	<a href="#">DConfigDlgModel()</a>	926
6.336.3.2	<a href="#">~DConfigDlgModel()</a>	926

---

6.337 Digikam::DConfigDlgModelPrivate Class Reference . . . . .	927
6.338 Digikam::DConfigDlgStackedWidget Class Reference . . . . .	928
6.339 Digikam::DConfigDlgTitle Class Reference . . . . .	929
6.339.1 Detailed Description . . . . .	930
6.339.2 Member Enumeration Documentation . . . . .	930
6.339.2.1 ImageAlignment . . . . .	930
6.339.2.2 MessageType . . . . .	931
6.339.3 Constructor & Destructor Documentation . . . . .	931
6.339.3.1 DConfigDlgTitle() . . . . .	931
6.339.4 Member Function Documentation . . . . .	931
6.339.4.1 autoHideTimeout() . . . . .	931
6.339.4.2 comment() . . . . .	932
6.339.4.3 pixmap() . . . . .	932
6.339.4.4 setAutoHideTimeout . . . . .	932
6.339.4.5 setBuddy() . . . . .	932
6.339.4.6 setComment . . . . .	933
6.339.4.7 setPixmap [1/4] . . . . .	933
6.339.4.8 setPixmap [2/4] . . . . .	933
6.339.4.9 setPixmap [3/4] . . . . .	934
6.339.4.10 setPixmap [4/4] . . . . .	934
6.339.4.11 setText [1/2] . . . . .	934
6.339.4.12 setText [2/2] . . . . .	935
6.339.4.13 setWidget() . . . . .	935
6.339.4.14 text() . . . . .	935
6.340 Digikam::DConfigDlgTitle::Private Class Reference . . . . .	935
6.340.1 Member Function Documentation . . . . .	936
6.340.1.1 iconTypeTolconName() . . . . .	936
6.341 Digikam::DConfigDlgView Class Reference . . . . .	936
6.341.1 Detailed Description . . . . .	938
6.341.2 Member Enumeration Documentation . . . . .	938
6.341.2.1 FaceType . . . . .	938
6.341.3 Constructor & Destructor Documentation . . . . .	939
6.341.3.1 DConfigDlgView() . . . . .	939
6.341.3.2 ~DConfigDlgView() . . . . .	939
6.341.4 Member Function Documentation . . . . .	939
6.341.4.1 createView() . . . . .	939
6.341.4.2 currentPage() . . . . .	939
6.341.4.3 currentPageChanged . . . . .	939
6.341.4.4 faceType() . . . . .	939
6.341.4.5 itemDelegate() . . . . .	940
6.341.4.6 model() . . . . .	940
6.341.4.7 setCurrentPage() . . . . .	940

6.341.4.8	setDefaultWidget()	940
6.341.4.9	setFaceType()	940
6.341.4.10	setItemDelegate()	940
6.341.4.11	setModel()	941
6.341.4.12	showPageHeader()	941
6.341.4.13	viewPosition()	941
6.342	Digikam::DConfigDlgViewPrivate Class Reference	942
6.342.1	Member Function Documentation	943
6.342.1.1	_k_dataChanged()	943
6.342.1.2	_k_modelChanged()	943
6.343	Digikam::DConfigDlgWdg Class Reference	943
6.343.1	Detailed Description	946
6.343.2	Constructor & Destructor Documentation	946
6.343.2.1	DConfigDlgWdg()	946
6.343.2.2	~DConfigDlgWdg()	946
6.343.3	Member Function Documentation	946
6.343.3.1	addPage() [1/2]	946
6.343.3.2	addPage() [2/2]	947
6.343.3.3	addSubPage() [1/2]	947
6.343.3.4	addSubPage() [2/2]	948
6.343.3.5	currentPage()	948
6.343.3.6	currentPageChanged	948
6.343.3.7	insertPage() [1/2]	949
6.343.3.8	insertPage() [2/2]	949
6.343.3.9	pageRemoved	950
6.343.3.10	pageToggled	950
6.343.3.11	removePage()	950
6.343.3.12	setCurrentPage()	951
6.344	Digikam::DConfigDlgWdgItem Class Reference	951
6.344.1	Detailed Description	952
6.344.2	Constructor & Destructor Documentation	952
6.344.2.1	DConfigDlgWdgItem() [1/2]	952
6.344.2.2	DConfigDlgWdgItem() [2/2]	953
6.344.2.3	~DConfigDlgWdgItem()	953
6.344.3	Member Function Documentation	953
6.344.3.1	changed	953
6.344.3.2	header()	953
6.344.3.3	icon()	953
6.344.3.4	isCheckedable()	954
6.344.3.5	isChecked()	954
6.344.3.6	isEnabled()	954
6.344.3.7	name()	954

6.344.3.8	setCheckable()	954
6.344.3.9	setChecked	954
6.344.3.10	setEnabled	954
6.344.3.11	setHeader()	955
6.344.3.12	setIcon()	955
6.344.3.13	setName()	955
6.344.3.14	toggled	955
6.344.3.15	widget()	955
6.344.4	Property Documentation	956
6.344.4.1	enabled	956
6.345	Digikam::DConfigDlgWdgModel Class Reference	956
6.345.1	Detailed Description	957
6.345.2	Constructor & Destructor Documentation	958
6.345.2.1	DConfigDlgWdgModel()	958
6.345.2.2	~DConfigDlgWdgModel()	958
6.345.3	Member Function Documentation	958
6.345.3.1	addPage() [1/2]	958
6.345.3.2	addPage() [2/2]	958
6.345.3.3	addSubPage() [1/2]	959
6.345.3.4	addSubPage() [2/2]	959
6.345.3.5	columnCount()	960
6.345.3.6	index()	960
6.345.3.7	insertPage() [1/2]	960
6.345.3.8	insertPage() [2/2]	961
6.345.3.9	item()	961
6.345.3.10	removePage()	962
6.345.3.11	toggled	962
6.346	Digikam::DConfigDlgWdgModelPrivate Class Reference	963
6.347	Digikam::DConfigDlgWdgPrivate Class Reference	964
6.348	Digikam::DCursorTracker Class Reference	965
6.348.1	Detailed Description	966
6.348.2	Member Function Documentation	966
6.348.2.1	setText()	966
6.349	Digikam::DDateEdit Class Reference	967
6.349.1	Detailed Description	968
6.349.2	Member Function Documentation	968
6.349.2.1	assignDate()	968
6.349.2.2	date()	968
6.349.2.3	dateChanged	969
6.349.2.4	isReadOnly()	969
6.349.2.5	setDate	969
6.349.2.6	setReadOnly()	969

---

6.349.2.7 setupKeywords()	969
6.350 Digikam::DDatePicker Class Reference	970
6.350.1 Detailed Description	972
6.350.2 Constructor & Destructor Documentation	972
6.350.2.1 DDatePicker() [1/2]	972
6.350.2.2 DDatePicker() [2/2]	972
6.350.2.3 ~DDatePicker()	972
6.350.3 Member Function Documentation	972
6.350.3.1 date()	972
6.350.3.2 dateChanged	973
6.350.3.3 dateEntered	973
6.350.3.4 dateSelected	973
6.350.3.5 dateTable()	973
6.350.3.6 fontSize()	973
6.350.3.7 hasCloseButton()	974
6.350.3.8 setCloseButton()	974
6.350.3.9 setDate()	974
6.350.3.10 setFontSize()	974
6.350.3.11 sizeHint()	974
6.350.3.12 tableClicked	975
6.351 Digikam::DDatePicker::Private Class Reference	975
6.351.1 Member Function Documentation	976
6.351.1.1 fillWeeksCombo()	976
6.352 Digikam::DDatePickerPopup Class Reference	976
6.352.1 Detailed Description	977
6.352.2 Constructor & Destructor Documentation	977
6.352.2.1 DDatePickerPopup()	977
6.352.3 Member Function Documentation	978
6.352.3.1 dateChanged	978
6.352.3.2 datePicker()	978
6.352.3.3 items()	978
6.353 Digikam::DDateTable Class Reference	979
6.353.1 Detailed Description	980
6.353.2 Member Function Documentation	981
6.353.2.1 aboutToShowContextMenu	981
6.353.2.2 date()	981
6.353.2.3 dateChanged [1/2]	981
6.353.2.4 dateChanged [2/2]	981
6.353.2.5 dateFromPos()	981
6.353.2.6 event()	982
6.353.2.7 mousePressEvent()	982
6.353.2.8 popupMenuEnabled()	982



6.353.2.9 posFromDate()	982
6.353.2.10 setCustomDatePainting()	982
6.353.2.11 setDate()	982
6.353.2.12 setFontSize()	983
6.353.2.13 setPopupMenuEnabled()	983
6.353.2.14 sizeHint()	983
6.353.2.15 tableClicked	983
6.353.2.16 unsetCustomDatePainting()	983
6.354 Digikam::DDateTable::Private Class Reference	984
6.354.1 Member Data Documentation	985
6.354.1.1 date	985
6.354.1.2 fontsize	985
6.354.1.3 maxCell	985
6.354.1.4 numDayColumns	986
6.354.1.5 numDaysThisMonth	986
6.354.1.6 numWeekRows	986
6.354.1.7 weekDayFirstOfMonth	986
6.355 Digikam::DDateTable::Private::DatePaintingMode Class Reference	986
6.356 Digikam::DDateTimeEdit Class Reference	987
6.356.1 Detailed Description	988
6.356.2 Constructor & Destructor Documentation	988
6.356.2.1 DDateTimeEdit()	988
6.356.2.2 ~DDateTimeEdit()	988
6.356.3 Member Function Documentation	988
6.356.3.1 dateTime()	988
6.356.3.2 dateTimeChanged	989
6.356.3.3 setDateTime()	989
6.357 Digikam::DDoubleNumInput Class Reference	989
6.358 Digikam::DDoubleSliderSpinBox Class Reference	991
6.358.1 Member Function Documentation	993
6.358.1.1 setInternalValue()	993
6.358.1.2 valueString()	993
6.359 Digikam::DefaultRenameParser Class Reference	994
6.360 Digikam::DefaultValueDialog Class Reference	995
6.361 Digikam::DefaultValueModifier Class Reference	996
6.361.1 Member Function Documentation	998
6.361.1.1 parseOperation()	998
6.362 Digikam::DefaultVersionNamingScheme Class Reference	999
6.362.1 Member Function Documentation	1000
6.362.1.1 baseName()	1000
6.362.1.2 directory()	1000
6.362.1.3 incrementedCounter()	1000

---

6.362.1.4 initialCounter() . . . . .	1000
6.362.1.5 intermediateDirectory() . . . . .	1001
6.362.1.6 intermediateFileName() . . . . .	1001
6.362.1.7 versionFileName() . . . . .	1001
6.363 Digikam::DeleteDialog Class Reference . . . . .	1002
6.364 Digikam::DeleteItem Class Reference . . . . .	1003
6.365 Digikam::DeleteItemList Class Reference . . . . .	1004
6.366 Digikam::DeleteJob Class Reference . . . . .	1005
6.367 Digikam::DeleteWidget Class Reference . . . . .	1006
6.368 Digikam::DeltaTime Class Reference . . . . .	1007
6.368.1 Detailed Description . . . . .	1007
6.368.2 Member Function Documentation . . . . .	1007
6.368.2.1 isNull() . . . . .	1007
6.369 Digikam::DetByClockPhotoButton Class Reference . . . . .	1008
6.370 Digikam::DetectionBenchmarker Class Reference . . . . .	1009
6.370.1 Member Function Documentation . . . . .	1011
6.370.1.1 result() . . . . .	1011
6.371 Digikam::DetectionWorker Class Reference . . . . .	1012
6.372 Digikam::DExpanderBox Class Reference . . . . .	1014
6.372.1 Member Function Documentation . . . . .	1015
6.372.1.1 addItem() . . . . .	1015
6.372.1.2 insertItem() . . . . .	1016
6.373 Digikam::DExpanderBoxExclusive Class Reference . . . . .	1017
6.373.1 Member Function Documentation . . . . .	1018
6.373.1.1 setIsToolBox() . . . . .	1018
6.374 Digikam::DFileDialog Class Reference . . . . .	1019
6.375 Digikam::DFileOperations Class Reference . . . . .	1020
6.375.1 Member Function Documentation . . . . .	1020
6.375.1.1 copyFile() . . . . .	1020
6.375.1.2 copyFiles() . . . . .	1021
6.375.1.3 copyFolderRecursively() . . . . .	1021
6.375.1.4 copyModificationTime() . . . . .	1021
6.375.1.5 findExecutable() . . . . .	1021
6.375.1.6 getUniqueFileUrl() . . . . .	1021
6.375.1.7 getUniqueFolderUrl() . . . . .	1021
6.375.1.8 localFileRename() . . . . .	1022
6.375.1.9 openFilesWithDefaultApplication() . . . . .	1022
6.375.1.10 openInFileManager() . . . . .	1022
6.375.1.11 removeAndCopyFile() . . . . .	1022
6.375.1.12 renameFile() . . . . .	1022
6.375.1.13 setModificationTime() . . . . .	1022
6.375.1.14 sidecarFiles() . . . . .	1023

6.376 Digikam::DFileSelector Class Reference . . . . .	1024
6.376.1 Detailed Description . . . . .	1025
6.377 Digikam::DFontProperties Class Reference . . . . .	1026
6.377.1 Member Enumeration Documentation . . . . .	1027
6.377.1.1 DisplayFlag . . . . .	1027
6.377.1.2 FontColumn . . . . .	1028
6.377.1.3 FontDiff . . . . .	1028
6.377.1.4 FontListCriteria . . . . .	1028
6.377.2 Constructor & Destructor Documentation . . . . .	1028
6.377.2.1 DFontProperties() . . . . .	1028
6.377.2.2 ~DFontProperties() . . . . .	1029
6.377.3 Member Function Documentation . . . . .	1029
6.377.3.1 backgroundColor() . . . . .	1029
6.377.3.2 color() . . . . .	1029
6.377.3.3 enableColumn() . . . . .	1029
6.377.3.4 font() . . . . .	1030
6.377.3.5 fontDiffFlags() . . . . .	1030
6.377.3.6 fontSelected . . . . .	1030
6.377.3.7 getFontList() . . . . .	1030
6.377.3.8 makeColumnVisible() . . . . .	1031
6.377.3.9 sampleText() . . . . .	1031
6.377.3.10 setBackgroundColor() . . . . .	1031
6.377.3.11 setColor() . . . . .	1031
6.377.3.12 setFont() . . . . .	1031
6.377.3.13 setSampleBoxVisible() . . . . .	1032
6.377.3.14 setSampleText() . . . . .	1032
6.377.3.15 setSizesRelative() . . . . .	1032
6.377.3.16 sizeHint() . . . . .	1032
6.377.3.17 sizelsRelative() . . . . .	1033
6.378 Digikam::DFontSelect Class Reference . . . . .	1033
6.379 Digikam::DGradientSlider Class Reference . . . . .	1035
6.380 Digikam::DHBox Class Reference . . . . .	1036
6.380.1 Detailed Description . . . . .	1037
6.381 Digikam::DHistoryView Class Reference . . . . .	1037
6.382 Digikam::DHueSaturationSelector Class Reference . . . . .	1038
6.382.1 Constructor & Destructor Documentation . . . . .	1040
6.382.1.1 DHueSaturationSelector() . . . . .	1040
6.382.1.2 ~DHueSaturationSelector() . . . . .	1040
6.382.2 Member Function Documentation . . . . .	1040
6.382.2.1 chooserMode() . . . . .	1040
6.382.2.2 colorValue() . . . . .	1040
6.382.2.3 drawContents() . . . . .	1041

---

6.382.2.4 drawPalette()	1041
6.382.2.5 hue()	1041
6.382.2.6 saturation()	1041
6.382.2.7 setChooserMode()	1041
6.382.2.8 setColorValue()	1042
6.382.2.9 setHue()	1042
6.382.2.10 setSaturation()	1042
6.382.2.11 updateContents()	1042
6.383 Digikam::DigikamApp Class Reference	1043
6.383.1 Member Function Documentation	1045
6.383.1.1 infoface()	1045
6.384 Digikam::DigikamApp::Private Class Reference	1046
6.385 Digikam::DigikamItemDelegate Class Reference	1049
6.385.1 Member Function Documentation	1053
6.385.1.1 updateRects()	1053
6.386 Digikam::DigikamItemDelegatePrivate Class Reference	1054
6.387 Digikam::DigikamItemView Class Reference	1057
6.387.1 Member Function Documentation	1063
6.387.1.1 activated()	1063
6.387.1.2 confirmFaces	1064
6.387.1.3 hasHiddenGroupedImages()	1064
6.387.1.4 ignoreFaces	1064
6.387.1.5 rejectFaces	1064
6.387.1.6 removeFaces	1064
6.387.1.7 setThumbnailSize()	1064
6.387.1.8 showContextMenu()	1065
6.387.1.9 showContextMenuOnInfo()	1065
6.387.1.10 slotSetupChanged()	1065
6.387.1.11 unknownFaces	1065
6.388 Digikam::DigikamItemView::Private Class Reference	1066
6.389 Digikam::DImageHistory Class Reference	1067
6.389.1 Member Function Documentation	1068
6.389.1.1 adjustReferredImages()	1068
6.389.1.2 clearReferredImages()	1068
6.389.1.3 entries()	1068
6.389.1.4 hasActions()	1068
6.389.1.5 isEmpty()	1068
6.389.1.6 isNull()	1068
6.389.1.7 isValid()	1069
6.389.1.8 moveCurrentReferredImage()	1069
6.389.1.9 operator<<() [1/2]	1069
6.389.1.10 operator<<() [2/2]	1069

---

6.389.1.11	<a href="#">purgePathFromReferredImages()</a>	1069
6.389.1.12	<a href="#">referredImages()</a>	1069
6.389.1.13	<a href="#">toXml()</a>	1070
6.390	<a href="#">Digikam::DImageHistory::Entry Class Reference</a>	1070
6.390.1	<a href="#">Member Data Documentation</a>	1070
6.390.1.1	<a href="#">action</a>	1070
6.391	<a href="#">Digikam::DImg Class Reference</a>	1071
6.391.1	<a href="#">Member Enumeration Documentation</a>	1075
6.391.1.1	<a href="#">FORMAT</a>	1075
6.391.1.2	<a href="#">PrepareMetadataFlag</a>	1075
6.391.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1076
6.391.2.1	<a href="#">DImg() [1/6]</a>	1076
6.391.2.2	<a href="#">DImg() [2/6]</a>	1077
6.391.2.3	<a href="#">DImg() [3/6]</a>	1077
6.391.2.4	<a href="#">DImg() [4/6]</a>	1077
6.391.2.5	<a href="#">DImg() [5/6]</a>	1077
6.391.2.6	<a href="#">DImg() [6/6]</a>	1077
6.391.3	<a href="#">Member Function Documentation</a>	1078
6.391.3.1	<a href="#">addAsReferredImage()</a>	1078
6.391.3.2	<a href="#">addCurrentUniqueImageld()</a>	1078
6.391.3.3	<a href="#">bitBlendImage()</a>	1078
6.391.3.4	<a href="#">bitBlendImageOnColor()</a>	1078
6.391.3.5	<a href="#">bitBltImage()</a>	1079
6.391.3.6	<a href="#">bitsDepth()</a>	1079
6.391.3.7	<a href="#">bytesDepth()</a>	1079
6.391.3.8	<a href="#">colorModelToString()</a>	1079
6.391.3.9	<a href="#">convertDepth()</a>	1079
6.391.3.10	<a href="#">convertToSixteenBit()</a>	1079
6.391.3.11	<a href="#">copy() [1/2]</a>	1080
6.391.3.12	<a href="#">copy() [2/2]</a>	1080
6.391.3.13	<a href="#">copyImageData()</a>	1080
6.391.3.14	<a href="#">copyMetaData()</a>	1080
6.391.3.15	<a href="#">copyQImage()</a>	1080
6.391.3.16	<a href="#">createHistoryImageld()</a>	1080
6.391.3.17	<a href="#">createImageUniqueId()</a>	1080
6.391.3.18	<a href="#">crop()</a>	1081
6.391.3.19	<a href="#">detach()</a>	1081
6.391.3.20	<a href="#">detectedFormat()</a>	1081
6.391.3.21	<a href="#">exifOrientation()</a>	1081
6.391.3.22	<a href="#">fileFormat()</a>	1081
6.391.3.23	<a href="#">fileOriginData()</a>	1082
6.391.3.24	<a href="#">fill()</a>	1082

---

6.391.3.25 format()	1082
6.391.3.26 getMetadata()	1082
6.391.3.27 getPixelColor()	1083
6.391.3.28 getUniqueHash()	1083
6.391.3.29 getUniqueHashVersion()	1083
6.391.3.30 hasTransparentPixels()	1083
6.391.3.31 imageSavedAs()	1084
6.391.3.32 isAnimatedImage()	1084
6.391.3.33 isReadOnly()	1084
6.391.3.34 lastSavedFilePath()	1084
6.391.3.35 loadItemInfo()	1084
6.391.3.36 operator=()	1084
6.391.3.37 operator==(())	1085
6.391.3.38 orientation()	1085
6.391.3.39 originalBitDepth()	1085
6.391.3.40 originalColorModel()	1085
6.391.3.41 originalFilePath()	1085
6.391.3.42 originalRatioSize()	1085
6.391.3.43 originalSize()	1085
6.391.3.44 prepareMetadataToSave()	1086
6.391.3.45 pureColorMask()	1086
6.391.3.46 putImageData() [1/2]	1086
6.391.3.47 putImageData() [2/2]	1086
6.391.3.48 rawDecodingSettings()	1086
6.391.3.49 removeAlphaChannel()	1087
6.391.3.50 reset()	1087
6.391.3.51 resetMetaData()	1087
6.391.3.52 resize()	1087
6.391.3.53 reverseExifRotate()	1087
6.391.3.54 reverseRotateAndFlip()	1087
6.391.3.55 rotateAndFlip()	1087
6.391.3.56 savedFormat()	1088
6.391.3.57 setHistoryBranchAfter()	1088
6.391.3.58 smoothScale()	1088
6.391.3.59 smoothScaleClipped()	1088
6.391.3.60 smoothScaleSection()	1089
6.391.3.61 stripImageData()	1089
6.391.3.62 transform()	1089
6.391.3.63 wasExifRotated()	1089
6.392 Digikam::DImgBuiltinFilter Class Reference	1089
6.392.1 Member Enumeration Documentation	1090
6.392.1.1 Type	1090

---

6.392.2 Constructor & Destructor Documentation	1090
6.392.2.1 DImgBuiltinFilter() [1/3]	1090
6.392.2.2 DImgBuiltinFilter() [2/3]	1091
6.392.2.3 DImgBuiltinFilter() [3/3]	1091
6.392.3 Member Function Documentation	1091
6.392.3.1 apply()	1091
6.392.3.2 createThreadedFilter()	1091
6.392.3.3 displayableName()	1091
6.392.3.4 filterAction()	1091
6.392.3.5 isSupported()	1092
6.392.3.6 isValid()	1092
6.392.3.7 reverseFilter()	1092
6.392.3.8 setAction()	1092
6.392.3.9 supportedVersions()	1092
6.393 Digikam::DImgChildItem Class Reference	1093
6.393.1 Constructor & Destructor Documentation	1095
6.393.1.1 DImgChildItem()	1095
6.393.2 Member Function Documentation	1095
6.393.2.1 boundingRect()	1095
6.393.2.2 originalRect()	1095
6.393.2.3 parentDImgItem()	1095
6.393.2.4 positionChanged	1095
6.393.2.5 positionOnImageChanged	1095
6.393.2.6 rect()	1096
6.393.2.7 relativeRect()	1096
6.393.2.8 setOriginalPos()	1096
6.393.2.9 setPos()	1096
6.393.2.10 setRectInSceneCoordinates()	1096
6.393.2.11 setRelativePos()	1096
6.394 Digikam::DImgFilterGenerator Class Reference	1097
6.394.1 Member Function Documentation	1098
6.394.1.1 createFilter()	1098
6.394.1.2 displayableName()	1098
6.394.1.3 isSupported()	1098
6.394.1.4 supportedFilters()	1098
6.394.1.5 supportedVersions()	1098
6.395 Digikam::DImgFilterManager Class Reference	1099
6.395.1 Member Function Documentation	1100
6.395.1.1 addGenerator()	1100
6.395.1.2 createFilter()	1100
6.395.1.3 displayableName()	1100
6.395.1.4 filterIcon()	1101

6.395.1.5 i18nDisplayName()	1101
6.395.1.6 isRawConversion()	1101
6.395.1.7 isSupported() [1/2]	1101
6.395.1.8 isSupported() [2/2]	1101
6.395.1.9 supportedFilters()	1101
6.395.1.10 supportedVersions()	1102
6.396 Digikam::DImgLoader Class Reference	1102
6.396.1 Member Enumeration Documentation	1103
6.396.1.1 LoadFlag	1103
6.396.2 Member Function Documentation	1104
6.396.2.1 checkAllocation()	1104
6.396.2.2 new_failureTolerant()	1104
6.397 Digikam::DImgLoaderObserver Class Reference	1105
6.397.1 Member Function Documentation	1106
6.397.1.1 continueQuery()	1106
6.397.1.2 granularity()	1106
6.397.1.3 progressInfo()	1106
6.398 Digikam::DImgLoaderSettings Class Reference	1106
6.398.1 Member Function Documentation	1107
6.398.1.1 parameters()	1107
6.398.1.2 setSettings()	1107
6.398.1.3 settings()	1107
6.398.1.4 signalSettingsChanged	1107
6.399 Digikam::DImgPreviewItem Class Reference	1108
6.399.1 Member Function Documentation	1110
6.399.1.1 userLoadingHint()	1110
6.400 Digikam::DImgStaticPriv Class Reference	1110
6.401 Digikam::DImgThreadedAnalyser Class Reference	1111
6.401.1 Constructor & Destructor Documentation	1114
6.401.1.1 DImgThreadedAnalyser()	1114
6.401.2 Member Function Documentation	1114
6.401.2.1 startAnalyse()	1114
6.402 Digikam::DImgThreadedFilter Class Reference	1114
6.402.1 Constructor & Destructor Documentation	1116
6.402.1.1 DImgThreadedFilter() [1/3]	1116
6.402.1.2 DImgThreadedFilter() [2/3]	1117
6.402.1.3 DImgThreadedFilter() [3/3]	1117
6.402.2 Member Function Documentation	1117
6.402.2.1 cancelFilter()	1117
6.402.2.2 cleanupFilter()	1117
6.402.2.3 filterAction()	1118
6.402.2.4 filterIdentifier()	1118



6.402.2.5 filterImage()	1118
6.402.2.6 finished	1118
6.402.2.7 initFilter()	1119
6.402.2.8 initSlave()	1119
6.402.2.9 modulateProgress()	1119
6.402.2.10 multithreadedSteps()	1119
6.402.2.11 parametersSuccessfullyRead()	1119
6.402.2.12 postProgress()	1120
6.402.2.13 progress	1120
6.402.2.14 run()	1120
6.402.2.15 setFilterVersion()	1120
6.402.2.16 setSlave()	1120
6.402.2.17 setupAndStartDirectly()	1120
6.402.2.18 setupFilter()	1121
6.402.2.19 started	1121
6.402.2.20 startFilter()	1121
6.402.2.21 startFilterDirectly()	1121
6.402.3 Member Data Documentation	1121
6.402.3.1 m_destImage	1121
6.402.3.2 m_master	1121
6.402.3.3 m_name	1121
6.402.3.4 m_orgImage	1122
6.402.3.5 m_progressBegin	1122
6.402.3.6 m_slave	1122
6.403 Digikam::DImgThreadedFilter::DefaultFilterAction< Filter > Class Template Reference	1123
6.403.1 Detailed Description	1125
6.403.2 Member Function Documentation	1125
6.403.2.1 supportOlderVersionIf()	1125
6.404 Digikam::DInfoInterface Class Reference	1126
6.404.1 Member Function Documentation	1128
6.404.1.1 albumChooser()	1128
6.404.1.2 currentSelectedItems()	1128
6.404.1.3 defaultUploadUrl()	1128
6.404.1.4 deleteImage()	1128
6.404.1.5 openSetupPage()	1128
6.404.1.6 passShortcutActionsToWidget()	1128
6.404.1.7 slotDateTimeForUrl()	1129
6.404.1.8 slotMetadataChangedForUrl()	1129
6.404.1.9 tagFilterModel()	1129
6.404.1.10 uploadWidget()	1129
6.405 Digikam::DIntNumInput Class Reference	1130
6.406 Digikam::DIntRangeBox Class Reference	1131

---

6.406.1 Member Function Documentation	1131
6.406.1.1 maxValue()	1131
6.406.1.2 minValue()	1132
6.406.1.3 setEnabled()	1132
6.406.1.4 setInterval()	1132
6.406.1.5 setRange()	1132
6.406.1.6 setSuffix()	1133
6.407 Digikam::DIO Class Reference	1133
6.407.1 Member Function Documentation	1134
6.407.1.1 copy()	1134
6.408 Digikam::DirectoryNameOption Class Reference	1135
6.408.1 Member Function Documentation	1136
6.408.1.1 parseOperation()	1136
6.409 Digikam::DisjointMetadata Class Reference	1137
6.409.1 Member Enumeration Documentation	1139
6.409.1.1 WriteMode	1139
6.409.2 Member Function Documentation	1139
6.409.2.1 changedFlags()	1139
6.409.2.2 colorLabel()	1139
6.409.2.3 colorLabelInterval()	1139
6.409.2.4 comments()	1140
6.409.2.5 dateTime()	1140
6.409.2.6 dateTimeChanged()	1140
6.409.2.7 dateTimeInterval()	1140
6.409.2.8 dateTimeStatus()	1140
6.409.2.9 keywords()	1140
6.409.2.10 metadataTemplate()	1140
6.409.2.11 pickLabel()	1141
6.409.2.12 pickLabelInterval()	1141
6.409.2.13 rating()	1141
6.409.2.14 ratingInterval()	1141
6.409.2.15 replaceColorLabel()	1141
6.409.2.16 setDateTime()	1141
6.409.2.17 tags()	1142
6.409.2.18 titles()	1142
6.409.2.19 willWriteMetadata()	1142
6.409.2.20 write()	1142
6.410 Digikam::DisjointMetadata::Private Class Reference	1143
6.411 Digikam::DisjointMetadataDataFields Class Reference	1145
6.411.1 Detailed Description	1146
6.411.2 Member Enumeration Documentation	1146
6.411.2.1 Status	1146

---

6.412 Digikam::DistortionFXFilter Class Reference	1147
6.412.1 Member Function Documentation	1150
6.412.1.1 filterAction()	1150
6.412.1.2 filterIdentifier()	1150
6.412.1.3 readParameters()	1150
6.413 Digikam::DItemDelegate Class Reference	1151
6.413.1 Member Function Documentation	1152
6.413.1.1 acceptsToolTip()	1152
6.413.1.2 gridSize()	1152
6.413.1.3 mouseMoved()	1153
6.413.1.4 setDefaultViewOptions()	1153
6.413.1.5 setThumbnailSize()	1153
6.414 Digikam::DItemDrag Class Reference	1154
6.414.1 Detailed Description	1154
6.415 Digikam::DItemInfo Class Reference	1155
6.415.1 Detailed Description	1156
6.416 Digikam::DItemsList Class Reference	1157
6.416.1 Member Function Documentation	1159
6.416.1.1 appendControlButtonsWidget()	1159
6.416.1.2 checkSelection()	1159
6.416.1.3 loadImagesFromCurrentAlbum()	1159
6.416.1.4 setControlButtonsPlacement()	1159
6.416.1.5 setIsLessThanHandler()	1159
6.417 Digikam::DItemsListView Class Reference	1160
6.418 Digikam::DItemsListViewItem Class Reference	1161
6.418.1 Member Function Documentation	1162
6.418.1.1 updateItemWidgets()	1162
6.419 Digikam::DItemToolTip Class Reference	1162
6.419.1 Member Function Documentation	1163
6.419.1.1 tipContents()	1163
6.420 Digikam::DKCamera Class Reference	1164
6.420.1 Member Function Documentation	1166
6.420.1.1 capture()	1166
6.420.1.2 getFreeSpace()	1166
6.420.1.3 getItemsInfoList()	1166
6.420.1.4 getPreview()	1166
6.421 Digikam::DLabelExpander Class Reference	1167
6.422 Digikam::DLineWidget Class Reference	1168
6.422.1 Detailed Description	1168
6.423 Digikam::DLogoAction Class Reference	1169
6.424 Digikam::DMessageBox Class Reference	1169
6.424.1 Member Function Documentation	1170

---

6.424.1.1 readMsgBoxShouldBeShown()	1170
6.424.1.2 saveMsgBoxShouldBeShown()	1170
6.424.1.3 showContinueCancel()	1170
6.424.1.4 showContinueCancelList()	1171
6.424.1.5 showContinueCancelWidget()	1171
6.424.1.6 showInformationList()	1171
6.424.1.7 showInformationWidget()	1171
6.424.1.8 showYesNo()	1172
6.424.1.9 showYesNoList()	1172
6.424.1.10 showYesNoWidget()	1172
6.425 Digikam::DMetadata Class Reference	1173
6.425.1 Member Enumeration Documentation	1183
6.425.1.1 VIDEOCOLORMODEL	1183
6.425.2 Member Function Documentation	1184
6.425.2.1 addToXmpTagStringBag()	1184
6.425.2.2 countryCodeMap()	1184
6.425.2.3 countryCodeMap2()	1184
6.425.2.4 getCameraSerialNumber()	1184
6.425.2.5 getCopyrightInformation()	1184
6.425.2.6 getIccProfile()	1184
6.425.2.7 getItemFacesMap()	1185
6.425.2.8 getLensDescription()	1185
6.425.2.9 getMetadataField()	1186
6.425.2.10 getMsecsInfo()	1186
6.425.2.11 getVideoInformation()	1186
6.425.2.12 getXmpKeywords()	1186
6.425.2.13 getXmpSubCategories()	1186
6.425.2.14 getXmpSubjects()	1186
6.425.2.15 load()	1186
6.425.2.16 loadUsingFFmpeg()	1187
6.425.2.17 loadUsingRawEngine()	1187
6.425.2.18 mSecTimeStamp()	1187
6.425.2.19 possibleValuesForEnumField()	1187
6.425.2.20 removeExifColorSpace()	1187
6.425.2.21 removeFromXmpTagStringBag()	1188
6.425.2.22 removeItemFacesMap()	1188
6.425.2.23 removeXmpKeywords()	1188
6.425.2.24 removeXmpSubCategories()	1188
6.425.2.25 removeXmpSubjects()	1188
6.425.2.26 setIccProfile()	1188
6.425.2.27 setItemFacesMap()	1188
6.425.2.28 setXmpKeywords()	1189

---

6.425.2.29 setXmpSubCategories()	1189
6.425.2.30 setXmpSubjects()	1189
6.425.2.31 valueToString()	1189
6.425.2.32 videoColorModelToString()	1189
6.426 Digikam::DMetadataSettings Class Reference	1190
6.426.1 Member Function Documentation	1190
6.426.1.1 instance()	1190
6.426.1.2 setSettings()	1191
6.426.1.3 settings()	1191
6.427 Digikam::DMetadataSettingsContainer Class Reference	1191
6.427.1 Detailed Description	1191
6.428 Digikam::DMetaInfolface Class Reference	1192
6.428.1 Member Function Documentation	1194
6.428.1.1 allAlbumItems()	1194
6.428.1.2 currentActiveItem()	1194
6.428.1.3 currentAlbumItems()	1194
6.428.1.4 currentSelectedItems()	1194
6.428.1.5 defaultUploadUrl()	1194
6.428.1.6 deleteImage()	1195
6.428.1.7 itemInfo()	1195
6.428.1.8 parseAlbumItemsRecursive()	1195
6.428.1.9 setItemInfo()	1195
6.428.1.10 slotDateTimeForUrl()	1195
6.428.1.11 slotMetadataChangedForUrl()	1195
6.428.1.12 supportAlbums()	1195
6.428.1.13 uploadUrl()	1196
6.428.1.14 uploadWidget()	1196
6.429 Digikam::DModelFactory Class Reference	1196
6.429.1 Detailed Description	1197
6.430 Digikam::DMultiTabBar Class Reference	1198
6.430.1 Detailed Description	1199
6.430.2 Member Enumeration Documentation	1199
6.430.2.1 TextStyle	1199
6.430.3 Member Function Documentation	1200
6.430.3.1 appendButton()	1200
6.430.3.2 appendTab()	1200
6.430.3.3 button()	1200
6.430.3.4 isTabRaised()	1200
6.430.3.5 position()	1201
6.430.3.6 removeButton()	1201
6.430.3.7 removeTab()	1201
6.430.3.8 setPosition()	1201

6.430.3.9	setStyle()	1201
6.430.3.10	setTab()	1201
6.430.3.11	tab()	1202
6.430.3.12	tabStyle()	1202
6.431	Digikam::DMultiTabBar::Private Class Reference	1202
6.432	Digikam::DMultiTabBarButton Class Reference	1203
6.432.1	Member Function Documentation	1204
6.432.1.1	signalClicked	1204
6.433	Digikam::DMultiTabBarFrame Class Reference	1205
6.433.1	Member Function Documentation	1206
6.433.1.1	contentsMouseEvent()	1206
6.434	Digikam::DMultiTabBarFrame::Private Class Reference	1206
6.435	Digikam::DMultiTabBarTab Class Reference	1207
6.435.1	Constructor & Destructor Documentation	1209
6.435.1.1	DMultiTabBarTab()	1209
6.435.2	Member Function Documentation	1209
6.435.2.1	setPosition	1209
6.435.2.2	setState	1209
6.435.2.3	setStyle	1209
6.436	Digikam::DMultiTabBarTab::Private Class Reference	1210
6.437	Digikam::DNGConvertSettings Class Reference	1210
6.438	Digikam::DNGSettings Class Reference	1211
6.439	Digikam::DNGWriter Class Reference	1212
6.439.1	Member Enumeration Documentation	1212
6.439.1.1	ConvertError	1212
6.439.1.2	JPEGPreview	1213
6.440	Digikam::DNGWriter::Private Class Reference	1213
6.440.1	Member Function Documentation	1214
6.440.1.1	debugExtractedRAWData()	1214
6.441	Digikam::DNGWriterHost Class Reference	1215
6.442	Digikam::DNNBaseDetectorModel Class Reference	1216
6.442.1	Member Function Documentation	1217
6.442.1.1	detectObjects() [1/2]	1217
6.442.1.2	detectObjects() [2/2]	1217
6.442.1.3	generateObjects() [1/2]	1218
6.442.1.4	generateObjects() [2/2]	1218
6.442.1.5	getInputImageSize()	1218
6.442.1.6	getPredefinedClasses()	1218
6.442.2	Member Data Documentation	1218
6.442.2.1	uiConfidenceThreshold	1218
6.443	Digikam::DNNFaceDetectorBase Class Reference	1219
6.443.1	Member Function Documentation	1220

---

6.443.1.1 selectBbox() . . . . .	1220
6.444 Digikam::DNNFaceDetectorSSD Class Reference . . . . .	1221
6.444.1 Member Function Documentation . . . . .	1222
6.444.1.1 detectFaces() . . . . .	1222
6.445 Digikam::DNNFaceDetectorYOLO Class Reference . . . . .	1223
6.445.1 Member Function Documentation . . . . .	1224
6.445.1.1 detectFaces() . . . . .	1224
6.446 Digikam::DNNFaceDetectorYuNet Class Reference . . . . .	1225
6.446.1 Member Function Documentation . . . . .	1226
6.446.1.1 detectFaces() . . . . .	1226
6.446.1.2 setFaceDetectionSize() . . . . .	1227
6.447 Digikam::DNNFaceExtractorBase Class Reference . . . . .	1227
6.447.1 Member Function Documentation . . . . .	1228
6.447.1.1 cosineDistance() . . . . .	1228
6.447.1.2 getThreshold() . . . . .	1228
6.447.1.3 loadModels() . . . . .	1228
6.447.1.4 vectortomat() . . . . .	1228
6.448 Digikam::DNNModelBase Class Reference . . . . .	1229
6.448.1 Member Function Documentation . . . . .	1230
6.448.1.1 getModelPath() . . . . .	1230
6.448.1.2 getThreshold() . . . . .	1230
6.449 Digikam::DNNModelConfig Class Reference . . . . .	1230
6.450 Digikam::DNNModelInfoContainer Class Reference . . . . .	1231
6.451 Digikam::DNNModelManager Class Reference . . . . .	1233
6.451.1 Member Function Documentation . . . . .	1233
6.451.1.1 getDownloadInformation() . . . . .	1233
6.451.1.2 getModel() . . . . .	1234
6.451.1.3 instance() . . . . .	1234
6.452 Digikam::DNNModelNet Class Reference . . . . .	1234
6.453 Digikam::DNNModelSFace Class Reference . . . . .	1236
6.454 Digikam::DNNModelYuNet Class Reference . . . . .	1237
6.455 Digikam::DNNOpenFaceExtractor Class Reference . . . . .	1239
6.455.1 Member Function Documentation . . . . .	1240
6.455.1.1 alignFace() [1/2] . . . . .	1240
6.455.1.2 alignFace() [2/2] . . . . .	1240
6.455.1.3 getFaceEmbedding() [1/2] . . . . .	1240
6.455.1.4 getFaceEmbedding() [2/2] . . . . .	1240
6.455.1.5 getThreshold() . . . . .	1240
6.455.1.6 loadModels() . . . . .	1241
6.456 Digikam::DNNResnetDetector Class Reference . . . . .	1241
6.456.1 Member Function Documentation . . . . .	1242
6.456.1.1 loadModels() . . . . .	1242

---

6.457 Digikam::DNNSFaceExtractor Class Reference	1243
6.457.1 Member Function Documentation	1244
6.457.1.1 alignFace() [1/2]	1244
6.457.1.2 alignFace() [2/2]	1244
6.457.1.3 getFaceEmbedding() [1/2]	1244
6.457.1.4 getFaceEmbedding() [2/2]	1244
6.457.1.5 getThreshold()	1244
6.457.1.6 loadModels()	1245
6.458 Digikam::DNNYoloDetector Class Reference	1245
6.458.1 Member Function Documentation	1246
6.458.1.1 loadModels()	1246
6.459 Digikam::DNotificationPopup Class Reference	1247
6.459.1 Detailed Description	1249
6.459.2 Member Enumeration Documentation	1249
6.459.2.1 PopupStyle	1249
6.459.3 Constructor & Destructor Documentation	1249
6.459.3.1 DNotificationPopup() [1/2]	1249
6.459.3.2 DNotificationPopup() [2/2]	1249
6.459.3.3 ~DNotificationPopup()	1250
6.459.4 Member Function Documentation	1250
6.459.4.1 anchor()	1250
6.459.4.2 autoDelete()	1250
6.459.4.3 clicked [1/2]	1250
6.459.4.4 clicked [2/2]	1250
6.459.4.5 defaultLocation()	1250
6.459.4.6 message() [1/14]	1251
6.459.4.7 message() [2/14]	1251
6.459.4.8 message() [3/14]	1251
6.459.4.9 message() [4/14]	1252
6.459.4.10 message() [5/14]	1252
6.459.4.11 message() [6/14]	1252
6.459.4.12 message() [7/14]	1252
6.459.4.13 message() [8/14]	1253
6.459.4.14 message() [9/14]	1253
6.459.4.15 message() [10/14]	1253
6.459.4.16 message() [11/14]	1254
6.459.4.17 message() [12/14]	1254
6.459.4.18 message() [13/14]	1254
6.459.4.19 message() [14/14]	1255
6.459.4.20 moveNear()	1255
6.459.4.21 positionSelf()	1255
6.459.4.22 setAnchor()	1255



---

6.459.4.23 setAutoDelete()	1256
6.459.4.24 setPopupStyle	1256
6.459.4.25 setTimeout	1256
6.459.4.26 setView() [1/3]	1256
6.459.4.27 setView() [2/3]	1256
6.459.4.28 setView() [3/3]	1257
6.459.4.29 show	1257
6.459.4.30 standardView()	1257
6.459.4.31 timeout()	1257
6.459.4.32 view()	1258
6.460 Digikam::DNotificationWidget Class Reference	1258
6.460.1 Detailed Description	1260
6.460.2 Member Enumeration Documentation	1260
6.460.2.1 MessageType	1260
6.460.3 Constructor & Destructor Documentation	1260
6.460.3.1 DNotificationWidget() [1/2]	1260
6.460.3.2 DNotificationWidget() [2/2]	1260
6.460.3.3 ~DNotificationWidget()	1260
6.460.4 Member Function Documentation	1260
6.460.4.1 addAction()	1260
6.460.4.2 animatedHide	1261
6.460.4.3 animatedShow	1261
6.460.4.4 animatedShowTemporized()	1261
6.460.4.5 clearAllActions()	1261
6.460.4.6 heightForWidth()	1261
6.460.4.7 hideAnimationFinished	1262
6.460.4.8 icon()	1262
6.460.4.9 isCloseButtonVisible()	1262
6.460.4.10 isHideAnimationRunning()	1262
6.460.4.11 isShowAnimationRunning()	1263
6.460.4.12 linkActivated	1263
6.460.4.13 linkHovered	1263
6.460.4.14 messageType()	1263
6.460.4.15 minimumSizeHint()	1264
6.460.4.16 removeAction()	1264
6.460.4.17 setCloseButtonVisible	1264
6.460.4.18 setIcon	1264
6.460.4.19 setMessageType	1265
6.460.4.20 setText	1265
6.460.4.21 setWordWrap	1265
6.460.4.22 showAnimationFinished	1265
6.460.4.23 sizeHint()	1266

---

6.460.4.24 text()	1266
6.460.4.25 wordWrap()	1266
6.461 Digikam::DNotificationWidget::Private Class Reference	1267
6.462 Digikam::DOnlineTranslator Class Reference	1268
6.462.1 Member Enumeration Documentation	1272
6.462.1.1 TranslationError	1272
6.462.2 Constructor & Destructor Documentation	1272
6.462.2.1 DOnlineTranslator()	1272
6.462.3 Member Function Documentation	1273
6.462.3.1 detectLanguage()	1273
6.462.3.2 engineName()	1273
6.462.3.3 error()	1273
6.462.3.4 errorString()	1273
6.462.3.5 fromRFC3066()	1273
6.462.3.6 isRunning()	1274
6.462.3.7 isSourceTranscriptionEnabled()	1274
6.462.3.8 isSourceTranslitEnabled()	1274
6.462.3.9 isSupportTranslation()	1274
6.462.3.10 isTranslationOptionsEnabled()	1274
6.462.3.11 isTranslationTranslitEnabled()	1275
6.462.3.12 language() [1/2]	1275
6.462.3.13 language() [2/2]	1275
6.462.3.14 languageCode()	1275
6.462.3.15 languageName()	1276
6.462.3.16 setEngineApiKey()	1276
6.462.3.17 setEngineUrl()	1276
6.462.3.18 setSourceTranscriptionEnabled()	1277
6.462.3.19 setSourceTranslitEnabled()	1277
6.462.3.20 setTranslationOptionsEnabled()	1277
6.462.3.21 setTranslationTranslitEnabled()	1277
6.462.3.22 signalFinished	1277
6.462.3.23 source()	1278
6.462.3.24 sourceLanguage()	1278
6.462.3.25 sourceLanguageName()	1278
6.462.3.26 sourceTranscription()	1278
6.462.3.27 sourceTranslit()	1278
6.462.3.28 supportedRFC3066()	1278
6.462.3.29 toJson()	1279
6.462.3.30 translate()	1279
6.462.3.31 translation()	1279
6.462.3.32 translationLanguage()	1279
6.462.3.33 translationLanguageName()	1279

6.462.3.34 translationOptions()	1280
6.462.3.35 translationTranslit()	1280
6.463 Digikam::DOnlineTranslator::Private Class Reference	1280
6.463.1 Member Data Documentation	1281
6.463.1.1 s_bingLanguageCodes	1281
6.463.1.2 s_googleLanguageCodes	1281
6.463.1.3 s_lingvaLanguageCodes	1282
6.463.1.4 s_rfc3066LanguageCodesBing	1282
6.463.1.5 s_rfc3066LanguageCodesGoogle	1282
6.463.1.6 s_rfc3066LanguageCodesLingva	1282
6.463.1.7 s_rfc3066LanguageCodesYandex	1282
6.463.1.8 s_yandexLanguageCodes	1283
6.464 Digikam::DOnlineTranslatorOption Struct Reference	1283
6.464.1 Detailed Description	1283
6.464.2 Member Function Documentation	1284
6.464.2.1 toJson()	1284
6.465 Digikam::DOnlineTts Class Reference	1284
6.465.1 Detailed Description	1285
6.465.2 Member Enumeration Documentation	1285
6.465.2.1 Emotion	1285
6.465.2.2 TtsError	1286
6.465.2.3 Voice	1287
6.465.3 Constructor & Destructor Documentation	1287
6.465.3.1 DOnlineTts()	1287
6.465.4 Member Function Documentation	1287
6.465.4.1 emotion()	1287
6.465.4.2 emotionCode()	1288
6.465.4.3 error()	1288
6.465.4.4 errorString()	1288
6.465.4.5 generateUrls()	1288
6.465.4.6 media()	1289
6.465.4.7 voice()	1289
6.465.4.8 voiceCode()	1289
6.466 Digikam::DOnlineTts::Private Class Reference	1290
6.466.1 Member Data Documentation	1290
6.466.1.1 s_emotionCodes	1290
6.466.1.2 s_voiceCodes	1290
6.467 Digikam::DownloadInfo Class Reference	1290
6.467.1 Member Data Documentation	1291
6.467.1.1 hash	1291
6.467.1.2 name	1291
6.467.1.3 path	1291

---

6.467.1.4 size	1291
6.468 Digikam::DownloadSettings Class Reference	1291
6.469 Digikam::DPixelsAliasFilter Class Reference	1292
6.469.1 Member Function Documentation	1292
6.469.1.1 pixelAntiAliasing()	1292
6.469.1.2 pixelAntiAliasing16()	1293
6.470 Digikam::DPlainTextEdit Class Reference	1293
6.470.1 Detailed Description	1294
6.470.2 Constructor & Destructor Documentation	1295
6.470.2.1 DPlainTextEdit() [1/3]	1295
6.470.2.2 DPlainTextEdit() [2/3]	1295
6.470.2.3 DPlainTextEdit() [3/3]	1295
6.470.2.4 ~DPlainTextEdit()	1295
6.470.3 Member Function Documentation	1295
6.470.3.1 acceptedCharacters()	1295
6.470.3.2 ignoredCharacters()	1295
6.470.3.3 isClearButtonEnabled()	1296
6.470.3.4 leftCharacters()	1296
6.470.3.5 returnPressed	1296
6.470.3.6 setCurrentLanguage()	1296
6.470.3.7 setLinesVisible()	1296
6.470.3.8 setMaxLength()	1296
6.470.3.9 spellCheckSettings()	1297
6.470.3.10 text()	1297
6.471 Digikam::DPlainTextEdit::Private Class Reference	1297
6.471.1 Member Function Documentation	1297
6.471.1.1 init()	1297
6.472 Digikam::DPlugin Class Reference	1298
6.472.1 Detailed Description	1299
6.472.2 Constructor & Destructor Documentation	1299
6.472.2.1 DPlugin()	1299
6.472.2.2 ~DPlugin()	1299
6.472.3 Member Function Documentation	1299
6.472.3.1 categories()	1299
6.472.3.2 cleanUp()	1300
6.472.3.3 count()	1300
6.472.3.4 extraAboutData()	1300
6.472.3.5 extraAboutDataRowTitles()	1300
6.472.3.6 extraAboutDataTitle()	1300
6.472.3.7 handbookChapter()	1300
6.472.3.8 handbookReference()	1300
6.472.3.9 handbookSection()	1301

---

6.472.3.10 hasVisibilityProperty()	1301
6.472.3.11 ifacelid()	1301
6.472.3.12 name()	1301
6.472.3.13 pluginAuthors()	1301
6.472.3.14 setShouldLoaded()	1301
6.472.3.15 setup()	1301
6.472.3.16 setVisible()	1302
6.472.3.17 shouldLoaded()	1302
6.472.3.18 version()	1302
6.473 Digikam::DPluginAboutDlg Class Reference	1302
6.474 Digikam::DPluginAction Class Reference	1303
6.474.1 Member Enumeration Documentation	1304
6.474.1.1 ActionCategory	1304
6.474.1.2 ActionType	1304
6.474.2 Member Function Documentation	1304
6.474.2.1 actionType()	1304
6.474.2.2 pluginId()	1304
6.474.2.3 setActionCategory()	1305
6.474.2.4 toString()	1305
6.474.2.5 xmlSection()	1305
6.475 Digikam::DPluginAuthor Class Reference	1305
6.475.1 Member Function Documentation	1305
6.475.1.1 toString()	1305
6.476 Digikam::DPluginBqm Class Reference	1306
6.476.1 Constructor & Destructor Documentation	1308
6.476.1.1 DPluginBqm()	1308
6.476.1.2 ~DPluginBqm()	1308
6.476.2 Member Function Documentation	1308
6.476.2.1 categories()	1308
6.476.2.2 count()	1308
6.476.2.3 findToolByName()	1308
6.476.2.4 hasVisibilityProperty()	1309
6.476.2.5 ifacelid()	1309
6.476.2.6 infoface()	1309
6.476.2.7 setVisible()	1309
6.476.2.8 tools()	1309
6.477 Digikam::DPluginConfView Class Reference	1310
6.477.1 Constructor & Destructor Documentation	1311
6.477.1.1 DPluginConfView()	1311
6.477.2 Member Function Documentation	1311
6.477.2.1 actived()	1311
6.477.2.2 apply()	1311

---

6.477.2.3	clearAll()	1311
6.477.2.4	count()	1311
6.477.2.5	filter()	1311
6.477.2.6	itemsVisible()	1311
6.477.2.7	itemsWithVisiblyProperty()	1312
6.477.2.8	selectAll()	1312
6.477.2.9	setFilter()	1312
6.477.2.10	signalSearchResult	1312
6.478	Digikam::DPluginConfViewBqm Class Reference	1313
6.478.1	Member Function Documentation	1314
6.478.1.1	loadPlugins()	1314
6.479	Digikam::DPluginConfViewDImg Class Reference	1315
6.479.1	Member Function Documentation	1316
6.479.1.1	loadPlugins()	1316
6.480	Digikam::DPluginConfViewEditor Class Reference	1317
6.480.1	Member Function Documentation	1318
6.480.1.1	loadPlugins()	1318
6.481	Digikam::DPluginConfViewGeneric Class Reference	1319
6.481.1	Member Function Documentation	1320
6.481.1.1	loadPlugins()	1320
6.482	Digikam::DPluginDialog Class Reference	1321
6.483	Digikam::DPluginDImg Class Reference	1322
6.483.1	Constructor & Destructor Documentation	1324
6.483.1.1	DPluginDImg()	1324
6.483.1.2	~DPluginDImg()	1324
6.483.2	Member Function Documentation	1324
6.483.2.1	canRead()	1324
6.483.2.2	canWrite()	1324
6.483.2.3	categories()	1324
6.483.2.4	count()	1324
6.483.2.5	exportWidget()	1325
6.483.2.6	extraAboutData()	1325
6.483.2.7	extraAboutDataRowTitles()	1325
6.483.2.8	extraAboutDataTitle()	1325
6.483.2.9	hasVisibilityProperty()	1325
6.483.2.10	ifacelid()	1325
6.483.2.11	loader()	1326
6.483.2.12	loaderName()	1326
6.483.2.13	previewSupported()	1326
6.483.2.14	setVisible()	1326
6.483.2.15	typeMimes()	1326
6.484	Digikam::DPluginEditor Class Reference	1327

---

6.484.1 Constructor & Destructor Documentation	1329
6.484.1.1 DPluginEditor()	1329
6.484.1.2 ~DPluginEditor()	1329
6.484.2 Member Function Documentation	1329
6.484.2.1 actions()	1329
6.484.2.2 categories()	1329
6.484.2.3 count()	1329
6.484.2.4 findActionByName()	1329
6.484.2.5 ifacelid()	1330
6.484.2.6 infolface()	1330
6.484.2.7 setVisible()	1330
6.485 Digikam::DPluginGeneric Class Reference	1331
6.485.1 Constructor & Destructor Documentation	1333
6.485.1.1 DPluginGeneric()	1333
6.485.1.2 ~DPluginGeneric()	1333
6.485.2 Member Function Documentation	1333
6.485.2.1 actions()	1333
6.485.2.2 categories()	1333
6.485.2.3 count()	1333
6.485.2.4 findActionByName()	1333
6.485.2.5 ifacelid()	1334
6.485.2.6 infolface()	1334
6.485.2.7 reactivateToolDialog()	1334
6.485.2.8 setVisible()	1334
6.486 Digikam::DPluginLoader Class Reference	1334
6.486.1 Detailed Description	1336
6.486.2 Member Function Documentation	1336
6.486.2.1 appendPluginToBlackList()	1336
6.486.2.2 appendPluginToWhiteList()	1337
6.486.2.3 canExport()	1337
6.486.2.4 canImport()	1337
6.486.2.5 cleanUp()	1337
6.486.2.6 configGroupName()	1337
6.486.2.7 exportWidget()	1337
6.486.2.8 init()	1338
6.486.2.9 instance()	1338
6.486.2.10 registerEditorPlugins()	1338
6.486.2.11 registerGenericPlugins()	1338
6.486.2.12 registerRawImportPlugins()	1338
6.487 Digikam::DPluginLoader::Private Class Reference	1338
6.487.1 Member Function Documentation	1339
6.487.1.1 appendPlugin()	1339

---

6.487.1.2 loadPlugins()	1339
6.487.1.3 pluginEntriesList()	1339
6.488 Digikam::DPluginRawImport Class Reference	1340
6.488.1 Constructor & Destructor Documentation	1342
6.488.1.1 DPluginRawImport()	1342
6.488.1.2 ~DPluginRawImport()	1342
6.488.2 Member Function Documentation	1342
6.488.2.1 categories()	1342
6.488.2.2 count()	1342
6.488.2.3 getRawProgram()	1342
6.488.2.4 ifacelid()	1342
6.488.2.5 run()	1343
6.488.2.6 setVisible()	1343
6.488.2.7 signalDecodedImage	1343
6.488.2.8 signalLoadRaw	1343
6.489 Digikam::DPluginSetup Class Reference	1343
6.490 Digikam::DPointSelect Class Reference	1344
6.490.1 Constructor & Destructor Documentation	1345
6.490.1.1 DPointSelect()	1345
6.490.2 Member Function Documentation	1346
6.490.2.1 contentsRect()	1346
6.490.2.2 drawContents()	1346
6.490.2.3 drawMarker()	1346
6.490.2.4 minimumSizeHint()	1346
6.490.2.5 setMarkerColor()	1346
6.490.2.6 setRange()	1347
6.490.2.7 setValues()	1347
6.490.2.8 setXValue()	1347
6.490.2.9 setYValue()	1347
6.490.2.10 valueChanged	1348
6.490.2.11 valuesFromPosition()	1348
6.490.2.12 xValue()	1348
6.490.2.13 yValue()	1348
6.491 Digikam::DPopupFrame Class Reference	1349
6.491.1 Constructor & Destructor Documentation	1350
6.491.1.1 DPopupFrame()	1350
6.491.1.2 ~DPopupFrame()	1350
6.491.2 Member Function Documentation	1350
6.491.2.1 close	1350
6.491.2.2 exec() [1/2]	1350
6.491.2.3 exec() [2/2]	1350
6.491.2.4 hideEvent()	1351



---

6.491.2.5 keyPressEvent()	1351
6.491.2.6 popup()	1351
6.491.2.7 resizeEvent()	1351
6.491.2.8 setMainWidget()	1351
6.492 Digikam::DPreviewImage Class Reference	1352
6.492.1 Member Function Documentation	1353
6.492.1.1 setSelectionArea()	1353
6.492.1.2 slotClearHighlight	1354
6.492.1.3 slotSetHighlightArea	1354
6.492.1.4 slotSetHighlightShown	1354
6.492.1.5 slotSetSelection	1354
6.493 Digikam::DPreviewManager Class Reference	1356
6.493.1 Member Function Documentation	1357
6.493.1.1 setSelectionArea()	1357
6.494 Digikam::DProgressDlg Class Reference	1358
6.495 Digikam::DProgressWdg Class Reference	1359
6.495.1 Member Function Documentation	1360
6.495.1.1 progressCompleted()	1360
6.495.1.2 progressScheduled()	1360
6.495.1.3 progressStatusChanged()	1360
6.495.1.4 progressThumbnailChanged()	1360
6.495.1.5 signalProgressCanceled	1361
6.496 Digikam::DragDropModelImplementation Class Reference	1361
6.496.1 Constructor & Destructor Documentation	1362
6.496.1.1 DragDropModelImplementation()	1362
6.496.2 Member Function Documentation	1362
6.496.2.1 dragDropFlags()	1362
6.496.2.2 dragDropFlagsV2()	1362
6.496.2.3 setDragDropHandler()	1363
6.496.2.4 supportedDropActions()	1363
6.497 Digikam::DragDropViewImplementation Class Reference	1364
6.497.1 Member Function Documentation	1365
6.497.1.1 asView()	1365
6.497.1.2 dragDropHandler()	1365
6.497.1.3 dragEnterEvent()	1365
6.497.1.4 mapIndexForDragDrop()	1365
6.497.1.5 pixmapForDrag()	1366
6.498 Digikam::DragHandle Class Reference	1366
6.498.1 Detailed Description	1366
6.499 Digikam::DRawDecoder Class Reference	1367
6.499.1 Constructor & Destructor Documentation	1368
6.499.1.1 DRawDecoder()	1368

---

6.499.1.2 ~DRawDecoder()	1369
6.499.2 Member Function Documentation	1369
6.499.2.1 cancel()	1369
6.499.2.2 checkToCancelWaitingData()	1369
6.499.2.3 decodeHalfRAWImage()	1369
6.499.2.4 decodeRAWImage()	1370
6.499.2.5 extractRAWData()	1370
6.499.2.6 librawUseGomp()	1370
6.499.2.7 librawVersion()	1370
6.499.2.8 loadEmbeddedPreview() [1/3]	1371
6.499.2.9 loadEmbeddedPreview() [2/3]	1371
6.499.2.10 loadEmbeddedPreview() [3/3]	1371
6.499.2.11 loadFullImage()	1371
6.499.2.12 loadHalfPreview() [1/3]	1371
6.499.2.13 loadHalfPreview() [2/3]	1372
6.499.2.14 loadHalfPreview() [3/3]	1372
6.499.2.15 loadRawPreview() [1/3]	1372
6.499.2.16 loadRawPreview() [2/3]	1372
6.499.2.17 loadRawPreview() [3/3]	1372
6.499.2.18 rawFileIdentify()	1373
6.499.2.19 rawFiles()	1373
6.499.2.20 rawFilesList()	1373
6.499.2.21 rawFilesVersion()	1373
6.499.2.22 setWaitingDataProgress()	1373
6.499.2.23 supportedCamera()	1373
6.499.3 Member Data Documentation	1374
6.499.3.1 m_cancel	1374
6.499.3.2 m_decoderSettings	1374
6.500 Digikam::DRawDecoder::Private Class Reference	1374
6.501 Digikam::DRawDecoderSettings Class Reference	1374
6.501.1 Member Enumeration Documentation	1375
6.501.1.1 DecodingQuality	1375
6.501.1.2 InputColorSpace	1376
6.501.1.3 NoiseReduction	1376
6.501.1.4 OutputColorSpace	1376
6.501.1.5 WhiteBalance	1376
6.501.2 Constructor & Destructor Documentation	1377
6.501.2.1 DRawDecoderSettings() [1/2]	1377
6.501.2.2 DRawDecoderSettings() [2/2]	1377
6.501.2.3 ~DRawDecoderSettings()	1377
6.501.3 Member Function Documentation	1377
6.501.3.1 operator==( )	1377

6.501.3.2 optimizeTimeLoading()	1377
6.501.4 Member Data Documentation	1377
6.501.4.1 autoBrightness	1377
6.501.4.2 blackPoint	1378
6.501.4.3 brightness	1378
6.501.4.4 customWhiteBalance	1378
6.501.4.5 dcbEnhanceFI	1378
6.501.4.6 dcbIterations	1378
6.501.4.7 deadPixelMap	1378
6.501.4.8 DontStretchPixels	1378
6.501.4.9 enableBlackPoint	1379
6.501.4.10 enableWhitePoint	1379
6.501.4.11 expoCorrection	1379
6.501.4.12 expoCorrectionHighlight	1379
6.501.4.13 expoCorrectionShift	1379
6.501.4.14 fixColorsHighlights	1379
6.501.4.15 halfSizeColorImage	1379
6.501.4.16 inputColorSpace	1380
6.501.4.17 inputProfile	1380
6.501.4.18 medianFilterPasses	1380
6.501.4.19 NRThreshold	1380
6.501.4.20 NRType	1380
6.501.4.21 outputColorSpace	1380
6.501.4.22 outputProfile	1380
6.501.4.23 RAWQuality	1381
6.501.4.24 RGBInterpolate4Colors	1381
6.501.4.25 sixteenBitsImage	1381
6.501.4.26 unclipColors	1381
6.501.4.27 whiteBalance	1381
6.501.4.28 whiteBalanceArea	1381
6.501.4.29 whitePoint	1381
6.502 Digikam::DRawDecoderWidget Class Reference	1382
6.502.1 Constructor & Destructor Documentation	1384
6.502.1.1 DRawDecoderWidget()	1384
6.502.2 Member Function Documentation	1384
6.502.2.1 readSettings()	1384
6.502.2.2 writeSettings()	1384
6.503 Digikam::DRawDecoding Class Reference	1385
6.503.1 Constructor & Destructor Documentation	1385
6.503.1.1 DRawDecoding() [1/2]	1385
6.503.1.2 DRawDecoding() [2/2]	1385
6.503.1.3 ~DRawDecoding()	1385

---

6.503.2 Member Function Documentation	1386
6.503.2.1 decodingSettingsToXml()	1386
6.503.2.2 operator==( )	1386
6.503.2.3 optimizeTimeLoading()	1386
6.503.2.4 postProcessingSettingsIsDirty()	1386
6.503.2.5 resetPostProcessingSettings()	1386
6.503.3 Member Data Documentation	1386
6.503.3.1 bcg	1386
6.503.3.2 curvesAdjust	1387
6.503.3.3 rawPrm	1387
6.503.3.4 wb	1387
6.504 Digikam::DRawInfo Class Reference	1387
6.504.1 Member Enumeration Documentation	1389
6.504.1.1 ImageOrientation	1389
6.504.2 Constructor & Destructor Documentation	1389
6.504.2.1 DRawInfo()	1389
6.504.2.2 ~DRawInfo()	1389
6.504.3 Member Data Documentation	1389
6.504.3.1 ambientAcceleration	1389
6.504.3.2 ambientElevationAngle	1389
6.504.3.3 ambientHumidity	1389
6.504.3.4 ambientPressure	1389
6.504.3.5 ambientTemperature	1390
6.504.3.6 ambientWaterDepth	1390
6.504.3.7 aperture	1390
6.504.3.8 baselineExposure	1390
6.504.3.9 blackPoint	1390
6.504.3.10 blackPointCh	1390
6.504.3.11 cameraColorMatrix1	1390
6.504.3.12 cameraMult	1390
6.504.3.13 colorKeys	1391
6.504.3.14 dateTime	1391
6.504.3.15 daylightMult	1391
6.504.3.16 description	1391
6.504.3.17 DNGVersion	1391
6.504.3.18 exposureIndex	1391
6.504.3.19 exposureProgram	1391
6.504.3.20 exposureTime	1391
6.504.3.21 filterPattern	1392
6.504.3.22 firmware	1392
6.504.3.23 flashUsed	1392
6.504.3.24 focalLength	1392

---

6.504.3.25 fullSize	1392
6.504.3.26 hasIccProfile	1392
6.504.3.27 iccData	1392
6.504.3.28 imageID	1392
6.504.3.29 imageSize	1393
6.504.3.30 isDecodable	1393
6.504.3.31 latitude	1393
6.504.3.32 leftMargin	1393
6.504.3.33 lensModel	1393
6.504.3.34 localizedCameraModel	1393
6.504.3.35 make	1393
6.504.3.36 meteringMode	1393
6.504.3.37 model	1394
6.504.3.38 orientation	1394
6.504.3.39 originalRawFileName	1394
6.504.3.40 outputSize	1394
6.504.3.41 owner	1394
6.504.3.42 pixelAspectRatio	1394
6.504.3.43 rawColors	1394
6.504.3.44 rawDataUniqueID	1395
6.504.3.45 rawImages	1395
6.504.3.46 sensitivity	1395
6.504.3.47 serialNumber	1395
6.504.3.48 software	1395
6.504.3.49 thumbnail	1395
6.504.3.50 thumbSize	1395
6.504.3.51 topMargin	1395
6.504.3.52 uniqueCameraModel	1396
6.504.3.53 whitePoint	1396
6.504.3.54 xmpData	1396
6.505 Digikam::DSaveSettingsWidget Class Reference	1396
6.506 Digikam::DSelectionItem Class Reference	1398
6.507 Digikam::DSelector Class Reference	1399
6.507.1 Detailed Description	1400
6.507.2 Member Function Documentation	1400
6.507.2.1 arrowDirection()	1400
6.507.2.2 contentsRect()	1401
6.507.2.3 drawArrow()	1401
6.507.2.4 drawContents()	1401
6.507.2.5 indent()	1401
6.507.2.6 setArrowDirection()	1401
6.507.2.7 setIndent()	1401

6.508 Digikam::DServiceInfo Class Reference	1402
6.509 Digikam::DServiceMenu Class Reference	1402
6.509.1 Member Function Documentation	1402
6.509.1.1 getIconFromService()	1402
6.509.1.2 runFiles() [1/2]	1402
6.509.1.3 runFiles() [2/2]	1403
6.509.1.4 servicesForOpenWith()	1403
6.510 Digikam::DSliderSpinBox Class Reference	1404
6.510.1 Member Function Documentation	1406
6.510.1.1 setInternalValue()	1406
6.510.1.2 valueString()	1406
6.511 Digikam::DSplashScreen Class Reference	1407
6.512 Digikam::DSqueezedClickLabel Class Reference	1408
6.513 Digikam::DTagListDrag Class Reference	1409
6.513.1 Detailed Description	1410
6.514 Digikam::DTextBrowser Class Reference	1410
6.515 Digikam::DTextEdit Class Reference	1411
6.515.1 Detailed Description	1412
6.515.2 Constructor & Destructor Documentation	1412
6.515.2.1 DTextEdit() [1/3]	1412
6.515.2.2 DTextEdit() [2/3]	1412
6.515.2.3 DTextEdit() [3/3]	1413
6.515.2.4 ~DTextEdit()	1413
6.515.3 Member Function Documentation	1413
6.515.3.1 acceptedCharacters()	1413
6.515.3.2 ignoredCharacters()	1413
6.515.3.3 isClearButtonEnabled()	1413
6.515.3.4 leftCharacters()	1413
6.515.3.5 returnPressed	1414
6.515.3.6 setCurrentLanguage()	1414
6.515.3.7 setLinesVisible()	1414
6.515.3.8 setMaxLength()	1414
6.515.3.9 spellCheckSettings()	1414
6.515.3.10 text()	1414
6.516 Digikam::DTextEdit::Private Class Reference	1415
6.516.1 Member Function Documentation	1415
6.516.1.1 init()	1415
6.517 Digikam::DTextEditClearButton Class Reference	1415
6.518 Digikam::DTextLabelName Class Reference	1416
6.519 Digikam::DTextLabelValue Class Reference	1418
6.520 Digikam::DTextList Class Reference	1419
6.521 Digikam::DToolTipStyleSheet Class Reference	1419

---

6.522 Digikam::DTrash Class Reference	1420
6.522.1 Member Function Documentation	1420
6.522.1.1 deleteDirRecursivley()	1420
6.522.1.2 deleteImage()	1421
6.522.1.3 extractJsonForItem()	1421
6.523 Digikam::DTrashItemInfo Class Reference	1421
6.524 Digikam::DTrashItemModel Class Reference	1422
6.524.1 Member Function Documentation	1423
6.524.1.1 append	1423
6.524.1.2 changeThumbSize()	1424
6.524.1.3 isEmpty()	1424
6.524.1.4 loadItemsForCollection()	1424
6.524.1.5 pixmapForItem()	1424
6.524.1.6 refreshThumbnails	1424
6.524.1.7 removeItems	1425
6.525 Digikam::DTrashItemsListingJob Class Reference	1426
6.526 Digikam::DuplicatesFinder Class Reference	1428
6.526.1 Constructor & Destructor Documentation	1431
6.526.1.1 DuplicatesFinder()	1431
6.527 Digikam::DuplicatesProgressObserver Class Reference	1431
6.527.1 Member Function Documentation	1432
6.527.1.1 imageProcessed()	1432
6.527.1.2 isCanceled()	1432
6.528 Digikam::DVBox Class Reference	1432
6.528.1 Detailed Description	1433
6.529 Digikam::DWItemDelegate Class Reference	1434
6.529.1 Detailed Description	1435
6.529.2 Constructor & Destructor Documentation	1435
6.529.2.1 DWItemDelegate()	1435
6.529.3 Member Function Documentation	1435
6.529.3.1 blockedEventTypes()	1435
6.529.3.2 createItemWidgets()	1436
6.529.3.3 focusedIndex()	1436
6.529.3.4 itemView()	1437
6.529.3.5 setBlockedEventTypes()	1437
6.529.3.6 updateItemWidgets()	1437
6.530 Digikam::DWItemDelegatePool Class Reference	1438
6.530.1 Constructor & Destructor Documentation	1438
6.530.1.1 DWItemDelegatePool()	1438
6.530.2 Member Function Documentation	1438
6.530.2.1 findWidgets()	1438
6.531 Digikam::DWItemDelegatePoolPrivate Class Reference	1439

---

6.532 Digikam::DWItemDelegatePrivate Class Reference . . . . .	1440
6.533 Digikam::DWizardDlg Class Reference . . . . .	1441
6.534 Digikam::DWizardPage Class Reference . . . . .	1442
6.535 Digikam::DWorkingPixmap Class Reference . . . . .	1443
6.535.1 Detailed Description . . . . .	1443
6.536 Digikam::DXmlGuiWindow Class Reference . . . . .	1444
6.536.1 Detailed Description . . . . .	1446
6.536.2 Member Function Documentation . . . . .	1446
6.536.2.1 allActions() . . . . .	1446
6.536.2.2 cleanupActions() . . . . .	1446
6.536.2.3 configFullScreenHideToolBarsEntry() . . . . .	1446
6.536.2.4 createFullScreenAction() . . . . .	1446
6.536.2.5 createHelpActions() . . . . .	1446
6.536.2.6 createSettingsActions() . . . . .	1446
6.536.2.7 createSidebarActions() . . . . .	1447
6.536.2.8 customizedFullScreenMode() . . . . .	1447
6.536.2.9 editKeyboardShortcuts() . . . . .	1447
6.536.2.10 fullScreenIsActive() . . . . .	1447
6.536.2.11 infolface() . . . . .	1447
6.536.2.12 readFullScreenSettings() . . . . .	1447
6.536.2.13 registerPluginsActions() . . . . .	1448
6.536.2.14 setConfigGroupName() . . . . .	1448
6.536.2.15 setFullScreenOptions() . . . . .	1448
6.536.2.16 setupIconTheme() . . . . .	1448
6.536.2.17 showSideBars() . . . . .	1448
6.536.2.18 showThumbBar() . . . . .	1448
6.536.2.19 thumbbarVisibility() . . . . .	1448
6.537 Digikam::DXmlGuiWindow::Private Class Reference . . . . .	1449
6.537.1 Member Data Documentation . . . . .	1449
6.537.1.1 dirtyMainToolBar . . . . .	1449
6.537.1.2 fsOptions . . . . .	1449
6.537.1.3 fullScreenAction . . . . .	1449
6.537.1.4 fullScreenBtn . . . . .	1450
6.537.1.5 fullScreenHideSideBars . . . . .	1450
6.537.1.6 fullScreenHideStatusBar . . . . .	1450
6.537.1.7 fullScreenHideThumbBar . . . . .	1450
6.537.1.8 fullScreenHideToolBars . . . . .	1450
6.537.1.9 fullScreenParent . . . . .	1450
6.537.1.10 menubarVisibility . . . . .	1450
6.537.1.11 statusbarVisibility . . . . .	1450
6.537.1.12 thumbbarVisibility . . . . .	1451
6.537.1.13 toolbarsVisibility . . . . .	1451



---

6.538 Digikam::DynamicLayout Class Reference . . . . .	1451
6.539 Digikam::DynamicThread Class Reference . . . . .	1452
6.539.1 Constructor & Destructor Documentation . . . . .	1453
6.539.1.1 DynamicThread() . . . . .	1453
6.539.1.2 ~DynamicThread() . . . . .	1453
6.539.2 Member Function Documentation . . . . .	1453
6.539.2.1 run() . . . . .	1453
6.539.2.2 runningFlag() . . . . .	1454
6.539.2.3 setPriority() . . . . .	1454
6.539.2.4 shutDown() . . . . .	1454
6.539.2.5 start() . . . . .	1454
6.539.2.6 starting . . . . .	1454
6.539.2.7 stop . . . . .	1454
6.539.2.8 threadMutex() . . . . .	1455
6.539.2.9 wait . . . . .	1455
6.540 Digikam::DZoomBar Class Reference . . . . .	1456
6.540.1 Member Enumeration Documentation . . . . .	1457
6.540.1.1 BarMode . . . . .	1457
6.541 Digikam::EditableSearchTreeView Class Reference . . . . .	1459
6.541.1 Detailed Description . . . . .	1463
6.541.2 Constructor & Destructor Documentation . . . . .	1463
6.541.2.1 EditableSearchTreeView() . . . . .	1463
6.541.2.2 ~EditableSearchTreeView() . . . . .	1463
6.541.3 Member Function Documentation . . . . .	1463
6.541.3.1 addCustomContextMenuActions() . . . . .	1463
6.541.3.2 contextMenuTitle() . . . . .	1464
6.541.3.3 handleCustomContextMenuAction() . . . . .	1464
6.542 Digikam::EditorCore Class Reference . . . . .	1465
6.542.1 Member Function Documentation . . . . .	1467
6.542.1.1 convertToPixmap() . . . . .	1467
6.542.1.2 getImgSelection() . . . . .	1468
6.542.1.3 rotate90() . . . . .	1468
6.542.1.4 setUndoImg() . . . . .	1468
6.543 Digikam::EditorCore::Private Class Reference . . . . .	1468
6.544 Digikam::EditorCore::Private::FileToSave Class Reference . . . . .	1469
6.545 Digikam::EditorStackView Class Reference . . . . .	1470
6.546 Digikam::EditorTool Class Reference . . . . .	1472
6.546.1 Member Function Documentation . . . . .	1474
6.546.1.1 init() . . . . .	1474
6.546.1.2 setInitPreview() . . . . .	1474
6.547 Digikam::EditorTooliface Class Reference . . . . .	1475
6.548 Digikam::EditorToolSettings Class Reference . . . . .	1476

6.549 Digikam::EditorToolThreaded Class Reference . . . . .	1478
6.549.1 Member Function Documentation . . . . .	1481
6.549.1.1 analyser() . . . . .	1481
6.549.1.2 deleteFilterInstance() . . . . .	1481
6.549.1.3 filter() . . . . .	1481
6.549.1.4 renderingMode() . . . . .	1481
6.549.1.5 setProgressMessage() . . . . .	1481
6.549.1.6 slotAnalyserStarted . . . . .	1481
6.549.1.7 slotFilterStarted . . . . .	1481
6.549.1.8 slotProgress . . . . .	1482
6.550 Digikam::EditorWindow Class Reference . . . . .	1482
6.550.1 Member Function Documentation . . . . .	1487
6.550.1.1 registerExtraPluginsActions() . . . . .	1487
6.550.1.2 saveDestinationUrl() . . . . .	1487
6.550.1.3 toggleZoomActions() . . . . .	1488
6.551 Digikam::EditorWindow::Private Class Reference . . . . .	1488
6.552 Digikam::EffectMngr Class Reference . . . . .	1489
6.552.1 Member Enumeration Documentation . . . . .	1490
6.552.1.1 EffectType . . . . .	1490
6.553 Digikam::EffectMngr::Private Class Reference . . . . .	1490
6.554 Digikam::EffectPreview Class Reference . . . . .	1491
6.555 Digikam::Ellipsoid Class Reference . . . . .	1491
6.555.1 Detailed Description . . . . .	1492
6.555.2 Constructor & Destructor Documentation . . . . .	1492
6.555.2.1 Ellipsoid() . . . . .	1492
6.555.3 Member Function Documentation . . . . .	1493
6.555.3.1 CLARKE_1866() . . . . .	1493
6.555.3.2 createEllipsoid() . . . . .	1493
6.555.3.3 createFlattenedSphere() . . . . .	1493
6.555.3.4 eccentricity() . . . . .	1494
6.555.3.5 GRS80() . . . . .	1494
6.555.3.6 INTERNATIONAL_1924() . . . . .	1494
6.555.3.7 inverseFlattening() . . . . .	1494
6.555.3.8 isIvfDefinitive() . . . . .	1494
6.555.3.9 isSphere() . . . . .	1495
6.555.3.10 orthodromicDistance() . . . . .	1495
6.555.3.11 radiusOfCurvature() . . . . .	1495
6.555.3.12 semiMajorAxis() . . . . .	1496
6.555.3.13 semiMinorAxis() . . . . .	1496
6.555.3.14 SPHERE() . . . . .	1496
6.555.3.15 WGS84() . . . . .	1496
6.555.4 Member Data Documentation . . . . .	1496

---

6.555.4.1 m_inverseFlattening	1496
6.555.4.2 m_ivfDefinitive	1497
6.555.4.3 m_semiMajorAxis	1497
6.555.4.4 m_semiMinorAxis	1497
6.556 Digikam::EmbossFilter Class Reference	1498
6.556.1 Member Function Documentation	1501
6.556.1.1 filterAction()	1501
6.556.1.2 filterIdentifier()	1501
6.556.1.3 readParameters()	1501
6.557 Digikam::EmptyDTrashItemsJob Class Reference	1502
6.558 Digikam::EmptyImageListProvider Class Reference	1504
6.558.1 Member Function Documentation	1505
6.558.1.1 atEnd()	1505
6.558.1.2 image()	1505
6.558.1.3 images()	1505
6.558.1.4 proceed()	1505
6.558.1.5 setImages()	1505
6.558.1.6 setUnpairedImages()	1505
6.558.1.7 size()	1505
6.559 Digikam::EqualizeFilter Class Reference	1506
6.559.1 Member Function Documentation	1509
6.559.1.1 filterAction()	1509
6.559.1.2 filterIdentifier()	1509
6.559.1.3 readParameters()	1509
6.560 Digikam::ExifMetaEngineMergeHelper Class Reference	1509
6.561 Digikam::ExifToolBinary Class Reference	1511
6.562 Digikam::ExifToolConfPanel Class Reference	1513
6.563 Digikam::ExifToolErrorView Class Reference	1514
6.564 Digikam::ExifToolListView Class Reference	1515
6.564.1 Member Function Documentation	1516
6.564.1.1 setGroupList()	1516
6.565 Digikam::ExifToolListViewGroup Class Reference	1516
6.566 Digikam::ExifToolListViewItem Class Reference	1517
6.567 Digikam::ExifToolLoadingView Class Reference	1518
6.568 Digikam::ExifToolParser Class Reference	1519
6.568.1 Member Typedef Documentation	1520
6.568.1.1 ExifToolData	1520
6.568.2 Member Function Documentation	1521
6.568.2.1 applyChanges() [1/2]	1521
6.568.2.2 applyChanges() [2/2]	1521
6.568.2.3 applyMetadataFile()	1522
6.568.2.4 changeTimestamps()	1522

6.568.2.5 copyTags()	1522
6.568.2.6 exifToolAvailable()	1522
6.568.2.7 load()	1523
6.568.2.8 loadChunk()	1523
6.568.2.9 readableFormats()	1523
6.568.2.10 setOutputStream()	1523
6.568.2.11 tagsDatabase()	1523
6.568.2.12 tagsDbToOrderedMap()	1523
6.568.2.13 translateTags()	1523
6.568.2.14 translationsList()	1524
6.568.2.15 version()	1524
6.568.2.16 writableFormats()	1524
6.569 Digikam::ExifToolParser::Private Class Reference	1524
6.569.1 Member Function Documentation	1525
6.569.1.1 actionString()	1525
6.570 Digikam::ExifToolProcess Class Reference	1525
6.570.1 Member Enumeration Documentation	1526
6.570.1.1 Action	1526
6.570.1.2 CopyTagsSource	1527
6.570.1.3 ResultStatus	1527
6.570.1.4 TranslateTagsOps	1527
6.570.1.5 WritingTagsMode	1528
6.570.2 Constructor & Destructor Documentation	1528
6.570.2.1 ExifToolProcess()	1528
6.570.2.2 ~ExifToolProcess()	1528
6.570.3 Member Function Documentation	1528
6.570.3.1 command()	1528
6.570.3.2 exifToolAvailable()	1529
6.570.3.3 exifToolError()	1529
6.570.3.4 exifToolErrorString()	1529
6.570.3.5 exifToolsBusy()	1529
6.570.3.6 getExifToolResult()	1529
6.570.3.7 initExifTool()	1529
6.570.3.8 setExifToolProgram()	1529
6.570.3.9 shutDownExifTool()	1530
6.570.3.10 waitForExifToolResult()	1530
6.571 Digikam::ExifToolProcess::Private Class Reference	1530
6.572 Digikam::ExifToolProcess::Private::Command Class Reference	1531
6.573 Digikam::ExifToolProcess::Result Class Reference	1532
6.574 Digikam::ExifToolThread Class Reference	1532
6.574.1 Member Function Documentation	1533
6.574.1.1 run()	1533

---

6.575 Digikam::ExifToolWidget Class Reference	1533
6.576 Digikam::ExifWidget Class Reference	1535
6.576.1 Member Function Documentation	1537
6.576.1.1 getMetadataTitle()	1537
6.576.1.2 getTagDescription()	1537
6.576.1.3 getTagTitle()	1537
6.576.1.4 loadFromURL()	1537
6.577 Digikam::ExposureDetector Class Reference	1538
6.577.1 Member Function Documentation	1539
6.577.1.1 detect()	1539
6.578 Digikam::ExposureSettingsContainer Class Reference	1539
6.578.1 Member Data Documentation	1539
6.578.1.1 exposureIndicatorMode	1539
6.579 Digikam::FaceClassifier Class Reference	1540
6.579.1 Member Function Documentation	1541
6.579.1.1 loadTrainingData()	1541
6.579.1.2 predict() [1/2]	1541
6.579.1.3 predict() [2/2]	1541
6.579.1.4 retrain()	1542
6.580 Digikam::FaceClassifierBase Class Reference	1542
6.581 Digikam::FaceDb Class Reference	1543
6.581.1 Member Function Documentation	1544
6.581.1.1 clearDNNTraining()	1544
6.581.1.2 insertFaceVector()	1544
6.581.1.3 integrityCheck()	1544
6.581.1.4 removeFaceVector() [1/2]	1545
6.581.1.5 removeFaceVector() [2/2]	1546
6.581.1.6 trainData()	1546
6.581.1.7 vacuum()	1546
6.582 Digikam::FaceDb::Private Class Reference	1546
6.583 Digikam::FaceDbAccess Class Reference	1547
6.583.1 Constructor & Destructor Documentation	1547
6.583.1.1 FaceDbAccess()	1547
6.583.2 Member Function Documentation	1547
6.583.2.1 setLastError()	1547
6.584 Digikam::FaceDbAccessUnlock Class Reference	1547
6.584.1 Constructor & Destructor Documentation	1548
6.584.1.1 FaceDbAccessUnlock()	1548
6.585 Digikam::FaceDbBackend Class Reference	1548
6.585.1 Member Function Documentation	1551
6.585.1.1 initSchema()	1551
6.586 Digikam::FaceDbOperationGroup Class Reference	1551

---

6.586.1 Detailed Description	1551
6.586.2 Constructor & Destructor Documentation	1551
6.586.2.1 FaceDbOperationGroup() [1/2]	1551
6.586.2.2 FaceDbOperationGroup() [2/2]	1552
6.586.3 Member Function Documentation	1552
6.586.3.1 allowLift()	1552
6.586.3.2 lift()	1552
6.586.3.3 resetTime()	1552
6.587 Digikam::FaceDbSchemaUpdater Class Reference	1552
6.588 Digikam::FaceDetector Class Reference	1553
6.588.1 Constructor & Destructor Documentation	1553
6.588.1.1 FaceDetector()	1553
6.588.2 Member Function Documentation	1553
6.588.2.1 detectFaces() [1/2]	1553
6.588.2.2 detectFaces() [2/2]	1554
6.588.2.3 recommendedImageSize()	1554
6.588.2.4 setParameter()	1554
6.589 Digikam::FaceGroup Class Reference	1555
6.589.1 Constructor & Destructor Documentation	1557
6.589.1.1 FaceGroup()	1557
6.589.2 Member Function Documentation	1557
6.589.2.1 aboutToSetInfo	1557
6.589.2.2 addFace	1557
6.589.2.3 closestItem()	1557
6.589.2.4 hasUnconfirmed()	1557
6.589.2.5 markAllAsIgnored	1558
6.589.2.6 rejectAll	1558
6.589.2.7 setAutoSuggest()	1558
6.589.2.8 setInfo	1558
6.589.2.9 setShowOnHover()	1558
6.589.2.10 setVisible	1558
6.590 Digikam::FaceGroup::Private Class Reference	1559
6.590.1 Member Data Documentation	1559
6.590.1.1 MaxFaceListSize	1559
6.590.1.2 MaxMouseDistance	1559
6.591 Digikam::Faceltem Class Reference	1560
6.592 Digikam::FaceltemRetriever Class Reference	1563
6.593 Digikam::FacePipeline Class Reference	1564
6.593.1 Member Enumeration Documentation	1566
6.593.1.1 FilterMode	1566
6.593.1.2 WriteMode	1566
6.593.2 Member Function Documentation	1567

6.593.2.1 addManually . . . . .	1567
6.593.2.2 cancel() . . . . .	1567
6.593.2.3 confirm . . . . .	1567
6.593.2.4 editRegion . . . . .	1567
6.593.2.5 editTag . . . . .	1567
6.593.2.6 plugDatabaseFilter() . . . . .	1568
6.593.2.7 process [1/2] . . . . .	1568
6.593.2.8 process [2/2] . . . . .	1568
6.593.2.9 remove . . . . .	1568
6.593.2.10 setPriority() . . . . .	1569
6.593.2.11 shutDown() . . . . .	1569
6.593.2.12 train . . . . .	1569
6.594 Digikam::FacePipeline::Private Class Reference . . . . .	1570
6.595 Digikam::FacePipelineBase Class Reference . . . . .	1572
6.595.1 Member Enumeration Documentation . . . . .	1574
6.595.1.1 FilterMode . . . . .	1574
6.595.1.2 WriteMode . . . . .	1575
6.595.2 Member Function Documentation . . . . .	1575
6.595.2.1 enqueue() . . . . .	1575
6.596 Digikam::FacePipelineDetect Class Reference . . . . .	1576
6.596.1 Member Function Documentation . . . . .	1579
6.596.1.1 addMoreWorkers() . . . . .	1579
6.596.1.2 classifier() . . . . .	1579
6.596.1.3 extractor() . . . . .	1579
6.596.1.4 finder() . . . . .	1579
6.596.1.5 loader() . . . . .	1579
6.596.1.6 start() . . . . .	1579
6.596.1.7 trainer() . . . . .	1579
6.596.1.8 writer() . . . . .	1580
6.597 Digikam::FacePipelineDetectRecognize Class Reference . . . . .	1580
6.597.1 Member Function Documentation . . . . .	1583
6.597.1.1 addMoreWorkers() . . . . .	1583
6.597.1.2 classifier() . . . . .	1583
6.597.1.3 extractor() . . . . .	1583
6.597.1.4 finder() . . . . .	1583
6.597.1.5 loader() . . . . .	1583
6.597.1.6 start() . . . . .	1583
6.597.1.7 trainer() . . . . .	1583
6.597.1.8 writer() . . . . .	1584
6.598 Digikam::FacePipelineEdit Class Reference . . . . .	1584
6.598.1 Member Function Documentation . . . . .	1587
6.598.1.1 addMoreWorkers() . . . . .	1587

---

6.598.1.2 classifier()	1587
6.598.1.3 extractor()	1587
6.598.1.4 finder()	1588
6.598.1.5 loader()	1588
6.598.1.6 start()	1588
6.598.1.7 trainer()	1588
6.598.1.8 writer()	1588
6.599 Digikam::FacePipelineExtendedPackage Class Reference	1589
6.600 Digikam::FacePipelineFaceTagsIface Class Reference	1591
6.600.1 Member Enumeration Documentation	1593
6.600.1.1 Role	1593
6.601 Digikam::FacePipelineFaceTagsIfaceList Class Reference	1594
6.602 Digikam::FacePipelinePackage Class Reference	1595
6.603 Digikam::FacePipelinePackageBase Class Reference	1596
6.604 Digikam::FacePipelineRecognize Class Reference	1598
6.604.1 Member Function Documentation	1601
6.604.1.1 addMoreWorkers()	1601
6.604.1.2 classifier()	1601
6.604.1.3 extractor()	1601
6.604.1.4 finder()	1601
6.604.1.5 loader()	1601
6.604.1.6 start()	1601
6.604.1.7 trainer()	1601
6.604.1.8 writer()	1602
6.605 Digikam::FacePipelineReset Class Reference	1602
6.605.1 Member Function Documentation	1605
6.605.1.1 addMoreWorkers()	1605
6.605.1.2 classifier()	1605
6.605.1.3 extractor()	1605
6.605.1.4 finder()	1605
6.605.1.5 loader()	1605
6.605.1.6 start()	1605
6.605.1.7 trainer()	1605
6.605.1.8 writer()	1606
6.606 Digikam::FacePipelineRetrain Class Reference	1606
6.606.1 Member Function Documentation	1609
6.606.1.1 addMoreWorkers()	1609
6.606.1.2 classifier()	1609
6.606.1.3 extractor()	1609
6.606.1.4 finder()	1609
6.606.1.5 loader()	1609
6.606.1.6 start()	1609



6.606.1.7 trainer()	1609
6.606.1.8 writer()	1610
6.607 Digikam::FacePreprocessor Class Reference	1610
6.608 Digikam::FacePreviewLoader Class Reference	1611
6.609 Digikam::FaceRejectionOverlay Class Reference	1616
6.609.1 Member Function Documentation	1619
6.609.1.1 checkIndex()	1619
6.609.1.2 createButton()	1619
6.609.1.3 setActive()	1619
6.609.1.4 updateButton()	1619
6.609.1.5 widgetEnterEvent()	1619
6.609.1.6 widgetLeaveEvent()	1620
6.610 Digikam::FaceRejectionOverlayButton Class Reference	1620
6.610.1 Member Function Documentation	1621
6.610.1.1 icon()	1621
6.610.1.2 sizeHint()	1622
6.610.1.3 updateToolTip()	1622
6.611 Digikam::FaceScanSettings Class Reference	1622
6.611.1 Member Enumeration Documentation	1623
6.611.1.1 AlreadyScannedHandling	1623
6.611.1.2 FaceDetectionModel	1624
6.611.1.3 FaceDetectionSize	1624
6.611.1.4 FaceRecognitionModel	1624
6.611.1.5 ScanTask	1624
6.611.2 Member Data Documentation	1625
6.611.2.1 detectAccuracy	1625
6.611.2.2 recognizeAccuracy	1625
6.612 Digikam::FaceScanWidget Class Reference	1626
6.612.1 Member Function Documentation	1627
6.612.1.1 doLoadState()	1627
6.612.1.2 doSaveState()	1627
6.613 Digikam::FaceScanWidget::Private Class Reference	1628
6.614 Digikam::FacesDetector Class Reference	1629
6.615 Digikam::FacesEngine Class Reference	1632
6.616 Digikam::FaceTags Class Reference	1635
6.616.1 Member Function Documentation	1635
6.616.1.1 allPersonNames()	1635
6.616.1.2 allPersonPaths()	1635
6.616.1.3 allPersonTags()	1636
6.616.1.4 applyTagIdentityMapping()	1636
6.616.1.5 ensureIsPerson()	1636
6.616.1.6 faceNameForTag()	1636

---

6.616.1.7	getOrCreateTagForIdentity()	1636
6.616.1.8	getOrCreateTagForPerson()	1636
6.616.1.9	isPerson()	1637
6.616.1.10	personParentTag()	1637
6.616.1.11	tagForPerson()	1637
6.617	Digikam::FaceTagsEditor Class Reference	1638
6.617.1	Member Function Documentation	1639
6.617.1.1	add()	1639
6.617.1.2	addNormalTag()	1640
6.617.1.3	changeRegion()	1640
6.617.1.4	changeSuggestedName()	1640
6.617.1.5	changeTag()	1640
6.617.1.6	confirmedEntry()	1640
6.617.1.7	confirmName()	1641
6.617.1.8	databaseFaces()	1641
6.617.1.9	faceCountForPersonInImage()	1641
6.617.1.10	getSuggestedNames()	1641
6.617.1.11	getTagRects()	1641
6.617.1.12	numberOfFaces()	1641
6.617.1.13	removeAllFaces()	1642
6.617.1.14	removeFace() [1/2]	1642
6.617.1.15	removeFace() [2/2]	1642
6.617.1.16	removeNormalTag()	1642
6.617.1.17	rotateFaces()	1642
6.617.1.18	unconfirmedEntry()	1642
6.617.1.19	unconfirmedFaceTagsIfaces()	1643
6.617.1.20	unconfirmedNameFaceTagsIfaces()	1643
6.618	Digikam::FaceTagsIface Class Reference	1644
6.618.1	Member Function Documentation	1646
6.618.1.1	attributeForType()	1646
6.618.1.2	attributesForFlags()	1646
6.618.1.3	fromListing()	1646
6.618.1.4	fromVariant()	1646
6.618.1.5	getAutodetectedPersonString()	1646
6.618.1.6	hash()	1646
6.618.1.7	removeFaceTraining()	1646
6.618.1.8	typeForAttribute()	1647
6.618.1.9	typeForId()	1647
6.619	Digikam::FaceUtils Class Reference	1648
6.619.1	Member Function Documentation	1650
6.619.1.1	addNormalTag()	1650
6.619.1.2	faceRectToDisplayRect()	1650

6.619.1.3 hasBeenScanned()	1650
6.619.1.4 markAsScanned()	1650
6.619.1.5 removeNormalTag()	1651
6.619.1.6 removeNormalTags()	1651
6.619.1.7 storeThumbnails()	1651
6.619.1.8 toFaceTagsIfaces()	1651
6.619.1.9 writeUnconfirmedResults()	1651
6.620 Digikam::FacialRecognitionWrapper Class Reference	1652
6.620.1 Member Function Documentation	1652
6.620.1.1 addIdentity()	1652
6.620.1.2 addIdentityAttributes()	1653
6.620.1.3 addIdentityDebug()	1653
6.620.1.4 allIdentities()	1653
6.620.1.5 clearAllTraining()	1653
6.620.1.6 clearTraining() [1/2]	1653
6.620.1.7 clearTraining() [2/2]	1653
6.620.1.8 deleteIdentities()	1653
6.620.1.9 deleteIdentity()	1654
6.620.1.10 findIdentity() [1/2]	1654
6.620.1.11 findIdentity() [2/2]	1654
6.620.1.12 integrityCheck()	1654
6.620.1.13 recognizeFaces()	1654
6.620.1.14 setParameter()	1654
6.620.1.15 train() [1/2]	1655
6.620.1.16 train() [2/2]	1655
6.620.1.17 vacuum()	1655
6.621 Digikam::FacialRecognitionWrapper::Private Class Reference	1655
6.621.1 Member Function Documentation	1656
6.621.1.1 findByAttributes()	1656
6.621.1.2 identityContains()	1656
6.622 Digikam::FFmpegBinary Class Reference	1657
6.623 Digikam::FFmpegConfigHelper Class Reference	1659
6.623.1 Member Function Documentation	1659
6.623.1.1 getAudioCodecsProperties()	1659
6.623.1.2 getExtensionsProperties()	1660
6.623.1.3 getVideoCodecsProperties()	1660
6.624 Digikam::FFmpegLauncher Class Reference	1661
6.624.1 Member Function Documentation	1662
6.624.1.1 encodeFrames()	1662
6.624.1.2 setSettings()	1662
6.624.1.3 soundTrackLength()	1662
6.624.1.4 supportedCodecs()	1663

---

6.624.1.5 supportedFormats()	1663
6.625 Digikam::FieldQueryBuilder Class Reference	1663
6.626 Digikam::FileActionItemInfoList Class Reference	1664
6.627 Digikam::FileActionMngr Class Reference	1666
6.627.1 Member Function Documentation	1667
6.627.1.1 transform	1667
6.628 Digikam::FileActionMngr::Private Class Reference	1668
6.629 Digikam::FileActionMngrDatabaseWorker Class Reference	1670
6.629.1 Member Function Documentation	1672
6.629.1.1 applyMetadata()	1672
6.629.1.2 assignColorLabel()	1672
6.629.1.3 assignPickLabel()	1672
6.629.1.4 assignRating()	1672
6.629.1.5 assignTags()	1673
6.629.1.6 copyAttributes()	1673
6.629.1.7 editGroup()	1673
6.629.1.8 removeTags()	1673
6.629.1.9 setExifOrientation()	1673
6.630 Digikam::FileActionMngrFileWorker Class Reference	1674
6.630.1 Member Function Documentation	1676
6.630.1.1 transform()	1676
6.630.1.2 writeMetadata()	1676
6.630.1.3 writeMetadataToFiles()	1676
6.630.1.4 writeOrientationToFiles()	1676
6.631 Digikam::FileActionProgress Class Reference	1677
6.632 Digikam::FileActionProgressItemContainer Class Reference	1680
6.633 Digikam::FileActionProgressItemCreator Class Reference	1681
6.634 Digikam::FilePropertiesOption Class Reference	1682
6.634.1 Member Function Documentation	1683
6.634.1.1 parseOperation()	1683
6.635 Digikam::FileReadLocker Class Reference	1684
6.636 Digikam::FileReadWriteLockKey Class Reference	1684
6.637 Digikam::FileSaveConflictBox Class Reference	1685
6.638 Digikam::FileSaveOptionsBox Class Reference	1686
6.638.1 Member Enumeration Documentation	1686
6.638.1.1 FORMAT	1686
6.638.2 Constructor & Destructor Documentation	1687
6.638.2.1 FileSaveOptionsBox()	1687
6.638.2.2 ~FileSaveOptionsBox()	1687
6.638.3 Member Function Documentation	1687
6.638.3.1 discoverFormat()	1687
6.639 Digikam::FileSaveOptionsDlg Class Reference	1688

6.640 Digikam::FilesDownloader Class Reference . . . . .	1688
6.641 Digikam::FileWorkerInterface Class Reference . . . . .	1689
6.642 Digikam::FileWriteLocker Class Reference . . . . .	1691
6.643 Digikam::FilmContainer Class Reference . . . . .	1691
6.644 Digikam::FilmContainer::ListItem Class Reference . . . . .	1692
6.645 Digikam::FilmContainer::Private Class Reference . . . . .	1693
6.646 Digikam::FilmFilter Class Reference . . . . .	1694
6.646.1 Member Function Documentation . . . . .	1697
6.646.1.1 filterAction() . . . . .	1697
6.646.1.2 filterIdentifier() . . . . .	1697
6.646.1.3 readParameters() . . . . .	1697
6.647 Digikam::FilmFilter::Private Class Reference . . . . .	1697
6.648 Digikam::FilmGrainContainer Class Reference . . . . .	1697
6.649 Digikam::FilmGrainFilter Class Reference . . . . .	1698
6.649.1 Constructor & Destructor Documentation . . . . .	1701
6.649.1.1 FilmGrainFilter() . . . . .	1701
6.649.2 Member Function Documentation . . . . .	1701
6.649.2.1 filterAction() . . . . .	1701
6.649.2.2 filterIdentifier() . . . . .	1701
6.649.2.3 readParameters() . . . . .	1701
6.650 Digikam::FilmGrainSettings Class Reference . . . . .	1702
6.651 Digikam::FilmProfile Class Reference . . . . .	1702
6.652 Digikam::Filter Class Reference . . . . .	1703
6.653 Digikam::FilterAction Class Reference . . . . .	1704
6.653.1 Member Enumeration Documentation . . . . .	1705
6.653.1.1 Category . . . . .	1705
6.653.1.2 Flag . . . . .	1706
6.653.2 Member Function Documentation . . . . .	1706
6.653.2.1 description() . . . . .	1706
6.653.2.2 hasParameters() . . . . .	1706
6.653.2.3 identifier() . . . . .	1706
6.653.2.4 parameter() [1/2] . . . . .	1707
6.653.2.5 parameter() [2/2] . . . . .	1707
6.653.2.6 version() . . . . .	1707
6.654 Digikam::FilterActionFilter Class Reference . . . . .	1708
6.654.1 Constructor & Destructor Documentation . . . . .	1711
6.654.1.1 FilterActionFilter() . . . . .	1711
6.654.2 Member Function Documentation . . . . .	1711
6.654.2.1 appliedFilterActions() . . . . .	1711
6.654.2.2 completelyApplied() . . . . .	1711
6.654.2.3 filterAction() . . . . .	1712
6.654.2.4 filterIdentifier() . . . . .	1712

---

6.654.2.5 filterImage()	1712
6.654.2.6 isComplexAction()	1712
6.654.2.7 isReproducible()	1712
6.654.2.8 isSupported()	1712
6.654.2.9 readParameters()	1712
6.654.2.10 setContinueOnError()	1713
6.654.2.11 setFilterActions()	1713
6.655 Digikam::FiltersHistoryWidget Class Reference	1713
6.656 Digikam::FilterSideBarWidget Class Reference	1714
6.656.1 Detailed Description	1716
6.656.2 Constructor & Destructor Documentation	1716
6.656.2.1 FilterSideBarWidget()	1716
6.656.2.2 ~FilterSideBarWidget()	1716
6.656.3 Member Function Documentation	1717
6.656.3.1 doLoadState()	1717
6.656.3.2 doSaveState()	1717
6.656.3.3 setConfigGroup()	1717
6.656.3.4 signalTagFilterChanged	1717
6.656.3.5 slotResetFilters	1718
6.657 Digikam::FilterStatusBar Class Reference	1718
6.658 Digikam::FindDuplicatesAlbum Class Reference	1719
6.659 Digikam::FindDuplicatesAlbumItem Class Reference	1720
6.659.1 Member Function Documentation	1721
6.659.1.1 calculateInfos()	1721
6.659.1.2 itemCount()	1721
6.660 Digikam::FindDuplicatesView Class Reference	1721
6.661 Digikam::FingerPrintsGenerator Class Reference	1723
6.661.1 Constructor & Destructor Documentation	1726
6.661.1.1 FingerPrintsGenerator()	1726
6.661.2 Member Function Documentation	1726
6.661.2.1 setUseMultiCoreCPU()	1726
6.662 Digikam::FingerprintsTask Class Reference	1727
6.663 Digikam::FirstRunDlg Class Reference	1728
6.664 Digikam::FocusPoint Class Reference	1729
6.664.1 Member Enumeration Documentation	1729
6.664.1.1 TypePoint	1729
6.664.2 Constructor & Destructor Documentation	1730
6.664.2.1 FocusPoint()	1730
6.664.3 Member Function Documentation	1730
6.664.3.1 getRectBySize()	1730
6.664.3.2 operator=()	1730
6.664.3.3 setCenterPosition()	1730

---

6.664.3.4 setType()	1730
6.665 Digikam::FocusPointGroup Class Reference	1731
6.665.1 Member Function Documentation	1732
6.665.1.1 setInfo	1732
6.665.1.2 setVisible	1732
6.666 Digikam::FocusPointGroup::Private Class Reference	1733
6.667 Digikam::FocusPointItem Class Reference	1734
6.668 Digikam::FocusPointsExtractor Class Reference	1737
6.668.1 Member Typedef Documentation	1738
6.668.1.1 ListAFPoints	1738
6.669 Digikam::FocusPointsWriter Class Reference	1738
6.670 Digikam::FrameOsd Class Reference	1738
6.670.1 Member Function Documentation	1739
6.670.1.1 insertMessageOsdToFrame()	1739
6.670.1.2 insertOsdToFrame()	1739
6.670.1.3 populateOSD()	1739
6.670.1.4 printComments()	1739
6.670.1.5 printTags()	1739
6.671 Digikam::FrameOsdSettings Class Reference	1740
6.671.1 Member Function Documentation	1740
6.671.1.1 readSettings()	1740
6.671.2 Member Data Documentation	1740
6.671.2.1 osdFont	1740
6.672 Digikam::FrameOsdWidget Class Reference	1741
6.673 Digikam::FrameUtils Class Reference	1741
6.674 Digikam::FreeRotationContainer Class Reference	1741
6.675 Digikam::FreeRotationFilter Class Reference	1743
6.675.1 Member Function Documentation	1746
6.675.1.1 filterAction()	1746
6.675.1.2 filterIdentifier()	1746
6.675.1.3 readParameters()	1746
6.676 Digikam::FreeRotationSettings Class Reference	1746
6.677 Digikam::FreeSpaceToolTip Class Reference	1748
6.677.1 Member Function Documentation	1749
6.677.1.1 repositionRect()	1749
6.677.1.2 tipContents()	1749
6.678 Digikam::FreeSpaceWidget Class Reference	1750
6.679 Digikam::FullObjectDetection Class Reference	1751
6.680 Digikam::FullScreenSettings Class Reference	1752
6.681 Digikam::FuzzySearchSideBarWidget Class Reference	1753
6.681.1 Member Function Documentation	1755
6.681.1.1 applySettings()	1755

6.681.1.2 changeAlbumFromHistory()	1755
6.681.1.3 doLoadState()	1755
6.681.1.4 doSaveState()	1755
6.681.1.5 getCaption()	1755
6.681.1.6 getIcon()	1756
6.681.1.7 setActive()	1756
6.682 Digikam::FuzzySearchView Class Reference	1757
6.682.1 Member Function Documentation	1759
6.682.1.1 doLoadState()	1759
6.682.1.2 doSaveState()	1759
6.682.1.3 setConfigGroup()	1759
6.683 Digikam::FuzzySearchView::Private Class Reference	1759
6.684 Digikam::GeoCoordinates Class Reference	1760
6.685 Digikam::GeodeticCalculator Class Reference	1761
6.685.1 Constructor & Destructor Documentation	1762
6.685.1.1 GeodeticCalculator()	1762
6.685.2 Member Function Documentation	1763
6.685.2.1 azimuth()	1763
6.685.2.2 checkAzimuth()	1763
6.685.2.3 checkLatitude()	1763
6.685.2.4 checkLongitude()	1764
6.685.2.5 checkOrthodromicDistance() [1/2]	1764
6.685.2.6 checkOrthodromicDistance() [2/2]	1764
6.685.2.7 computeDestinationPoint()	1764
6.685.2.8 computeDirection()	1764
6.685.2.9 destinationGeographicPoint()	1765
6.685.2.10 ellipsoid()	1765
6.685.2.11 meridianArcLength()	1765
6.685.2.12 meridianArcLengthRadians()	1765
6.685.2.13 orthodromicDistance()	1766
6.685.2.14 setDestinationGeographicPoint()	1766
6.685.2.15 setDirection()	1766
6.685.2.16 setStartingGeographicPoint()	1767
6.685.3 Member Data Documentation	1767
6.685.3.1 a01	1767
6.685.3.2 fo	1767
6.685.3.3 m_A	1767
6.685.3.4 m_destinationValid	1767
6.685.3.5 m_directionValid	1768
6.685.3.6 m_distance	1768
6.685.3.7 m_eccentricitySquared	1768
6.685.3.8 m_ellipsoid	1768



6.685.3.9 m_lat1	1768
6.685.3.10 m_lat2	1768
6.685.3.11 m_maxOrthodromicDistance	1768
6.685.3.12 m_semiMajorAxis	1769
6.685.3.13 m_semiMinorAxis	1769
6.685.3.14 m_TOLERANCE_0	1769
6.685.3.15 m_TOLERANCE_CHECK	1769
6.685.3.16 T1	1769
6.686 Digikam::GeoDragDropHandler Class Reference	1770
6.687 Digikam::GeofaceCluster Class Reference	1770
6.688 Digikam::GeofaceGlobalObject Class Reference	1771
6.689 Digikam::GeofaceInternalWidgetInfo Class Reference	1773
6.689.1 Detailed Description	1774
6.690 Digikam::GeofaceSharedData Class Reference	1774
6.690.1 Member Function Documentation	1775
6.690.1.1 hasRegionSelection()	1775
6.691 Digikam::GeolocationFilter Class Reference	1776
6.692 Digikam::GeolocationSettings Class Reference	1777
6.692.1 Member Function Documentation	1778
6.692.1.1 applySettingsToWidget()	1778
6.692.1.2 instance()	1778
6.692.1.3 mainMarbleWidget()	1778
6.692.1.4 registerWidget()	1778
6.692.1.5 setSettings()	1778
6.692.1.6 settings()	1778
6.692.1.7 unregisterWidget()	1779
6.693 Digikam::GeolocationSettingsContainer Class Reference	1779
6.693.1 Detailed Description	1779
6.694 Digikam::GeoModelHelper Class Reference	1780
6.694.1 Detailed Description	1781
6.694.2 Member Function Documentation	1781
6.694.2.1 bestRepresentativeIndexFromList()	1781
6.694.2.2 itemCoordinates()	1782
6.694.2.3 itemIcon()	1782
6.694.2.4 model()	1782
6.694.2.5 onIndicesClicked()	1782
6.694.2.6 pixmapFromRepresentativeIndex()	1783
6.694.2.7 selectionModel()	1783
6.695 Digikam::GeoPluginAboutDlg Class Reference	1783
6.696 Digikam::GPCamera Class Reference	1784
6.696.1 Detailed Description	1786
6.696.2 Member Function Documentation	1786

---

6.696.2.1 cameraAbout()	1786
6.696.2.2 cameraDriverType()	1787
6.696.2.3 cameraManual()	1787
6.696.2.4 cameraMD5ID()	1787
6.696.2.5 cameraSummary()	1787
6.696.2.6 cancel()	1787
6.696.2.7 capture()	1787
6.696.2.8 deleteItem()	1787
6.696.2.9 doConnect()	1788
6.696.2.10 downloadItem()	1788
6.696.2.11 getFolders()	1788
6.696.2.12 getFreeSpace()	1788
6.696.2.13 getItemInfo()	1788
6.696.2.14 getItemsInfoList()	1788
6.696.2.15 getMetadata()	1789
6.696.2.16 getPreview()	1789
6.696.2.17 getThumbnail()	1789
6.696.2.18 setLockItem()	1789
6.696.2.19 uploadItem()	1789
6.697 Digikam::GPSBookmarkModelHelper Class Reference	1790
6.697.1 Member Function Documentation	1791
6.697.1.1 itemCoordinates()	1791
6.697.1.2 itemFlags()	1792
6.697.1.3 itemIcon()	1792
6.697.1.4 model()	1792
6.697.1.5 modelFlags()	1792
6.697.1.6 selectionModel()	1792
6.697.1.7 snapItemsTo()	1793
6.698 Digikam::GPSBookmarkOwner Class Reference	1793
6.699 Digikam::GPSCorrelatorWidget Class Reference	1794
6.700 Digikam::GPSDataContainer Class Reference	1795
6.701 Digikam::GPSDBJobInfo Class Reference	1796
6.702 Digikam::GPSDBJobsThread Class Reference	1798
6.702.1 Member Function Documentation	1799
6.702.1.1 GPSListing()	1799
6.703 Digikam::GPSGeofaceModelHelper Class Reference	1800
6.703.1 Member Function Documentation	1802
6.703.1.1 bestRepresentativeIndexFromList()	1802
6.703.1.2 itemCoordinates()	1802
6.703.1.3 model()	1802
6.703.1.4 modelFlags()	1802
6.703.1.5 onIndicesMoved()	1802

---

6.703.1.6 pixmapFromRepresentativeIndex()	1802
6.703.1.7 selectionModel()	1803
6.704 Digikam::GPSItemContainer Class Reference	1803
6.704.1 Member Function Documentation	1805
6.704.1.1 getTagList()	1805
6.704.1.2 isTagListDirty()	1805
6.704.1.3 loadImageData()	1805
6.704.1.4 restoreRGTagList()	1805
6.704.1.5 saveChanges()	1805
6.704.1.6 setTagList()	1805
6.704.1.7 writeLocations()	1806
6.704.1.8 writeTagsToXmp()	1806
6.705 Digikam::GPSItemDelegate Class Reference	1806
6.706 Digikam::GPSItemInfo Class Reference	1807
6.707 Digikam::GPSItemInfoSorter Class Reference	1808
6.708 Digikam::GPSItemList Class Reference	1809
6.709 Digikam::GPSItemListContextMenu Class Reference	1810
6.710 Digikam::GPSItemListDragDropHandler Class Reference	1812
6.710.1 Member Function Documentation	1812
6.710.1.1 createMimeData()	1812
6.711 Digikam::GPSItemModel Class Reference	1813
6.712 Digikam::GPSItemSortProxyModel Class Reference	1814
6.713 Digikam::GPSJob Class Reference	1815
6.714 Digikam::GPSLinkItemSelectionModel Class Reference	1817
6.714.1 Detailed Description	1818
6.715 Digikam::GPSMarkerTiler Class Reference	1818
6.715.1 Constructor & Destructor Documentation	1821
6.715.1.1 GPSMarkerTiler()	1821
6.715.2 Member Function Documentation	1821
6.715.2.1 bestRepresentativeIndexFromList()	1821
6.715.2.2 getGlobalGroupState()	1821
6.715.2.3 getTile()	1822
6.715.2.4 getTileGroupState()	1822
6.715.2.5 getTileMarkerCount()	1822
6.715.2.6 getTileRepresentativeMarker()	1822
6.715.2.7 getTileSelectedCount()	1822
6.715.2.8 indicesEqual()	1823
6.715.2.9 onIndicesClicked()	1823
6.715.2.10 pixmapFromRepresentativeIndex()	1823
6.715.2.11 prepareTiles()	1823
6.715.2.12 regenerateTiles()	1824
6.715.2.13 setActive()	1824

---

6.715.2.14 setPositiveFilterIsActive()	1824
6.715.2.15 slotNewModelData	1824
6.715.2.16 tileNew()	1824
6.716 Digikam::GPSModelIndexProxyMapper Class Reference	1825
6.716.1 Detailed Description	1826
6.716.2 Member Function Documentation	1826
6.716.2.1 mapLeftToRight()	1826
6.716.2.2 mapRightToLeft()	1826
6.716.2.3 mapSelectionLeftToRight()	1826
6.716.2.4 mapSelectionRightToLeft()	1826
6.716.3 Property Documentation	1827
6.716.3.1 isConnected	1827
6.717 Digikam::GPSSearchSideBarWidget Class Reference	1828
6.717.1 Member Function Documentation	1830
6.717.1.1 applySettings()	1830
6.717.1.2 changeAlbumFromHistory()	1830
6.717.1.3 doLoadState()	1830
6.717.1.4 doSaveState()	1830
6.717.1.5 getCaption()	1830
6.717.1.6 getIcon()	1831
6.717.1.7 setActive()	1831
6.718 Digikam::GPSSearchView Class Reference	1832
6.718.1 Constructor & Destructor Documentation	1833
6.718.1.1 GPSSearchView()	1833
6.718.2 Member Function Documentation	1834
6.718.2.1 doLoadState()	1834
6.718.2.2 doSaveState()	1834
6.718.2.3 setActive()	1834
6.718.2.4 setConfigGroup()	1834
6.719 Digikam::GPSUndoCommand Class Reference	1835
6.720 Digikam::GPSUndoCommand::UndoInfo Class Reference	1836
6.721 Digikam::Graph< VertexProperties, EdgeProperties > Class Template Reference	1837
6.721.1 Detailed Description	1840
6.721.2 Member Typedef Documentation	1841
6.721.2.1 graph_traits	1841
6.721.3 Member Enumeration Documentation	1841
6.721.3.1 AdjacencyFlags	1841
6.721.4 Member Function Documentation	1841
6.721.4.1 copyProperties()	1841
6.721.4.2 edgeDifference()	1841
6.721.4.3 findZeroDegree()	1841
6.721.4.4 getGraph()	1842

6.721.4.5 leaves()	1842
6.721.4.6 listPath()	1842
6.721.4.7 longestPathTouching()	1842
6.721.4.8 mostRemoteNodes()	1842
6.721.4.9 roots()	1842
6.721.4.10 rootsOf()	1843
6.721.4.11 shortestDistancesFrom()	1843
6.721.4.12 shortestPath()	1843
6.721.4.13 toList()	1843
6.721.4.14 topologicalSort()	1843
6.721.4.15 transitiveClosure()	1844
6.721.4.16 transitiveReduction()	1844
6.721.4.17 verticesBreadthFirst()	1844
6.721.4.18 verticesDepthFirstSorted()	1844
6.721.4.19 verticesDominatedBy() [1/2]	1844
6.721.4.20 verticesDominatedBy() [2/2]	1845
6.721.4.21 verticesDominatedByDepthFirstSorted()	1845
6.722 Digikam::Graph< VertexProperties, EdgeProperties >::DominatorTree Class Reference	1845
6.723 Digikam::Graph< VertexProperties, EdgeProperties >::Edge Class Reference	1845
6.724 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch Class Reference	1846
6.724.1 Member Function Documentation	1846
6.724.1.1 depth_first_search_sorted()	1846
6.725 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::BreadthFirstSearchVisitor Class Reference	1847
6.726 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor Class Reference	1848
6.727 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::DepthFirstSearchVisitor Class Reference	1849
6.728 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::lessThanMapEdgeTo↔Target< GraphType, VertexLessThan > Class Template Reference	1850
6.729 Digikam::Graph< VertexProperties, EdgeProperties >::Path Class Reference	1850
6.729.1 Detailed Description	1850
6.729.2 Member Function Documentation	1850
6.729.2.1 longestPath()	1850
6.729.2.2 shortestPath()	1851
6.730 Digikam::Graph< VertexProperties, EdgeProperties >::Vertex Class Reference	1851
6.730.1 Detailed Description	1851
6.731 Digikam::GraphicsDImgItem Class Reference	1852
6.731.1 Member Function Documentation	1853
6.731.1.1 setImage()	1853
6.732 Digikam::GraphicsDImgView Class Reference	1854
6.732.1 Member Function Documentation	1856
6.732.1.1 item()	1856

---

6.732.1.2 previewItem()	1856
6.732.1.3 scrollPointOnPoint()	1856
6.732.1.4 setItem()	1856
6.733 Digikam::GreycstorageContainer Class Reference	1856
6.734 Digikam::GreycstorageFilter Class Reference	1858
6.734.1 Member Enumeration Documentation	1861
6.734.1.1 MODE	1861
6.734.2 Constructor & Destructor Documentation	1861
6.734.2.1 GreycstorageFilter() [1/2]	1861
6.734.2.2 GreycstorageFilter() [2/2]	1861
6.734.3 Member Function Documentation	1862
6.734.3.1 cancelFilter()	1862
6.734.3.2 filterAction()	1862
6.734.3.3 filterIdentifier()	1862
6.734.3.4 readParameters()	1862
6.735 Digikam::GreycstorageSettings Class Reference	1863
6.736 Digikam::GroupedImagesFinder Class Reference	1863
6.736.1 Constructor & Destructor Documentation	1864
6.736.1.1 GroupedImagesFinder()	1864
6.737 Digikam::GroupIndicatorOverlay Class Reference	1865
6.737.1 Member Function Documentation	1867
6.737.1.1 checkIndex()	1867
6.737.1.2 createWidget()	1868
6.737.1.3 setActive()	1868
6.737.1.4 slotEntered()	1868
6.737.1.5 visualChange()	1868
6.738 Digikam::GroupIndicatorOverlayWidget Class Reference	1869
6.739 Digikam::GroupingViewImplementation Class Reference	1870
6.739.1 Member Function Documentation	1871
6.739.1.1 hasHiddenGroupedImages()	1871
6.740 Digikam::GroupItemFilterSettings Class Reference	1871
6.740.1 Member Function Documentation	1871
6.740.1.1 matches()	1871
6.740.1.2 setAllOpen()	1871
6.740.1.3 setOpen()	1872
6.741 Digikam::GroupStateComputer Class Reference	1872
6.742 Digikam::Haar::Calculator Class Reference	1872
6.742.1 Member Function Documentation	1872
6.742.1.1 calcHaar()	1872
6.742.1.2 transform()	1872
6.743 Digikam::Haar::ImageData Class Reference	1873
6.743.1 Member Function Documentation	1873

---

6.743.1.1 fillPixelData() [1/2]	1873
6.743.1.2 fillPixelData() [2/2]	1873
6.744 Digikam::Haar::SignatureData Class Reference	1873
6.744.1 Member Data Documentation	1873
6.744.1.1 avg	1873
6.744.1.2 sig	1874
6.745 Digikam::Haar::SignatureMap Class Reference	1874
6.745.1 Detailed Description	1874
6.746 Digikam::Haar::WeightBin Class Reference	1874
6.746.1 Constructor & Destructor Documentation	1874
6.746.1.1 WeightBin()	1874
6.746.2 Member Data Documentation	1875
6.746.2.1 m_bin	1875
6.747 Digikam::Haar::Weights Class Reference	1875
6.748 Digikam::Haarface Class Reference	1875
6.748.1 Member Enumeration Documentation	1876
6.748.1.1 RefImageSelMethod	1876
6.748.2 Member Function Documentation	1877
6.748.2.1 bestMatchesForImageWithThreshold() [1/2]	1877
6.748.2.2 bestMatchesForImageWithThreshold() [2/2]	1877
6.748.2.3 findDuplicates()	1877
6.748.2.4 fulfillsRestrictions()	1878
6.748.2.5 getBestAndWorstPossibleScore()	1878
6.748.2.6 imagesFromAlbumsAndTags()	1878
6.748.2.7 indexImage()	1878
6.748.2.8 loadQImage()	1878
6.748.2.9 rebuildDuplicatesAlbums()	1879
6.748.2.10 retrieveSignatureFromDB()	1879
6.748.2.11 setAlbumRootsToSearch()	1879
6.748.2.12 signatureAsText()	1879
6.749 Digikam::Haarface::Private Class Reference	1879
6.750 Digikam::HaarProgressObserver Class Reference	1880
6.751 Digikam::HidingStateChanger Class Reference	1881
6.751.1 Constructor & Destructor Documentation	1883
6.751.1.1 HidingStateChanger() [1/2]	1883
6.751.1.2 HidingStateChanger() [2/2]	1883
6.751.2 Member Function Documentation	1884
6.751.2.1 finished	1884
6.751.2.2 stateChanged	1884
6.752 Digikam::Highlighter Class Reference	1884
6.753 Digikam::HistogramBox Class Reference	1885
6.754 Digikam::HistogramPainter Class Reference	1886

6.754.1 Detailed Description	1887
6.754.2 Constructor & Destructor Documentation	1887
6.754.2.1 HistogramPainter()	1887
6.754.2.2 ~HistogramPainter()	1887
6.754.3 Member Function Documentation	1887
6.754.3.1 disableHistogramGuide()	1887
6.754.3.2 enableHistogramGuideByColor()	1888
6.754.3.3 initFrom()	1888
6.754.3.4 render()	1888
6.754.3.5 setChannelType()	1888
6.754.3.6 setHighlightSelection()	1889
6.754.3.7 setHistogram()	1889
6.754.3.8 setRenderXGrid()	1889
6.754.3.9 setScale()	1889
6.754.3.10 setSelection()	1890
6.755 Digikam::HistogramWidget Class Reference	1891
6.755.1 Constructor & Destructor Documentation	1892
6.755.1.1 HistogramWidget()	1892
6.755.2 Member Function Documentation	1893
6.755.2.1 currentHistogram()	1893
6.755.2.2 stopHistogramComputation()	1893
6.755.2.3 updateData()	1893
6.755.2.4 updateSelectionData()	1893
6.756 Digikam::HistoryEdgeProperties Class Reference	1893
6.756.1 Detailed Description	1893
6.757 Digikam::HistoryImageId Class Reference	1894
6.757.1 Member Enumeration Documentation	1895
6.757.1.1 Type	1895
6.757.2 Constructor & Destructor Documentation	1895
6.757.2.1 HistoryImageId()	1895
6.757.3 Member Data Documentation	1895
6.757.3.1 m_originalUUID	1895
6.757.3.2 m_uuid	1896
6.758 Digikam::HistoryVertexProperties Class Reference	1896
6.758.1 Detailed Description	1896
6.759 Digikam::HotPixelContainer Class Reference	1896
6.760 Digikam::HotPixelFixer Class Reference	1898
6.760.1 Member Function Documentation	1901
6.760.1.1 filterAction()	1901
6.760.1.2 filterIdentifier()	1901
6.760.1.3 readParameters()	1901
6.761 Digikam::HotPixelProps Class Reference	1901



6.761.1 Member Function Documentation	1902
6.761.1.1 operator==( )	1902
6.762 Digikam::HotPixelSettings Class Reference	1902
6.763 Digikam::HotPixelsWeights Class Reference	1903
6.764 Digikam::HoverButtonDelegateOverlay Class Reference	1904
6.764.1 Member Function Documentation	1906
6.764.1.1 createButton()	1906
6.764.1.2 createWidget()	1906
6.764.1.3 setActive()	1906
6.764.1.4 updateButton()	1906
6.764.1.5 visualChange()	1907
6.765 Digikam::HSLContainer Class Reference	1907
6.766 Digikam::HSLFilter Class Reference	1908
6.766.1 Member Function Documentation	1911
6.766.1.1 filterAction()	1911
6.766.1.2 filterIdentifier()	1911
6.766.1.3 readParameters()	1911
6.767 Digikam::HSLSettings Class Reference	1911
6.768 Digikam::HSPreviewWidget Class Reference	1912
6.769 Digikam::HTMLWidget Class Reference	1913
6.770 Digikam::HTMLWidgetPage Class Reference	1914
6.771 Digikam::IccManager Class Reference	1915
6.771.1 Constructor & Destructor Documentation	1916
6.771.1.1 IccManager()	1916
6.771.2 Member Function Documentation	1917
6.771.2.1 displaySoftProofingTransform()	1917
6.771.2.2 imageProfile()	1917
6.771.2.3 isSRGB()	1917
6.771.2.4 needsPostLoadingManagement()	1917
6.771.2.5 transform()	1917
6.771.2.6 transformDefault()	1917
6.771.2.7 transformForDisplay() [1/2]	1918
6.771.2.8 transformForDisplay() [2/2]	1918
6.771.2.9 transformForOutput()	1918
6.771.2.10 transformToSRGB() [1/2]	1918
6.771.2.11 transformToSRGB() [2/2]	1918
6.772 Digikam::IccPostLoadingManager Class Reference	1919
6.772.1 Constructor & Destructor Documentation	1921
6.772.1.1 IccPostLoadingManager()	1921
6.772.2 Member Function Documentation	1921
6.772.2.1 postLoadingManage()	1921
6.773 Digikam::ICCPreviewWidget Class Reference	1921

6.774 Digikam::IccProfile Class Reference	1922
6.774.1 Member Enumeration Documentation	1922
6.774.1.1 ProfileType	1922
6.774.2 Constructor & Destructor Documentation	1923
6.774.2.1 IccProfile() [1/3]	1923
6.774.2.2 IccProfile() [2/3]	1923
6.774.2.3 IccProfile() [3/3]	1923
6.774.3 Member Function Documentation	1923
6.774.3.1 close()	1923
6.774.3.2 data()	1923
6.774.3.3 defaultProfiles()	1924
6.774.3.4 defaultSearchPaths()	1924
6.774.3.5 description()	1924
6.774.3.6 filePath()	1924
6.774.3.7 handle()	1924
6.774.3.8 isOpen()	1924
6.774.3.9 isSameProfileAs()	1924
6.774.3.10 open()	1925
6.774.3.11 operator==( )	1925
6.774.3.12 sRGB()	1925
6.774.3.13 type()	1925
6.774.3.14 writeToFile()	1925
6.775 Digikam::IccProfileInfoDlg Class Reference	1925
6.776 Digikam::IccProfilesComboBox Class Reference	1926
6.776.1 Constructor & Destructor Documentation	1927
6.776.1.1 IccProfilesComboBox()	1927
6.776.2 Member Function Documentation	1927
6.776.2.1 addProfileSqueezed()	1927
6.776.2.2 addProfilesSqueezed()	1928
6.776.2.3 currentProfile()	1928
6.776.2.4 replaceProfilesSqueezed()	1928
6.776.2.5 setCurrentProfile()	1928
6.776.2.6 setNoProfileIfEmpty()	1928
6.777 Digikam::IccProfilesMenuAction Class Reference	1929
6.777.1 Member Function Documentation	1930
6.777.1.1 addProfile()	1930
6.777.1.2 addProfiles()	1930
6.777.1.3 disableIfEmpty()	1930
6.777.1.4 parentObject()	1930
6.777.1.5 replaceProfiles()	1930
6.778 Digikam::IccProfilesSettings Class Reference	1931
6.779 Digikam::IccProfileWidget Class Reference	1933

---

6.779.1 Member Function Documentation	1935
6.779.1.1 getMetadataTitle()	1935
6.779.1.2 getTagDescription()	1935
6.779.1.3 getTagTitle()	1935
6.779.1.4 loadFromURL()	1935
6.780 Digikam::lccRenderingIntentComboBox Class Reference	1936
6.781 Digikam::lccSettings Class Reference	1937
6.781.1 Member Function Documentation	1938
6.781.1.1 displayProfiles()	1938
6.781.1.2 inputProfiles()	1938
6.781.1.3 instance()	1938
6.781.1.4 isEnabled()	1938
6.781.1.5 loadAllProfilesProperties()	1939
6.781.1.6 monitorProfile()	1939
6.781.1.7 monitorProfileFromSystem()	1939
6.781.1.8 outputProfiles()	1939
6.781.1.9 profilesForDescription()	1939
6.781.1.10 setSettings()	1939
6.781.1.11 settings()	1939
6.781.1.12 setUseManagedView()	1940
6.781.1.13 useManagedPreviews()	1940
6.781.1.14 workspaceProfiles()	1940
6.782 Digikam::lccSettings::Private Class Reference	1940
6.783 Digikam::ICCSettingsContainer Class Reference	1940
6.783.1 Member Enumeration Documentation	1941
6.783.1.1 BehaviorEnum	1941
6.784 Digikam::lccTransform Class Reference	1942
6.784.1 Member Function Documentation	1942
6.784.1.1 apply() [1/2]	1942
6.784.1.2 apply() [2/2]	1943
6.784.1.3 close()	1943
6.784.1.4 effectiveInputProfile()	1943
6.784.1.5 embeddedProfile()	1943
6.784.1.6 init()	1943
6.784.1.7 setDoNotEmbedOutputProfile()	1943
6.784.1.8 setEmbeddedProfile()	1943
6.784.1.9 setIntent()	1944
6.784.1.10 setOutputProfile()	1944
6.784.1.11 setProofProfile()	1944
6.784.1.12 willHaveEffect()	1944
6.785 Digikam::lccTransformFilter Class Reference	1945
6.785.1 Member Function Documentation	1948

6.785.1.1 filterAction()	1948
6.785.1.2 filterIdentifier()	1948
6.785.1.3 filterImage()	1948
6.785.1.4 parametersSuccessfullyRead()	1949
6.785.1.5 progressInfo()	1949
6.785.1.6 readParameters()	1949
6.785.1.7 readParametersError()	1949
6.786 Digikam::Identity Class Reference	1949
6.786.1 Constructor & Destructor Documentation	1950
6.786.1.1 Identity()	1950
6.786.2 Member Function Documentation	1950
6.786.2.1 attribute()	1950
6.786.2.2 attributesMap()	1950
6.786.2.3 id()	1950
6.787 Digikam::IdentityProvider Class Reference	1950
6.787.1 Member Function Documentation	1951
6.787.1.1 addIdentity()	1951
6.787.1.2 addIdentityDebug()	1951
6.787.1.3 addTraining()	1951
6.787.1.4 allIdentities()	1951
6.787.1.5 clearAllTraining()	1952
6.787.1.6 clearTraining()	1952
6.787.1.7 deleteIdentities()	1952
6.787.1.8 deleteIdentity()	1952
6.787.1.9 findIdentity() [1/2]	1952
6.787.1.10 findIdentity() [2/2]	1952
6.787.1.11 getTrainingData()	1952
6.787.1.12 integrityCheck()	1953
6.787.1.13 isValidId()	1953
6.787.1.14 vacuum()	1953
6.788 Digikam::ImageChangeset Class Reference	1953
6.788.1 Constructor & Destructor Documentation	1953
6.788.1.1 ImageChangeset()	1953
6.789 Digikam::ImageCommonContainer Class Reference	1954
6.790 Digikam::ImageCurves Class Reference	1954
6.790.1 Member Enumeration Documentation	1955
6.790.1.1 CurveType	1955
6.790.2 Member Function Documentation	1955
6.790.2.1 channelToBinary()	1955
6.790.2.2 fillFromOtherCurves()	1956
6.790.2.3 getContainer() [1/2]	1956
6.790.2.4 getContainer() [2/2]	1956

6.790.2.5 isLinear()	1956
6.790.2.6 setChannelFromBinary()	1957
6.790.3 Member Data Documentation	1957
6.790.3.1 MULTIPLIER_16BIT	1957
6.790.3.2 NUM_CHANNELS	1957
6.790.3.3 NUMBER_OF_POINTS	1957
6.791 Digikam::ImageDialog Class Reference	1958
6.792 Digikam::ImageDialog::Private Class Reference	1959
6.793 Digikam::ImageDialogIconProvider Class Reference	1959
6.794 Digikam::ImageDialogPreview Class Reference	1960
6.795 Digikam::ImageDialogToolTip Class Reference	1961
6.796 Digikam::ImageGuideWidget Class Reference	1963
6.797 Digikam::ImageHistogram Class Reference	1965
6.797.1 Member Function Documentation	1967
6.797.1.1 calculate()	1967
6.797.1.2 calculationAboutToStart	1967
6.797.1.3 calculationStarted	1967
6.797.1.4 isSixteenBit()	1967
6.797.1.5 run()	1967
6.797.1.6 stopCalculation()	1968
6.798 Digikam::ImageHistoryEntry Class Reference	1968
6.799 Digikam::Imageface Class Reference	1968
6.799.1 Member Enumeration Documentation	1969
6.799.1.1 PreviewType	1969
6.799.2 Constructor & Destructor Documentation	1969
6.799.2.1 Imageface()	1969
6.799.3 Member Function Documentation	1969
6.799.3.1 colorInfoFromOriginal()	1969
6.799.3.2 convertOriginalColorDepth()	1969
6.799.3.3 convertToPixmap()	1969
6.799.3.4 crop()	1970
6.799.3.5 original()	1970
6.799.3.6 originalIccProfile()	1970
6.799.3.7 originalSize()	1970
6.799.3.8 paint()	1970
6.799.3.9 preview()	1970
6.799.3.10 previewReference()	1970
6.799.3.11 previewSize()	1971
6.799.3.12 selection()	1971
6.799.3.13 selectionRect()	1971
6.799.3.14 setOriginal()	1971
6.799.3.15 setOriginalIccProfile()	1971

---

6.799.3.16 setPreview()	1971
6.799.3.17 setPreviewIccProfile()	1972
6.799.3.18 setPreviewSize()	1972
6.799.3.19 setPreviewType()	1972
6.799.3.20 setSelection()	1972
6.800 Digikam::ImageLevels Class Reference	1972
6.800.1 Member Function Documentation	1973
6.800.1.1 levelsChannelReset()	1973
6.800.1.2 saveLevelsToGimpLevelsFile()	1973
6.800.1.3 setLevelGammaValue()	1973
6.801 Digikam::ImageListProvider Class Reference	1974
6.801.1 Detailed Description	1974
6.802 Digikam::ImageMetadataContainer Class Reference	1975
6.803 Digikam::ImagePreviewItem Class Reference	1976
6.804 Digikam::ImageQualityCalculator Class Reference	1977
6.805 Digikam::ImageQualityCalculator::ResultDetection Struct Reference	1978
6.806 Digikam::ImageQualityConfSelector Class Reference	1978
6.806.1 Member Enumeration Documentation	1979
6.806.1.1 SettingsType	1979
6.807 Digikam::ImageQualityContainer Class Reference	1979
6.808 Digikam::ImageQualityParser Class Reference	1980
6.808.1 Constructor & Destructor Documentation	1981
6.808.1.1 ImageQualityParser()	1981
6.808.2 Member Function Documentation	1981
6.808.2.1 startAnalyse()	1981
6.809 Digikam::ImageQualityParser::Private Class Reference	1981
6.810 Digikam::ImageQualitySettings Class Reference	1982
6.811 Digikam::ImageQualitySorter Class Reference	1983
6.811.1 Member Enumeration Documentation	1985
6.811.1.1 QualityScanMode	1985
6.811.2 Constructor & Destructor Documentation	1986
6.811.2.1 ImageQualitySorter()	1986
6.811.3 Member Function Documentation	1986
6.811.3.1 setUseMultiCoreCPU()	1986
6.812 Digikam::ImageQualityTask Class Reference	1987
6.813 Digikam::ImageQualityThread Class Reference	1988
6.814 Digikam::ImageQualityThreadPool Class Reference	1989
6.815 Digikam::ImageRegionItem Class Reference	1990
6.816 Digikam::ImageRegionWidget Class Reference	1992
6.816.1 Member Function Documentation	1994
6.816.1.1 getOriginalImageRegionToRender()	1994
6.816.1.2 getOriginalRegionImage()	1994

6.817 Digikam::ImageRelation Class Reference	1994
6.818 Digikam::ImageSortFilterModel Class Reference	1995
6.818.1 Member Function Documentation	1997
6.818.1.1 imageFilterModel()	1997
6.818.1.2 imageInfosSorted()	1997
6.818.1.3 mapListToSource()	1997
6.818.1.4 setDirectSourceItemModel()	1997
6.819 Digikam::ImageTagChangeset Class Reference	1997
6.819.1 Member Enumeration Documentation	1998
6.819.1.1 Operation	1998
6.819.2 Member Function Documentation	1998
6.819.2.1 operator<<()	1998
6.820 Digikam::ImageTagProperty Class Reference	1998
6.821 Digikam::ImageTagPropertyName Class Reference	1999
6.822 Digikam::ImageWindow Class Reference	2000
6.822.1 Member Function Documentation	2005
6.822.1.1 infoface()	2005
6.822.1.2 versionManager()	2006
6.823 Digikam::ImageWindow::Private Class Reference	2006
6.824 Digikam::ImageZoomSettings Class Reference	2006
6.824.1 Member Function Documentation	2007
6.824.1.1 fitToSize()	2007
6.824.1.2 fitToSizeZoomFactor()	2007
6.824.1.3 imageSize()	2008
6.824.1.4 mapImageToZoom() [1/2]	2008
6.824.1.5 mapImageToZoom() [2/2]	2008
6.824.1.6 mapZoomToImage()	2008
6.824.1.7 originalImageSize()	2008
6.824.1.8 realZoomFactor()	2008
6.824.1.9 setDisplayWidget()	2008
6.824.1.10 setImageSize()	2009
6.824.1.11 setImageSmoothScale()	2009
6.824.1.12 setZoomFactor()	2009
6.824.1.13 snappedZoomFactor()	2009
6.824.1.14 snappedZoomStep()	2009
6.824.1.15 sourceRect()	2009
6.824.1.16 zoomedSize()	2010
6.824.1.17 zoomFactor()	2010
6.825 Digikam::ImportCategorizedView Class Reference	2011
6.825.1 Member Function Documentation	2016
6.825.1.1 activated()	2016
6.825.1.2 addOverlay()	2016

---

6.825.1.3 camItemInfoActivated	2016
6.825.1.4 deselected	2016
6.825.1.5 dragDropHandler()	2016
6.825.1.6 filterModel()	2016
6.825.1.7 hintAt	2017
6.825.1.8 importFilterModel()	2017
6.825.1.9 importThumbnailModel()	2017
6.825.1.10 indexActivated()	2017
6.825.1.11 modelChanged	2017
6.825.1.12 nextIndexHint()	2017
6.825.1.13 nextInOrder()	2018
6.825.1.14 selected	2018
6.825.1.15 setCurrentInfo	2018
6.825.1.16 setCurrentUrl	2018
6.825.1.17 setCurrentWhenAvailable	2018
6.825.1.18 setSelectedCamItemInfos	2018
6.825.1.19 setSelectedUrls	2019
6.825.1.20 showContextMenuOnIndex()	2019
6.825.1.21 toIndex()	2019
6.826 Digikam::ImportCategoryDrawer Class Reference	2020
6.826.1 Member Function Documentation	2021
6.826.1.1 categoryHeight()	2021
6.826.1.2 drawCategory()	2021
6.827 Digikam::ImportContextMenuHelper Class Reference	2022
6.827.1 Constructor & Destructor Documentation	2023
6.827.1.1 ImportContextMenuHelper()	2023
6.827.2 Member Function Documentation	2023
6.827.2.1 addAction() [1/3]	2023
6.827.2.2 addAction() [2/3]	2024
6.827.2.3 addAction() [3/3]	2024
6.827.2.4 addAssignTagsMenu()	2024
6.827.2.5 addGroupMenu()	2025
6.827.2.6 addLabelsAction()	2025
6.827.2.7 addRemoveTagsMenu()	2025
6.827.2.8 addRotateMenu()	2026
6.827.2.9 addSeparator()	2026
6.827.2.10 addServicesMenu()	2026
6.827.2.11 addSubMenu()	2026
6.827.2.12 exec()	2027
6.827.2.13 setImportFilterModel()	2027
6.828 Digikam::ImportCoordinatesOverlay Class Reference	2028
6.828.1 Member Function Documentation	2030



---

6.828.1.1 checkIndex()	2030
6.828.1.2 createWidget()	2030
6.828.1.3 setActive()	2031
6.828.1.4 slotEntered()	2031
6.828.1.5 visualChange()	2031
6.829 Digikam::ImportDelegate Class Reference	2032
6.829.1 Member Function Documentation	2036
6.829.1.1 acceptsActivation()	2036
6.829.1.2 acceptsToolTip()	2036
6.829.1.3 clearCaches()	2036
6.829.1.4 clearModelDataCaches()	2037
6.829.1.5 imageInformationRect()	2037
6.829.1.6 invalidatePaintingCache()	2037
6.829.1.7 pixmapForDrag()	2037
6.829.1.8 pixmapRect()	2037
6.829.1.9 retrieveThumbnailPixmap()	2037
6.829.1.10 setDefaultViewOptions()	2038
6.829.1.11 setSpacing()	2038
6.829.1.12 updateContentWidth()	2038
6.829.1.13 updateRects()	2038
6.829.1.14 updateSizeRectsAndPxmmaps()	2038
6.830 Digikam::ImportDelegate::ImportDelegatePrivate Class Reference	2039
6.830.1 Member Function Documentation	2041
6.830.1.1 clearRects()	2041
6.831 Digikam::ImportDownloadOverlay Class Reference	2042
6.831.1 Member Function Documentation	2044
6.831.1.1 checkIndex()	2044
6.831.1.2 createWidget()	2044
6.831.1.3 setActive()	2045
6.831.1.4 slotEntered()	2045
6.831.1.5 visualChange()	2045
6.832 Digikam::ImportDragDropHandler Class Reference	2046
6.832.1 Member Function Documentation	2047
6.832.1.1 accepts()	2047
6.832.1.2 createMimeData()	2047
6.832.1.3 dropEvent()	2047
6.832.1.4 mimeTypes()	2048
6.833 Digikam::ImportFilterComboBox Class Reference	2048
6.834 Digikam::ImportFilterDlg Class Reference	2049
6.835 Digikam::ImportFilterModel Class Reference	2051
6.835.1 Member Enumeration Documentation	2053
6.835.1.1 ImportFilterModelRoles	2053

---

6.835.2 Member Function Documentation	2054
6.835.2.1 camItemInfosAdded	2054
6.835.2.2 categoryIdentifier()	2054
6.835.2.3 compareCategories()	2054
6.835.2.4 compareInfosCategories()	2055
6.835.2.5 importFilterModel()	2055
6.835.2.6 infosLessThan()	2055
6.835.2.7 setDirectSourceImportModel()	2055
6.835.2.8 subSortLessThan()	2056
6.836 Digikam::ImportIconView Class Reference	2057
6.836.1 Member Function Documentation	2063
6.836.1.1 activated()	2063
6.836.1.2 setThumbnailSize()	2063
6.836.1.3 showContextMenu()	2063
6.836.1.4 showContextMenuOnInfo()	2063
6.836.1.5 slotSetupChanged()	2063
6.837 Digikam::ImportIconView::Private Class Reference	2064
6.838 Digikam::ImportItemModel Class Reference	2065
6.838.1 Member Enumeration Documentation	2068
6.838.1.1 ImportItemModelRoles	2068
6.838.2 Member Function Documentation	2068
6.838.2.1 addCamItemInfoSynchronously()	2068
6.838.2.2 allRefreshingFinished	2068
6.838.2.3 camItemInfo() [1/2]	2068
6.838.2.4 camItemInfo() [2/2]	2068
6.838.2.5 camItemInfosCleared()	2069
6.838.2.6 clearCamItemInfos()	2069
6.838.2.7 indexForCamItemInfo()	2069
6.838.2.8 indexForUrl()	2069
6.838.2.9 isRefreshing()	2069
6.838.2.10 itemInfosAboutToBeAdded	2069
6.838.2.11 itemInfosAboutToBeRemoved [1/2]	2069
6.838.2.12 itemInfosAboutToBeRemoved() [2/2]	2070
6.838.2.13 itemInfosAdded	2070
6.838.2.14 itemInfosRemoved	2070
6.838.2.15 preprocess	2070
6.838.2.16 readyForIncrementalRefresh	2070
6.838.2.17 removeIndex()	2070
6.838.2.18 requestIncrementalRefresh()	2071
6.838.2.19 retrieveCamItemInfo()	2071
6.838.2.20 setCameraThumbsController()	2071
6.838.2.21 setCamItemInfos()	2071

---

6.838.2.22 setKeepsFileUrlCache()	2071
6.838.2.23 setSendRemovalSignals()	2071
6.838.2.24 startIncrementalRefresh()	2072
6.838.2.25 startRefresh()	2072
6.839 Digikam::ImportItemPropertiesSideBarImport Class Reference	2073
6.839.1 Member Function Documentation	2075
6.839.1.1 applySettings()	2075
6.839.1.2 doLoadState()	2076
6.839.1.3 doSaveState()	2076
6.840 Digikam::ImportItemPropertiesTab Class Reference	2077
6.841 Digikam::ImportLockOverlay Class Reference	2079
6.841.1 Member Function Documentation	2081
6.841.1.1 checkIndex()	2081
6.841.1.2 createWidget()	2081
6.841.1.3 setActive()	2082
6.841.1.4 slotEntered()	2082
6.841.1.5 visualChange()	2082
6.842 Digikam::ImportNormalDelegate Class Reference	2083
6.842.1 Member Function Documentation	2087
6.842.1.1 updateRects()	2087
6.843 Digikam::ImportNormalDelegatePrivate Class Reference	2088
6.844 Digikam::ImportOverlayWidget Class Reference	2090
6.845 Digikam::ImportPreviewView Class Reference	2091
6.845.1 Member Function Documentation	2093
6.845.1.1 acceptsMouseClicked()	2093
6.846 Digikam::ImportRatingOverlay Class Reference	2094
6.846.1 Member Function Documentation	2096
6.846.1.1 createWidget()	2096
6.846.1.2 hide()	2096
6.846.1.3 setActive()	2097
6.846.1.4 slotEntered()	2097
6.846.1.5 visualChange()	2097
6.846.1.6 widgetEnterEvent()	2097
6.846.1.7 widgetLeaveEvent()	2097
6.847 Digikam::ImportRenameParser Class Reference	2098
6.848 Digikam::ImportRotateOverlay Class Reference	2099
6.848.1 Member Function Documentation	2102
6.848.1.1 checkIndex()	2102
6.848.1.2 createButton()	2102
6.848.1.3 setActive()	2102
6.848.1.4 updateButton()	2102
6.848.1.5 widgetEnterEvent()	2102

---

6.848.1.6 widgetLeaveEvent()	2103
6.849 Digikam::ImportRotateOverlayButton Class Reference	2103
6.849.1 Member Function Documentation	2105
6.849.1.1 icon()	2105
6.849.1.2 sizeHint()	2105
6.849.1.3 updateToolTip()	2105
6.850 Digikam::ImportSettings Class Reference	2106
6.851 Digikam::ImportSortFilterModel Class Reference	2109
6.851.1 Member Function Documentation	2111
6.851.1.1 camItemInfosSorted()	2111
6.851.1.2 importFilterModel()	2111
6.851.1.3 mapToSourceImportModel()	2111
6.851.1.4 setDirectSourceImportModel()	2111
6.852 Digikam::ImportStackedView Class Reference	2112
6.852.1 Member Enumeration Documentation	2113
6.852.1.1 StackedViewMode	2113
6.853 Digikam::ImportThumbnailBar Class Reference	2114
6.853.1 Member Function Documentation	2119
6.853.1.1 setModelsFiltered()	2119
6.853.1.2 setScrollBarPolicy()	2119
6.853.1.3 slotSetupChanged()	2119
6.854 Digikam::ImportThumbnailDelegate Class Reference	2120
6.854.1 Member Function Documentation	2124
6.854.1.1 acceptsActivation()	2124
6.854.1.2 maximumSize()	2124
6.854.1.3 setDefaultViewOptions()	2124
6.854.1.4 updateContentWidth()	2125
6.854.1.5 updateRects()	2125
6.855 Digikam::ImportThumbnailDelegatePrivate Class Reference	2126
6.855.1 Constructor & Destructor Documentation	2128
6.855.1.1 ImportThumbnailDelegatePrivate()	2128
6.856 Digikam::ImportThumbnailModel Class Reference	2129
6.856.1 Constructor & Destructor Documentation	2132
6.856.1.1 ImportThumbnailModel()	2132
6.856.2 Member Function Documentation	2132
6.856.2.1 data()	2132
6.856.2.2 setCameraThumbsController()	2132
6.856.2.3 setData()	2133
6.856.2.4 setEmitDataChanged()	2133
6.857 Digikam::ImportUI Class Reference	2134
6.857.1 Member Function Documentation	2136
6.857.1.1 infoface()	2136

6.858 Digikam::ImportUI::Private Class Reference	2137
6.859 Digikam::ImportView Class Reference	2139
6.860 Digikam::InfoDlg Class Reference	2141
6.861 Digikam::InfraredContainer Class Reference	2142
6.862 Digikam::InfraredFilter Class Reference	2143
6.862.1 Member Function Documentation	2146
6.862.1.1 filterAction()	2146
6.862.1.2 filterIdentifier()	2146
6.862.1.3 readParameters()	2146
6.863 Digikam::InitializationObserver Class Reference	2147
6.864 Digikam::InsertBookmarksCommand Class Reference	2148
6.865 Digikam::InternalTagName Class Reference	2149
6.866 Digikam::InvertFilter Class Reference	2150
6.866.1 Member Function Documentation	2153
6.866.1.1 filterAction()	2153
6.866.1.2 filterIdentifier()	2153
6.866.1.3 readParameters()	2153
6.867 Digikam::IOFileSettings Class Reference	2153
6.867.1 Member Data Documentation	2154
6.867.1.1 JPEGSubSampling	2154
6.868 Digikam::IOJob Class Reference	2154
6.869 Digikam::IOJobData Class Reference	2155
6.870 Digikam::IOJobsManager Class Reference	2157
6.870.1 Member Function Documentation	2157
6.870.1.1 buildCollectionTrashCounters()	2157
6.870.1.2 instance()	2158
6.870.1.3 startDTrashItemsListingForCollection()	2158
6.870.1.4 startIOJobs()	2158
6.871 Digikam::IOJobsThread Class Reference	2159
6.871.1 Member Function Documentation	2161
6.871.1.1 copyOrMove()	2161
6.871.1.2 deleteFiles()	2161
6.871.1.3 emptyDTrashItems()	2161
6.871.1.4 errorsList()	2161
6.871.1.5 hasErrors()	2162
6.871.1.6 isCanceled()	2162
6.871.1.7 jobData()	2162
6.871.1.8 listDTrashItems()	2162
6.871.1.9 renameFile()	2162
6.871.1.10 restoreDTrashItems()	2163
6.872 Digikam::IptcCoreContactInfo Class Reference	2163
6.873 Digikam::IptcCoreLocationInfo Class Reference	2163

---

6.874 Digikam::IptcMetaEngineMergeHelper Class Reference	2164
6.875 Digikam::IptcWidget Class Reference	2165
6.875.1 Member Function Documentation	2167
6.875.1.1 getMetadataTitle()	2167
6.875.1.2 getTagDescription()	2167
6.875.1.3 getTagTitle()	2167
6.875.1.4 loadFromURL()	2167
6.876 Digikam::ItemAlbumFilterModel Class Reference	2168
6.876.1 Member Function Documentation	2172
6.876.1.1 compareInfosCategories() [1/2]	2172
6.876.1.2 compareInfosCategories() [2/2]	2173
6.876.1.3 setItemFilterSettings()	2173
6.877 Digikam::ItemAlbumModel Class Reference	2174
6.877.1 Member Function Documentation	2179
6.877.1.1 openAlbum	2179
6.877.1.2 refresh	2179
6.877.1.3 slotImageChange	2179
6.878 Digikam::ItemAttributesWatch Class Reference	2180
6.878.1 Member Function Documentation	2180
6.878.1.1 signalFileMetadataChanged	2180
6.878.1.2 signalImageRatingChanged	2181
6.878.1.3 signalImagesChanged	2181
6.878.1.4 signalImageTagsChanged	2181
6.879 Digikam::ItemCategorizedView Class Reference	2182
6.879.1 Member Function Documentation	2187
6.879.1.1 activated()	2187
6.879.1.2 albumAt()	2187
6.879.1.3 dragDropHandler()	2187
6.879.1.4 filterModel()	2188
6.879.1.5 hintAt	2188
6.879.1.6 indexActivated()	2188
6.879.1.7 nextIndexHint()	2188
6.879.1.8 nextInOrder()	2188
6.879.1.9 setCurrentInfo	2188
6.879.1.10 setCurrentUrl	2189
6.879.1.11 setCurrentUrlWhenAvailable	2189
6.879.1.12 setCurrentWhenAvailable	2189
6.879.1.13 setSelectedItemInfos	2189
6.879.1.14 setSelectedUrls	2189
6.879.1.15 showContextMenuOnIndex()	2189
6.879.1.16 toIndex()	2189
6.880 Digikam::ItemCategoryDrawer Class Reference	2190

6.880.1 Member Function Documentation	2191
6.880.1.1 categoryHeight()	2191
6.880.1.2 drawCategory()	2191
6.881 Digikam::ItemChangeHint Class Reference	2192
6.881.1 Member Enumeration Documentation	2192
6.881.1.1 ChangeType	2192
6.882 Digikam::ItemComments Class Reference	2193
6.882.1 Member Enumeration Documentation	2194
6.882.1.1 LanguageChoiceBehavior	2194
6.882.1.2 UniqueBehavior	2194
6.882.2 Constructor & Destructor Documentation	2194
6.882.2.1 ItemComments() [1/3]	2194
6.882.2.2 ItemComments() [2/3]	2195
6.882.2.3 ItemComments() [3/3]	2195
6.882.3 Member Function Documentation	2195
6.882.3.1 addComment()	2195
6.882.3.2 addHeadline()	2195
6.882.3.3 addTitle()	2195
6.882.3.4 apply()	2196
6.882.3.5 changeComment()	2196
6.882.3.6 commentForLanguage()	2196
6.882.3.7 defaultComment()	2196
6.882.3.8 language()	2196
6.882.3.9 numberOfComments()	2196
6.882.3.10 remove()	2197
6.882.3.11 removeAll() [1/2]	2197
6.882.3.12 removeAll() [2/2]	2197
6.882.3.13 removeAllComments()	2197
6.882.3.14 replaceComments()	2197
6.882.3.15 replaceFrom()	2197
6.882.3.16 setUniqueBehavior()	2197
6.882.3.17 toCaptionsMap()	2198
6.882.3.18 type()	2198
6.883 Digikam::ItemCoordinatesOverlay Class Reference	2199
6.883.1 Member Function Documentation	2201
6.883.1.1 checkIndex()	2201
6.883.1.2 createWidget()	2201
6.883.1.3 setActive()	2202
6.883.1.4 slotEntered()	2202
6.883.1.5 visualChange()	2202
6.884 Digikam::ItemCopyMoveHint Class Reference	2202
6.884.1 Constructor & Destructor Documentation	2203

---

6.884.1.1 ItemCopyMoveHint()	2203
6.885 Digikam::ItemCopyright Class Reference	2203
6.885.1 Member Enumeration Documentation	2204
6.885.1.1 ReplaceMode	2204
6.885.2 Constructor & Destructor Documentation	2205
6.885.2.1 ItemCopyright()	2205
6.885.3 Member Function Documentation	2205
6.885.3.1 contactInfo()	2205
6.885.3.2 copyrightNotice()	2205
6.885.3.3 creator()	2205
6.885.3.4 creatorJobTitle()	2205
6.885.3.5 fillTemplate()	2206
6.885.3.6 instructions()	2206
6.885.3.7 provider()	2206
6.885.3.8 removeAll()	2206
6.885.3.9 replaceFrom()	2206
6.885.3.10 rightsUsageTerms()	2206
6.885.3.11 setCopyrightNotice()	2207
6.885.3.12 setCreator()	2207
6.885.3.13 setFromTemplate()	2207
6.885.3.14 source()	2207
6.886 Digikam::ItemDelegate Class Reference	2208
6.886.1 Member Function Documentation	2212
6.886.1.1 acceptsActivation()	2212
6.886.1.2 acceptsToolTip()	2212
6.886.1.3 clearCaches()	2212
6.886.1.4 clearModelDataCaches()	2213
6.886.1.5 imageInformationRect()	2213
6.886.1.6 invalidatePaintingCache()	2213
6.886.1.7 pixmapForDrag()	2213
6.886.1.8 pixmapRect()	2213
6.886.1.9 retrieveThumbnailPixmap()	2213
6.886.1.10 setDefaultViewOptions()	2214
6.886.1.11 setSpacing()	2214
6.886.1.12 updateContentWidth()	2214
6.886.1.13 updateRects()	2214
6.886.1.14 updateSizeRectsAndPxmmaps()	2214
6.887 Digikam::ItemDelegate::ItemDelegatePrivate Class Reference	2215
6.887.1 Member Function Documentation	2217
6.887.1.1 clearRects()	2217
6.888 Digikam::ItemDelegateOverlay Class Reference	2217
6.888.1 Member Function Documentation	2218



6.888.1.1 affectsMultiple()	2218
6.888.1.2 mouseMoved()	2218
6.888.1.3 setActive()	2218
6.888.1.4 viewHasMultiSelection()	2219
6.888.1.5 visualChange	2219
6.889 Digikam::ItemDelegateOverlayContainer Class Reference	2220
6.889.1 Constructor & Destructor Documentation	2221
6.889.1.1 ItemDelegateOverlayContainer()	2221
6.889.2 Member Function Documentation	2221
6.889.2.1 asDelegate()	2221
6.890 Digikam::ItemDescEditTab Class Reference	2222
6.891 Digikam::ItemDescEditTab::Private Class Reference	2224
6.892 Digikam::ItemDragDropHandler Class Reference	2226
6.892.1 Member Function Documentation	2227
6.892.1.1 accepts()	2227
6.892.1.2 createMimeData()	2227
6.892.1.3 dropEvent()	2228
6.892.1.4 mimeTypees()	2228
6.892.1.5 setReadOnlyDrop()	2228
6.893 Digikam::ItemExtendedProperties Class Reference	2228
6.893.1 Constructor & Destructor Documentation	2229
6.893.1.1 ItemExtendedProperties()	2229
6.893.2 Member Function Documentation	2229
6.893.2.1 intellectualGenre()	2229
6.893.2.2 jobId()	2229
6.893.2.3 location()	2229
6.893.2.4 scene()	2230
6.893.2.5 similarityTo()	2230
6.893.2.6 subjectCode()	2230
6.894 Digikam::ItemFaceDelegate Class Reference	2231
6.894.1 Member Function Documentation	2235
6.894.1.1 thumbnailPixmap()	2235
6.894.1.2 updateRects()	2235
6.895 Digikam::ItemFaceDelegatePrivate Class Reference	2236
6.896 Digikam::ItemFilterModel Class Reference	2239
6.896.1 Member Enumeration Documentation	2243
6.896.1.1 ItemFilterModelRoles	2243
6.896.2 Member Function Documentation	2243
6.896.2.1 addPrepareHook()	2243
6.896.2.2 categoryIdentifier()	2244
6.896.2.3 compareCategories()	2244
6.896.2.4 compareInfosCategories() [1/2]	2245

6.896.2.5 compareInfosCategories() [2/2]	2245
6.896.2.6 data()	2245
6.896.2.7 filterMatches	2245
6.896.2.8 filterMatchesForText	2245
6.896.2.9 filterSettingsChanged	2246
6.896.2.10 imageFilterModel()	2246
6.896.2.11 imageInfosAdded	2246
6.896.2.12 infosLessThan()	2246
6.896.2.13 isGroupOpen()	2246
6.896.2.14 setDayFilter	2246
6.896.2.15 setDirectSourceItemModel()	2247
6.896.2.16 setGroupItemFilterSettings	2247
6.896.2.17 setItemFilterSettings	2247
6.896.2.18 setItemSortSettings	2247
6.896.2.19 setSendItemInfoSignals()	2247
6.896.2.20 setVersionItemFilterSettings	2247
6.896.2.21 subSortLessThan()	2248
6.896.2.22 suggestedWatchFlags()	2248
6.897 Digikam::ItemFilterModelFilterer Class Reference	2249
6.897.1 Member Function Documentation	2251
6.897.1.1 process()	2251
6.898 Digikam::ItemFilterModelPrepareHook Class Reference	2251
6.899 Digikam::ItemFilterModelPreparer Class Reference	2252
6.899.1 Member Function Documentation	2254
6.899.1.1 process()	2254
6.900 Digikam::ItemFilterModelTodoPackage Class Reference	2254
6.901 Digikam::ItemFilterModelWorker Class Reference	2255
6.902 Digikam::ItemFilterSettings Class Reference	2257
6.902.1 Member Function Documentation	2258
6.902.1.1 matches()	2258
6.902.1.2 watchFlags()	2258
6.903 Digikam::ItemFiltersHistoryItemDelegate Class Reference	2259
6.904 Digikam::ItemFiltersHistoryModel Class Reference	2260
6.905 Digikam::ItemFiltersHistoryTreeItem Class Reference	2261
6.906 Digikam::ItemFullScreenOverlay Class Reference	2262
6.906.1 Member Function Documentation	2265
6.906.1.1 checkIndex()	2265
6.906.1.2 createButton()	2265
6.906.1.3 setActive()	2265
6.906.1.4 updateButton()	2265
6.906.1.5 widgetEnterEvent()	2265
6.906.1.6 widgetLeaveEvent()	2266

---

6.907 Digikam::ItemFullScreenOverlayButton Class Reference	2266
6.907.1 Member Function Documentation	2267
6.907.1.1 icon()	2267
6.907.1.2 sizeHint()	2268
6.907.1.3 updateToolTip()	2268
6.908 Digikam::ItemGPS Class Reference	2269
6.908.1 Member Function Documentation	2271
6.908.1.1 loadImageData()	2271
6.908.1.2 saveChanges()	2272
6.909 Digikam::ItemGPSModelHelper Class Reference	2272
6.909.1 Member Function Documentation	2273
6.909.1.1 bestRepresentativeIndexFromList()	2273
6.909.1.2 itemCoordinates()	2274
6.909.1.3 model()	2274
6.909.1.4 pixmapFromRepresentativeIndex()	2274
6.909.1.5 selectionModel()	2274
6.910 Digikam::ItemHistoryGraph Class Reference	2274
6.910.1 Member Enumeration Documentation	2275
6.910.1.1 HistoryLoadingFlag	2275
6.910.2 Member Function Documentation	2275
6.910.2.1 addHistory()	2275
6.910.2.2 addRelations()	2276
6.910.2.3 addScannedHistory()	2276
6.910.2.4 allImages()	2276
6.910.2.5 categorize()	2276
6.910.2.6 clear()	2276
6.910.2.7 dropUnresolvedEntries()	2276
6.910.2.8 fromInfo()	2277
6.910.2.9 hasEdges()	2277
6.910.2.10 hasUnresolvedEntries()	2277
6.910.2.11 leafImages()	2277
6.910.2.12 prepareForDisplay()	2277
6.910.2.13 reduceEdges()	2277
6.910.2.14 relationCloud()	2277
6.910.2.15 rootImages()	2278
6.910.2.16 sortForInfo()	2278
6.911 Digikam::ItemHistoryGraphData Class Reference	2279
6.912 Digikam::ItemHistoryGraphModel Class Reference	2283
6.912.1 Member Function Documentation	2285
6.912.1.1 imageModel()	2285
6.912.1.2 imageModelIndex()	2285
6.912.1.3 indexForInfo()	2285

---

6.912.1.4 setHistory()	2285
6.913 Digikam::ItemIconView Class Reference	2286
6.913.1 Member Function Documentation	2290
6.913.1.1 allNeedGroupResolving()	2290
6.913.1.2 allUrls()	2290
6.913.1.3 selectedUrls()	2290
6.913.1.4 slotFitToWindow	2291
6.913.1.5 slotImageQualitySorter	2291
6.913.1.6 slotRemoveTag	2291
6.914 Digikam::ItemIconView::Private Class Reference	2291
6.915 Digikam::ItemInfo Class Reference	2292
6.915.1 Detailed Description	2295
6.915.2 Constructor & Destructor Documentation	2295
6.915.2.1 ItemInfo() [1/4]	2295
6.915.2.2 ItemInfo() [2/4]	2295
6.915.2.3 ItemInfo() [3/4]	2295
6.915.2.4 ItemInfo() [4/4]	2295
6.915.2.5 ~ItemInfo()	2295
6.915.3 Member Function Documentation	2296
6.915.3.1 addTagPaths()	2296
6.915.3.2 addToGroup()	2296
6.915.3.3 albumId()	2296
6.915.3.4 albumRootId()	2296
6.915.3.5 aspectRatio()	2296
6.915.3.6 category()	2296
6.915.3.7 clearGroup()	2297
6.915.3.8 colorLabel()	2297
6.915.3.9 comment()	2297
6.915.3.10 copyItem()	2297
6.915.3.11 currentReferenceImage()	2297
6.915.3.12 dateTime()	2298
6.915.3.13 dimensions()	2298
6.915.3.14 faceCount()	2298
6.915.3.15 filePath()	2298
6.915.3.16 fileSize()	2298
6.915.3.17 fileUrl()	2298
6.915.3.18 format()	2299
6.915.3.19 fromLocalFile()	2299
6.915.3.20 fromLocationAlbumAndName()	2299
6.915.3.21 fromUniqueHash()	2299
6.915.3.22 getDatabaseFieldsRaw()	2299
6.915.3.23 getSuggestedNames()	2299

---

6.915.3.24 groupedImages()	2300
6.915.3.25 groupImage()	2300
6.915.3.26 hasDerivedImages()	2300
6.915.3.27 hasGroupedImages()	2300
6.915.3.28 hash()	2300
6.915.3.29 historyImageId()	2300
6.915.3.30 id()	2300
6.915.3.31 imageComments()	2301
6.915.3.32 imageCommonContainer()	2301
6.915.3.33 imageCopyright()	2301
6.915.3.34 imageExtendedProperties()	2301
6.915.3.35 imageHistory()	2301
6.915.3.36 imagePosition()	2301
6.915.3.37 imageTagPair()	2301
6.915.3.38 isGrouped()	2302
6.915.3.39 isLocationAvailable()	2302
6.915.3.40 isNull()	2302
6.915.3.41 isRemoved()	2302
6.915.3.42 isVisible()	2302
6.915.3.43 longitudeNumber()	2302
6.915.3.44 manualOrder()	2302
6.915.3.45 markDerivedFrom()	2303
6.915.3.46 metadataTemplate()	2303
6.915.3.47 modDateTime()	2303
6.915.3.48 name()	2303
6.915.3.49 orientation()	2303
6.915.3.50 pickLabel()	2303
6.915.3.51 rating()	2303
6.915.3.52 relationCloud()	2304
6.915.3.53 relativePath()	2304
6.915.3.54 removeAllTags()	2304
6.915.3.55 removeFromGroup()	2304
6.915.3.56 removeMetadataTemplate()	2304
6.915.3.57 removeTag()	2304
6.915.3.58 setColorLabel()	2304
6.915.3.59 setDateTime()	2305
6.915.3.60 setManualOrder()	2305
6.915.3.61 setMetadataTemplate()	2305
6.915.3.62 setModDateTime()	2305
6.915.3.63 setName()	2306
6.915.3.64 setOrientation()	2306
6.915.3.65 setPickLabel()	2306

6.915.3.66 setRating()	2306
6.915.3.67 setTag()	2306
6.915.3.68 setVisible()	2306
6.915.3.69 tagIds()	2307
6.915.3.70 thumbnailIdentifier()	2307
6.915.3.71 title()	2307
6.915.3.72 unconfirmedFaceCount()	2307
6.915.3.73 uniqueHash()	2307
6.915.3.74 uuid()	2308
6.916 Digikam::ItemInfoAlbumsJob Class Reference	2308
6.917 Digikam::ItemInfoCache Class Reference	2309
6.917.1 Member Function Documentation	2309
6.917.1.1 albumRelativePath()	2309
6.917.1.2 cacheByName()	2310
6.917.1.3 dropInfo()	2310
6.917.1.4 getImageGroupedCount()	2310
6.917.1.5 infoForId()	2310
6.917.1.6 infoForPath()	2310
6.917.1.7 invalidate()	2310
6.918 Digikam::ItemInfoData Class Reference	2311
6.919 Digikam::ItemInfoJob Class Reference	2313
6.920 Digikam::ItemInfoList Class Reference	2314
6.920.1 Member Function Documentation	2314
6.920.1.1 singleGroupMainItem()	2314
6.921 Digikam::ItemInfoReadLocker Class Reference	2315
6.922 Digikam::ItemInfoSet Class Reference	2315
6.922.1 Detailed Description	2315
6.923 Digikam::ItemInfoStatic Class Reference	2315
6.924 Digikam::ItemInfoTaskSplitter Class Reference	2317
6.925 Digikam::ItemInfoWriteLocker Class Reference	2319
6.926 Digikam::ItemListDragDropHandler Class Reference	2320
6.927 Digikam::ItemLISTER Class Reference	2320
6.927.1 Member Function Documentation	2321
6.927.1.1 list()	2321
6.927.1.2 listAreaRange()	2321
6.927.1.3 listDateRange()	2322
6.927.1.4 listHaarSearch()	2322
6.927.1.5 listImageTagPropertySearch()	2322
6.927.1.6 listPALbum()	2322
6.927.1.7 listSearch()	2323
6.927.1.8 listTag()	2323
6.927.1.9 setListOnlyAvailable()	2323

6.927.1.10 setRecursive()	2323
6.928 Digikam::ItemLister::Private Class Reference	2323
6.929 Digikam::ItemListerJobGrowingPartsSendingReceiver Class Reference	2324
6.929.1 Member Function Documentation	2325
6.929.1.1 receive()	2325
6.930 Digikam::ItemListerJobPartsSendingReceiver Class Reference	2326
6.930.1 Member Function Documentation	2327
6.930.1.1 receive()	2327
6.931 Digikam::ItemListerJobReceiver Class Reference	2328
6.931.1 Member Function Documentation	2329
6.931.1.1 error()	2329
6.932 Digikam::ItemListerReceiver Class Reference	2330
6.933 Digikam::ItemListerRecord Class Reference	2331
6.934 Digikam::ItemListerValueListReceiver Class Reference	2332
6.934.1 Member Function Documentation	2333
6.934.1.1 error()	2333
6.934.1.2 receive()	2333
6.935 Digikam::ItemListModel Class Reference	2334
6.935.1 Member Function Documentation	2338
6.935.1.1 imageInfosRemoved	2338
6.936 Digikam::ItemMarkerTiler Class Reference	2339
6.936.1 Member Function Documentation	2340
6.936.1.1 bestRepresentativeIndexFromList()	2340
6.936.1.2 getGlobalGroupState()	2341
6.936.1.3 getTile()	2341
6.936.1.4 getTileGroupState()	2341
6.936.1.5 getTileMarkerCount()	2341
6.936.1.6 getTileRepresentativeMarker()	2341
6.936.1.7 getTileSelectedCount()	2341
6.936.1.8 indicesEqual()	2341
6.936.1.9 onIndicesClicked()	2342
6.936.1.10 onIndicesMoved()	2342
6.936.1.11 pixmapFromRepresentativeIndex()	2342
6.936.1.12 prepareTiles()	2342
6.936.1.13 regenerateTiles()	2342
6.936.1.14 removeMarkerIndexFromGrid()	2342
6.936.1.15 setActive()	2343
6.936.1.16 tileNew()	2343
6.936.1.17 tilerFlags()	2343
6.937 Digikam::ItemMetadataAdjustmentHint Class Reference	2343
6.937.1 Member Enumeration Documentation	2343
6.937.1.1 AdjustmentStatus	2343

---

6.938 Digikam::ItemModel Class Reference	2345
6.938.1 Member Enumeration Documentation	2348
6.938.1.1 ItemModelRoles	2348
6.938.2 Member Function Documentation	2349
6.938.2.1 addItemInfo()	2349
6.938.2.2 addItemInfoSynchronously()	2349
6.938.2.3 allRefreshingFinished	2349
6.938.2.4 clearItemInfos()	2349
6.938.2.5 ensureHasGroupedImages()	2349
6.938.2.6 ensureHasItemInfo()	2349
6.938.2.7 imageChange	2350
6.938.2.8 imageInfo() [1/2]	2350
6.938.2.9 imageInfo() [2/2]	2350
6.938.2.10 imageInfosAboutToBeAdded	2350
6.938.2.11 imageInfosAboutToBeRemoved [1/2]	2350
6.938.2.12 imageInfosAboutToBeRemoved() [2/2]	2350
6.938.2.13 imageInfosAdded	2351
6.938.2.14 imageInfosCleared()	2351
6.938.2.15 imageInfosRemoved	2351
6.938.2.16 imageTagChange	2351
6.938.2.17 indexForItemInfo()	2351
6.938.2.18 indexForPath()	2351
6.938.2.19 isRefreshing()	2352
6.938.2.20 preprocess	2352
6.938.2.21 readyForIncrementalRefresh	2352
6.938.2.22 removeIndex()	2352
6.938.2.23 requestIncrementalRefresh()	2352
6.938.2.24 retrieveItemInfo()	2352
6.938.2.25 setItemInfos()	2352
6.938.2.26 setKeepsFilePathCache()	2353
6.938.2.27 setPreprocessor()	2353
6.938.2.28 setSendRemovalSignals()	2353
6.938.2.29 setWatchFlags()	2353
6.938.2.30 startIncrementalRefresh()	2353
6.938.2.31 startRefresh()	2353
6.939 Digikam::ItemPosition Class Reference	2354
6.939.1 Constructor & Destructor Documentation	2354
6.939.1.1 ItemPosition() [1/2]	2354
6.939.1.2 ItemPosition() [2/2]	2355
6.939.2 Member Function Documentation	2355
6.939.2.1 altitude()	2355
6.939.2.2 altitudeFormatted()	2355



6.939.2.3 apply()	2355
6.939.2.4 isEmpty()	2355
6.939.2.5 latitude()	2355
6.939.2.6 latitudeFormatted()	2355
6.939.2.7 latitudeNumber()	2356
6.939.2.8 latitudeUserPresentableNumbers()	2356
6.939.2.9 remove()	2356
6.939.2.10 removeAltitude()	2356
6.939.2.11 setAltitude()	2356
6.939.2.12 setLatitude() [1/2]	2356
6.939.2.13 setLatitude() [2/2]	2357
6.940 Digikam::ItemPreviewCanvas Class Reference	2358
6.941 Digikam::ItemPreviewView Class Reference	2361
6.941.1 Member Function Documentation	2363
6.941.1.1 acceptsMouseClicked()	2363
6.942 Digikam::ItemPropertiesColorsTab Class Reference	2364
6.943 Digikam::ItemPropertiesGPSTab Class Reference	2365
6.944 Digikam::ItemPropertiesHistoryTab Class Reference	2366
6.945 Digikam::ItemPropertiesMetadataTab Class Reference	2367
6.946 Digikam::ItemPropertiesSideBar Class Reference	2368
6.946.1 Member Function Documentation	2371
6.946.1.1 doLoadState()	2371
6.946.1.2 doSaveState()	2371
6.947 Digikam::ItemPropertiesSideBarDB Class Reference	2372
6.947.1 Member Function Documentation	2376
6.947.1.1 doLoadState()	2376
6.947.1.2 doSaveState()	2376
6.947.1.3 itemChanged()	2376
6.948 Digikam::ItemPropertiesTab Class Reference	2377
6.948.1 Member Function Documentation	2380
6.948.1.1 aspectRatioToString()	2380
6.948.1.2 humanReadableBytesCount()	2380
6.948.1.3 permissionsString()	2380
6.948.1.4 shortenedMakeInfo()	2380
6.948.1.5 shortenedTagPaths()	2380
6.949 Digikam::ItemPropertiesTab::Private Class Reference	2381
6.950 Digikam::ItemPropertiesVersionsTab Class Reference	2382
6.951 Digikam::ItemQueryBuilder Class Reference	2383
6.951.1 Member Function Documentation	2383
6.951.1.1 setImageTagPropertiesJoined()	2383
6.952 Digikam::ItemQueryPostHook Class Reference	2383
6.952.1 Constructor & Destructor Documentation	2384

---

6.952.1.1 ItemQueryPostHook()	2384
6.953 Digikam::ItemQueryPostHooks Class Reference	2384
6.953.1 Member Function Documentation	2384
6.953.1.1 addHook()	2384
6.953.1.2 checkPosition()	2384
6.954 Digikam::ItemRatingOverlay Class Reference	2385
6.954.1 Member Function Documentation	2387
6.954.1.1 createWidget()	2387
6.954.1.2 hide()	2387
6.954.1.3 setActive()	2388
6.954.1.4 slotEntered()	2388
6.954.1.5 visualChange()	2388
6.954.1.6 widgetEnterEvent()	2388
6.954.1.7 widgetLeaveEvent()	2388
6.955 Digikam::ItemRotateOverlay Class Reference	2389
6.955.1 Member Function Documentation	2392
6.955.1.1 checkIndex()	2392
6.955.1.2 createButton()	2392
6.955.1.3 setActive()	2392
6.955.1.4 updateButton()	2392
6.955.1.5 widgetEnterEvent()	2392
6.955.1.6 widgetLeaveEvent()	2393
6.956 Digikam::ItemRotateOverlayButton Class Reference	2393
6.956.1 Member Function Documentation	2395
6.956.1.1 icon()	2395
6.956.1.2 sizeHint()	2395
6.956.1.3 updateToolTip()	2395
6.957 Digikam::ItemScanInfo Class Reference	2395
6.958 Digikam::ItemScanner Class Reference	2395
6.958.1 Constructor & Destructor Documentation	2397
6.958.1.1 ItemScanner() [1/3]	2397
6.958.1.2 ItemScanner() [2/3]	2398
6.958.1.3 ItemScanner() [3/3]	2398
6.958.2 Member Function Documentation	2398
6.958.2.1 cleanScan()	2398
6.958.2.2 commit()	2398
6.958.2.3 copiedFrom()	2398
6.958.2.4 creationDateFromFilesystem()	2398
6.958.2.5 fileModified()	2399
6.958.2.6 fillCommonContainer()	2399
6.958.2.7 fillVideoMetadataContainer()	2399
6.958.2.8 formatToString()	2399

6.958.2.9	hasHistoryToResolve()	2399
6.958.2.10	id()	2399
6.958.2.11	iptcCorePropertyName()	2400
6.958.2.12	itemScanInfo()	2400
6.958.2.13	loadFromDisk()	2400
6.958.2.14	newFile()	2400
6.958.2.15	newFileFullScan()	2400
6.958.2.16	rescan()	2400
6.958.2.17	resolvedImageHistory()	2401
6.958.2.18	resolveHistoryImageId()	2401
6.958.2.19	resolveImageHistory()	2401
6.958.2.20	sameReferredImage()	2401
6.958.2.21	setCategory()	2401
6.958.2.22	sortByProximity()	2401
6.958.2.23	tagItemHistoryGraph()	2402
6.959	Digikam::ItemScanner::Private Class Reference	2402
6.960	Digikam::ItemScannerCommit Class Reference	2402
6.961	Digikam::ItemSelectionOverlay Class Reference	2403
6.961.1	Member Function Documentation	2406
6.961.1.1	createButton()	2406
6.961.1.2	setActive()	2406
6.961.1.3	updateButton()	2406
6.962	Digikam::ItemSelectionOverlayButton Class Reference	2407
6.962.1	Member Function Documentation	2408
6.962.1.1	icon()	2408
6.962.1.2	sizeHint()	2409
6.962.1.3	updateToolTip()	2409
6.963	Digikam::ItemSelectionPropertiesTab Class Reference	2410
6.964	Digikam::ItemShortInfo Class Reference	2412
6.965	Digikam::ItemSortCollator Class Reference	2412
6.965.1	Member Function Documentation	2413
6.965.1.1	instance()	2413
6.966	Digikam::ItemSortSettings Class Reference	2413
6.966.1	Member Enumeration Documentation	2414
6.966.1.1	CategorizationMode	2414
6.966.1.2	SortOrder	2414
6.966.1.3	SortRole	2415
6.966.2	Member Function Documentation	2415
6.966.2.1	compare()	2415
6.966.2.2	compareByOrder()	2415
6.966.2.3	compareCategories()	2415
6.966.2.4	compareValue()	2415

---

6.966.2.5 lessThan() [1/2]	2416
6.966.2.6 lessThan() [2/2]	2416
6.966.2.7 lessThanByOrder()	2416
6.966.2.8 naturalCompare()	2416
6.966.2.9 watchFlags()	2416
6.967 Digikam::ItemTagPair Class Reference	2417
6.967.1 Constructor & Destructor Documentation	2417
6.967.1.1 ItemTagPair() [1/2]	2417
6.967.1.2 ItemTagPair() [2/2]	2418
6.967.2 Member Function Documentation	2418
6.967.2.1 addProperty()	2418
6.967.2.2 assignTag()	2418
6.967.2.3 availablePairs()	2418
6.967.2.4 isAssigned()	2418
6.967.2.5 unAssignTag()	2418
6.968 Digikam::ItemThumbnailBar Class Reference	2419
6.968.1 Member Function Documentation	2425
6.968.1.1 hasHiddenGroupedImages()	2425
6.968.1.2 setModelsFiltered()	2425
6.968.1.3 setScrollBarPolicy()	2425
6.968.1.4 slotSetupChanged()	2425
6.969 Digikam::ItemThumbnailDelegate Class Reference	2426
6.969.1 Member Function Documentation	2430
6.969.1.1 acceptsActivation()	2430
6.969.1.2 maximumSize()	2430
6.969.1.3 setDefaultViewOptions()	2430
6.969.1.4 updateContentWidth()	2431
6.969.1.5 updateRects()	2431
6.970 Digikam::ItemThumbnailDelegatePrivate Class Reference	2432
6.971 Digikam::ItemThumbnailModel Class Reference	2435
6.971.1 Constructor & Destructor Documentation	2439
6.971.1.1 ItemThumbnailModel()	2439
6.971.2 Member Function Documentation	2439
6.971.2.1 data()	2439
6.971.2.2 imageInfosCleared()	2439
6.971.2.3 preloadThumbnails	2439
6.971.2.4 setData()	2440
6.971.2.5 setEmitDataChanged()	2440
6.971.2.6 setPreloadThumbnails()	2440
6.971.2.7 setThumbnailLoadThread()	2440
6.972 Digikam::ItemVersionsModel Class Reference	2441
6.973 Digikam::ItemViewCategorized Class Reference	2442

---

6.973.1 Member Function Documentation	2445
6.973.1.1 clicked	2445
6.973.1.2 filterModel()	2445
6.973.1.3 indexForCategoryAt()	2445
6.973.1.4 keyPressed	2446
6.973.1.5 mapIndexForDragDrop()	2446
6.973.1.6 nextIndexHint()	2446
6.973.1.7 pixmapForDrag()	2446
6.973.1.8 rowsRemoved()	2446
6.973.1.9 scrollToRelaxed()	2447
6.973.1.10 selectionChanged	2447
6.973.1.11 selectionCleared	2447
6.973.1.12 setInitialSelectedItem()	2447
6.973.1.13 setScrollCurrentToCenter()	2447
6.973.1.14 setScrollStepGranularity()	2447
6.973.1.15 setSpacing()	2447
6.973.1.16 setUsePointingHandCursor()	2448
6.973.1.17 showContextMenuOnIndex()	2448
6.973.1.18 showToolTip()	2448
6.973.1.19 toFirstIndex()	2448
6.973.1.20 viewportClicked	2448
6.974 Digikam::ItemViewDelegate Class Reference	2449
6.974.1 Member Function Documentation	2452
6.974.1.1 acceptsActivation()	2452
6.974.1.2 acceptsToolTip()	2452
6.974.1.3 asDelegate()	2452
6.974.1.4 drawThumbnail()	2453
6.974.1.5 gridSize()	2453
6.974.1.6 imageInformationRect()	2453
6.974.1.7 mouseMoved()	2453
6.974.1.8 pixmapRect()	2453
6.974.1.9 ratingPixmap()	2454
6.974.1.10 ratingRect()	2454
6.974.1.11 setDefaultViewOptions()	2454
6.974.1.12 setRatingEdited()	2454
6.974.1.13 setSpacing()	2454
6.974.1.14 setThumbnailSize()	2454
6.975 Digikam::ItemViewDelegatePrivate Class Reference	2455
6.975.1 Member Function Documentation	2456
6.975.1.1 clearRects()	2456
6.976 Digikam::ItemViewHoverButton Class Reference	2456
6.976.1 Member Function Documentation	2457

---

6.976.1.1 icon()	2457
6.976.1.2 setup()	2457
6.976.1.3 sizeHint()	2458
6.976.1.4 updateToolTip()	2458
6.977 Digikam::ItemViewImportDelegate Class Reference	2459
6.977.1 Member Function Documentation	2462
6.977.1.1 acceptsActivation()	2462
6.977.1.2 acceptsToolTip()	2462
6.977.1.3 asDelegate()	2462
6.977.1.4 gridSize()	2463
6.977.1.5 imageInformationRect()	2463
6.977.1.6 invalidatePaintingCache()	2463
6.977.1.7 mouseMoved()	2463
6.977.1.8 pixmapRect()	2463
6.977.1.9 prepareRatingPixmaps()	2463
6.977.1.10 ratingRect()	2464
6.977.1.11 setDefaultViewOptions()	2464
6.977.1.12 setRatingEdited()	2464
6.977.1.13 setSpacing()	2464
6.977.1.14 setThumbnailSize()	2464
6.978 Digikam::ItemViewImportDelegatePrivate Class Reference	2465
6.978.1 Member Function Documentation	2466
6.978.1.1 clearRects()	2466
6.979 Digikam::ItemViewToolTip Class Reference	2467
6.979.1 Member Function Documentation	2468
6.979.1.1 repositionRect()	2468
6.979.1.2 show()	2468
6.979.1.3 tipContents()	2468
6.980 Digikam::ItemViewUtilities Class Reference	2469
6.981 Digikam::ItemVisibilityController Class Reference	2471
6.981.1 Member Enumeration Documentation	2472
6.981.1.1 IncludeFadingOutMode	2472
6.981.1.2 State	2473
6.981.2 Member Function Documentation	2473
6.981.2.1 addItem()	2473
6.981.2.2 clear()	2473
6.981.2.3 createAnimation()	2473
6.981.2.4 hasVisibleItems()	2474
6.981.2.5 hiddenAndRemoved	2474
6.981.2.6 hideAndRemoveItem	2474
6.981.2.7 items()	2474
6.981.2.8 propertiesAssigned [1/2]	2474

---

6.981.2.9 propertiesAssigned [2/2]	2474
6.981.2.10 setEasingCurve()	2474
6.981.2.11 setItemThatShallBeShown	2475
6.981.2.12 setShallBeShown	2475
6.981.2.13 show	2475
6.981.2.14 showItem	2475
6.981.2.15 visibleItems()	2475
6.982 Digikam::ItemVisibilityControllerPropertyObject Class Reference	2476
6.982.1 Constructor & Destructor Documentation	2477
6.982.1.1 ItemVisibilityControllerPropertyObject()	2477
6.983 Digikam::JPEGUtils::digikam_source_mgr Struct Reference	2477
6.984 Digikam::JPEGUtils::JpegRotator Class Reference	2477
6.984.1 Constructor & Destructor Documentation	2478
6.984.1.1 JpegRotator()	2478
6.984.1.2 ~JpegRotator()	2478
6.984.2 Member Function Documentation	2478
6.984.2.1 autoExifTransform()	2478
6.984.2.2 exifTransform() [1/2]	2478
6.984.2.3 exifTransform() [2/2]	2479
6.984.2.4 setCurrentOrientation()	2479
6.984.2.5 setDestinationFile()	2479
6.984.2.6 setDocumentName()	2479
6.985 Digikam::KDNodeBase Class Reference	2480
6.985.1 Member Function Documentation	2481
6.985.1.1 createNode()	2481
6.985.1.2 getClosestNeighbors()	2481
6.985.1.3 getIdentity()	2481
6.985.1.4 getPosition()	2481
6.985.1.5 insert()	2482
6.985.1.6 setNodeId()	2482
6.986 Digikam::KDNodeBase::NodeCompareResult Struct Reference	2482
6.987 Digikam::KDNodeOpenFace Class Reference	2483
6.987.1 Member Function Documentation	2484
6.987.1.1 createNode()	2484
6.987.1.2 nodeCompare()	2484
6.988 Digikam::KDNodeSFace Class Reference	2485
6.988.1 Member Function Documentation	2486
6.988.1.1 createNode()	2486
6.988.1.2 nodeCompare()	2486
6.989 Digikam::KDTreeBase Class Reference	2487
6.989.1 Constructor & Destructor Documentation	2487
6.989.1.1 KDTreeBase()	2487

6.989.2 Member Function Documentation	2488
6.989.2.1 add()	2488
6.989.2.2 createNode()	2488
6.989.2.3 getClosestNeighbors()	2488
6.990 Digikam::KDTreeOpenFace Class Reference	2489
6.991 Digikam::KDTreeSFace Class Reference	2490
6.992 Digikam::KeywordSearchReader Class Reference	2491
6.993 Digikam::KeywordSearchWriter Class Reference	2493
6.994 Digikam::LabelsSideBarWidget Class Reference	2495
6.994.1 Member Function Documentation	2496
6.994.1.1 applySettings()	2496
6.994.1.2 changeAlbumFromHistory()	2497
6.994.1.3 doLoadState()	2497
6.994.1.4 doSaveState()	2497
6.994.1.5 getCaption()	2497
6.994.1.6 getIcon()	2497
6.994.1.7 setActive()	2497
6.995 Digikam::LabelsTreeView Class Reference	2498
6.995.1 Member Function Documentation	2499
6.995.1.1 colorRectPixmap()	2499
6.995.1.2 doLoadState()	2500
6.995.1.3 doSaveState()	2500
6.995.1.4 goldenStarPixmap()	2500
6.995.1.5 isCheckable()	2500
6.995.1.6 isLoadingState()	2500
6.995.1.7 restoreSelectionFromHistory()	2500
6.995.1.8 selectedLabels()	2501
6.996 Digikam::LanguagesList Class Reference	2501
6.997 Digikam::LcmsLock Class Reference	2502
6.997.1 Constructor & Destructor Documentation	2502
6.997.1.1 LcmsLock()	2502
6.998 Digikam::LensDistortionFilter Class Reference	2503
6.998.1 Member Function Documentation	2506
6.998.1.1 filterAction()	2506
6.998.1.2 filterIdentifier()	2506
6.998.1.3 readParameters()	2506
6.999 Digikam::LensDistortionPixelAccess Class Reference	2506
6.999.1 Detailed Description	2506
6.1000 Digikam::LensFunCameraSelector Class Reference	2507
6.1000.1 Member Function Documentation	2508
6.1000.1.1 setPassiveMetadataUsage()	2508
6.1001 Digikam::LensFunContainer Class Reference	2508



---

6.1002 Digikam::LensFunFilter Class Reference	2509
6.1002.1 Member Function Documentation	2512
6.1002.1.1 filterAction()	2512
6.1002.1.2 filterIdentifier()	2512
6.1002.1.3 readParameters()	2512
6.1003 Digikam::LensFunface Class Reference	2512
6.1003.1 Member Function Documentation	2513
6.1003.1.1 lensDescription()	2513
6.1003.1.2 makeDescription()	2513
6.1003.1.3 modelDescription()	2513
6.1004 Digikam::LensFunSettings Class Reference	2514
6.1005 Digikam::LessThanByProximityToSubject Class Reference	2515
6.1006 Digikam::LevelsContainer Class Reference	2515
6.1007 Digikam::LevelsFilter Class Reference	2516
6.1007.1 Member Function Documentation	2519
6.1007.1.1 filterAction()	2519
6.1007.1.2 filterIdentifier()	2519
6.1007.1.3 readParameters()	2519
6.1008 Digikam::LibsInfoDlg Class Reference	2520
6.1008.1 Constructor & Destructor Documentation	2521
6.1008.1.1 LibsInfoDlg()	2521
6.1009 Digikam::LightTablePreview Class Reference	2522
6.1010 Digikam::LightTableThumbBar Class Reference	2525
6.1011 Digikam::LightTableView Class Reference	2532
6.1012 Digikam::LightTableWindow Class Reference	2534
6.1012.1 Member Function Documentation	2536
6.1012.1.1 infoface()	2536
6.1012.1.2 loadItemInfos()	2537
6.1012.1.3 setLeftRightItems()	2537
6.1012.1.4 slotApplicationSettingsChanged	2537
6.1013 Digikam::LightTableWindow::Private Class Reference	2537
6.1014 Digikam::ListItem Class Reference	2539
6.1014.1 Member Function Documentation	2540
6.1014.1.1 containsItem()	2540
6.1015 Digikam::ListViewComboBox Class Reference	2541
6.1015.1 Constructor & Destructor Documentation	2542
6.1015.1.1 ListViewComboBox()	2542
6.1015.2 Member Function Documentation	2543
6.1015.2.1 installView()	2543
6.1015.2.2 sendViewportEventToView()	2543
6.1015.2.3 view()	2543
6.1016 Digikam::LoadingCache Class Reference	2544

---

6.1016.1 Member Function Documentation	2545
6.1016.1.1 addLoadingProcess()	2545
6.1016.1.2 fileChanged	2545
6.1016.1.3 getCacheSize()	2545
6.1016.1.4 isCacheable()	2546
6.1016.1.5 notifyFileChanged()	2546
6.1016.1.6 notifyNewLoadingProcess()	2546
6.1016.1.7 putImage()	2546
6.1016.1.8 putThumbnail()	2546
6.1016.1.9 removeImage()	2546
6.1016.1.10 removeImages()	2547
6.1016.1.11 removeLoadingProcess()	2547
6.1016.1.12 removeThumbnail()	2547
6.1016.1.13 removeThumbnails()	2547
6.1016.1.14 retrieveImage()	2547
6.1016.1.15 retrieveLoadingProcess()	2547
6.1016.1.16 retrieveThumbnail()	2547
6.1016.1.17 setCacheSize()	2548
6.1016.1.18 setFileWatch()	2548
6.1016.1.19 setThumbnailCacheSize()	2548
6.1017 Digikam::LoadingCache::CacheLock Class Reference	2548
6.1017.1 Detailed Description	2548
6.1018 Digikam::LoadingCacheFileWatch Class Reference	2549
6.1018.1 Member Function Documentation	2550
6.1018.1.1 notifyFileChanged()	2550
6.1019 Digikam::LoadingCacheInterface Class Reference	2550
6.1019.1 Member Function Documentation	2550
6.1019.1.1 cleanCache()	2550
6.1019.1.2 cleanThumbnailCache()	2550
6.1019.1.3 cleanUp()	2550
6.1019.1.4 connectToSignalFileChanged()	2551
6.1019.1.5 fileChanged()	2551
6.1019.1.6 putImage()	2551
6.1019.1.7 setCacheOptions()	2551
6.1020 Digikam::LoadingDescription Class Reference	2551
6.1020.1 Member Enumeration Documentation	2552
6.1020.1.1 ColorManagementSettings	2552
6.1020.1.2 RawDecodingHint	2552
6.1020.2 Constructor & Destructor Documentation	2553
6.1020.2.1 LoadingDescription() [1/4]	2553
6.1020.2.2 LoadingDescription() [2/4]	2553
6.1020.2.3 LoadingDescription() [3/4]	2553

6.1020.2.4 LoadingDescription() [ 4 / 4 ] . . . . .	2553
6.1020.3 Member Function Documentation . . . . .	2554
6.1020.3.1 cacheKey() . . . . .	2554
6.1020.3.2 equalsIgnoreReducedVersion() . . . . .	2554
6.1020.3.3 equalsOrBetterThan() . . . . .	2554
6.1020.3.4 isPreviewImage() . . . . .	2554
6.1020.3.5 isReducedVersion() . . . . .	2554
6.1020.3.6 isThumbnail() . . . . .	2554
6.1020.3.7 lookupCacheKeys() . . . . .	2554
6.1020.3.8 needCheckRawDecoding() . . . . .	2555
6.1020.3.9 operator==( ) . . . . .	2555
6.1020.3.10 possibleCacheKeys() . . . . .	2555
6.1020.3.11 thumbnailIdentifier() . . . . .	2555
6.1021 Digikam::LoadingDescription::PostProcessingParameters Class Reference . . . . .	2555
6.1022 Digikam::LoadingDescription::PreviewParameters Class Reference . . . . .	2556
6.1023 Digikam::LoadingProcess Class Reference . . . . .	2557
6.1024 Digikam::LoadingProcessListener Class Reference . . . . .	2558
6.1025 Digikam::LoadingTask Class Reference . . . . .	2559
6.1025.1 Member Function Documentation . . . . .	2560
6.1025.1.1 continueQuery() . . . . .	2560
6.1025.1.2 execute() . . . . .	2561
6.1025.1.3 progressInfo() . . . . .	2561
6.1025.1.4 type() . . . . .	2561
6.1026 Digikam::LoadSaveFileInfoProvider Class Reference . . . . .	2561
6.1026.1 Member Function Documentation . . . . .	2562
6.1026.1.1 dimensionsHint() . . . . .	2562
6.1026.1.2 orientationHint() . . . . .	2562
6.1027 Digikam::LoadSaveNotifier Class Reference . . . . .	2563
6.1027.1 Member Function Documentation . . . . .	2564
6.1027.1.1 thumbnailLoaded() . . . . .	2564
6.1028 Digikam::LoadSaveTask Class Reference . . . . .	2565
6.1029 Digikam::LoadSaveThread Class Reference . . . . .	2567
6.1029.1 Member Enumeration Documentation . . . . .	2569
6.1029.1.1 AccessMode . . . . .	2569
6.1029.1.2 NotificationPolicy . . . . .	2570
6.1029.2 Constructor & Destructor Documentation . . . . .	2570
6.1029.2.1 ~LoadSaveThread() . . . . .	2570
6.1029.3 Member Function Documentation . . . . .	2570
6.1029.3.1 exifOrientation() . . . . .	2570
6.1029.3.2 imageLoaded() . . . . .	2570
6.1029.3.3 imageSaved() . . . . .	2570
6.1029.3.4 imageStartedLoading() . . . . .	2571

---

6.1029.3.5 imageStartedSaving()	2571
6.1029.3.6 load()	2571
6.1029.3.7 loadingProgress()	2571
6.1029.3.8 moreCompleteLoadingAvailable()	2571
6.1029.3.9 run()	2571
6.1029.3.10 save()	2572
6.1029.3.11 savingProgress()	2572
6.1029.3.12 signalImageLoaded	2572
6.1029.3.13 signalImageStartedLoading	2572
6.1029.3.14 signalLoadingProgress	2572
6.1029.3.15 signalMoreCompleteLoadingAvailable	2573
6.1029.3.16 thumbnailLoaded()	2573
6.1030 Digikam::LocalContrastContainer Class Reference	2573
6.1031 Digikam::LocalContrastFilter Class Reference	2574
6.1031.1 Member Function Documentation	2577
6.1031.1.1 filterAction()	2577
6.1031.1.2 filterIdentifier()	2577
6.1031.1.3 readParameters()	2577
6.1032 Digikam::LocalContrastSettings Class Reference	2577
6.1033 Digikam::LocalizeConfig Class Reference	2578
6.1034 Digikam::LocalizeContainer Class Reference	2579
6.1034.1 Detailed Description	2579
6.1034.2 Member Data Documentation	2579
6.1034.2.1 ignoredWords	2579
6.1035 Digikam::LocalizeSelector Class Reference	2580
6.1036 Digikam::LocalizeSelectorList Class Reference	2581
6.1037 Digikam::LocalizeSettings Class Reference	2582
6.1037.1 Member Function Documentation	2583
6.1037.1.1 instance()	2583
6.1037.1.2 setSettings()	2583
6.1037.1.3 settings()	2583
6.1038 Digikam::LookupAltitude Class Reference	2584
6.1039 Digikam::LookupAltitude::Request Class Reference	2585
6.1040 Digikam::LookupAltitudeGeonames Class Reference	2586
6.1040.1 Member Function Documentation	2587
6.1040.1.1 backendHumanName()	2587
6.1040.1.2 backendName()	2587
6.1040.1.3 cancel()	2587
6.1040.1.4 errorMessage()	2588
6.1040.1.5 getRequest()	2588
6.1040.1.6 getRequests()	2588
6.1040.1.7 getStatus()	2588

---

6.1040.1.8 startLookup()	2588
6.1041 Digikam::LookupFactory Class Reference	2588
6.1042 Digikam::MaintenanceData Class Reference	2589
6.1043 Digikam::MaintenanceDlg Class Reference	2589
6.1044 Digikam::MaintenanceDlg::Private Class Reference	2590
6.1045 Digikam::MaintenanceMngr Class Reference	2591
6.1046 Digikam::MaintenanceSettings Class Reference	2592
6.1047 Digikam::MaintenanceThread Class Reference	2594
6.1047.1 Member Function Documentation	2596
6.1047.1.1 signalAddItemsToProcess	2596
6.1047.1.2 signalAdvance [1/2]	2596
6.1047.1.3 signalAdvance [2/2]	2596
6.1047.1.4 signalCanceled	2596
6.1047.1.5 signalCompleted	2596
6.1047.1.6 signalData	2596
6.1047.1.7 signalFinished	2597
6.1047.1.8 signalRemovePending	2597
6.1047.1.9 signalStarted	2597
6.1048 Digikam::MaintenanceTool Class Reference	2597
6.1048.1 Member Function Documentation	2599
6.1048.1.1 setNotificationEnabled()	2599
6.1048.1.2 setUseMultiCoreCPU()	2599
6.1048.1.3 signalCanceled	2600
6.1048.1.4 signalComplete	2600
6.1049 Digikam::MakerNoteWidget Class Reference	2601
6.1049.1 Member Function Documentation	2603
6.1049.1.1 getMetadataTitle()	2603
6.1049.1.2 getTagDescription()	2603
6.1049.1.3 getTagTitle()	2603
6.1049.1.4 loadFromURL()	2603
6.1050 Digikam::ManagedLoadSaveThread Class Reference	2604
6.1050.1 Member Enumeration Documentation	2607
6.1050.1.1 LoadingMode	2607
6.1050.1.2 LoadingPolicy	2608
6.1050.1.3 LoadingTaskFilter	2608
6.1050.1.4 TerminationPolicy	2608
6.1050.2 Constructor & Destructor Documentation	2609
6.1050.2.1 ManagedLoadSaveThread()	2609
6.1050.3 Member Function Documentation	2609
6.1050.3.1 load()	2609
6.1050.3.2 save()	2609
6.1050.3.3 setLoadingPolicy()	2609

---

6.1050.3.4 stopLoading() [1/2]	2609
6.1050.3.5 stopLoading() [2/2]	2609
6.1050.3.6 stopSaving()	2610
6.1051 Digikam::MapBackend Class Reference	2610
6.1051.1 Member Function Documentation	2612
6.1051.1.1 mapWidget()	2612
6.1051.1.2 mouseModeChanged()	2612
6.1051.1.3 setActive()	2612
6.1052 Digikam::MapDragData Class Reference	2612
6.1053 Digikam::MapDragDropHandler Class Reference	2613
6.1053.1 Member Function Documentation	2614
6.1053.1.1 accepts()	2614
6.1053.1.2 createMimeData()	2614
6.1053.1.3 dropEvent()	2614
6.1054 Digikam::MapViewModelHelper Class Reference	2615
6.1054.1 Member Function Documentation	2616
6.1054.1.1 bestRepresentativeIndexFromList()	2616
6.1054.1.2 itemCoordinates()	2617
6.1054.1.3 model()	2617
6.1054.1.4 onIndicesClicked()	2617
6.1054.1.5 pixmapFromRepresentativeIndex()	2618
6.1054.1.6 selectionModel()	2618
6.1055 Digikam::MapWidget Class Reference	2618
6.1055.1 Detailed Description	2622
6.1055.2 Constructor & Destructor Documentation	2622
6.1055.2.1 ~MapWidget()	2622
6.1055.3 Member Function Documentation	2622
6.1055.3.1 addUngroupedModel()	2622
6.1055.3.2 adjustBoundariesToGroupedMarkers()	2622
6.1055.3.3 applyCacheToBackend()	2623
6.1055.3.4 convertZoomToBackendZoom()	2623
6.1055.3.5 dragEnterEvent()	2623
6.1055.3.6 getColorInfos() [1/2]	2623
6.1055.3.7 getColorInfos() [2/2]	2623
6.1055.3.8 getDecoratedPixmapForCluster()	2624
6.1055.3.9 removeUngroupedModel()	2624
6.1055.3.10 setBackend()	2624
6.1055.3.11 setGroupedModel()	2624
6.1055.3.12 setSortKey()	2624
6.1055.3.13 setThumbnailSize()	2624
6.1055.3.14 slotClustersClicked	2624
6.1055.3.15 slotClustersMoved	2624

6.1055.3.16 slotItemDisplaySettingsChanged	2625
6.1055.3.17 slotMouseModeChanged	2625
6.1055.3.18 slotNewSelectionFromMap	2625
6.1055.3.19 slotUpdateActionsEnabled	2625
6.1055.3.20 updateClusters()	2625
6.1056 Digikam::MapWidget::Private Class Reference	2625
6.1057 Digikam::MapWidgetView Class Reference	2626
6.1057.1 Constructor & Destructor Documentation	2628
6.1057.1.1 MapWidgetView()	2628
6.1057.2 Member Function Documentation	2629
6.1057.2.1 currentCamItemInfo()	2629
6.1057.2.2 currentItemInfo()	2629
6.1057.2.3 doLoadState()	2629
6.1057.2.4 doSaveState()	2629
6.1057.2.5 getActiveState()	2629
6.1057.2.6 setActive()	2629
6.1058 Digikam::Mat Struct Reference	2630
6.1058.1 Detailed Description	2630
6.1058.2 Member Data Documentation	2630
6.1058.2.1 cols	2630
6.1058.2.2 data	2630
6.1058.2.3 rows	2630
6.1059 Digikam::MdKeyListItem Class Reference	2631
6.1060 Digikam::MediaPlayerView Class Reference	2632
6.1061 Digikam::MetadataHub Class Reference	2633
6.1061.1 Member Enumeration Documentation	2634
6.1061.1.1 Status	2634
6.1061.1.2 WriteMode	2634
6.1061.2 Constructor & Destructor Documentation	2634
6.1061.2.1 MetadataHub()	2634
6.1061.3 Member Function Documentation	2635
6.1061.3.1 cleanupTags()	2635
6.1061.3.2 load() [1/2]	2635
6.1061.3.3 load() [2/2]	2635
6.1061.3.4 loadTags()	2635
6.1061.3.5 willWriteMetadata()	2635
6.1061.3.6 write() [1/3]	2636
6.1061.3.7 write() [2/3]	2636
6.1061.3.8 write() [3/3]	2637
6.1061.3.9 writeTags() [1/2]	2637
6.1061.3.10 writeTags() [2/2]	2638
6.1061.3.11 writeToBaloo()	2638

6.1061.3.12 writeToMetadata()	2638
6.1062 Digikam::MetadataHubMngr Class Reference	2639
6.1063 Digikam::MetadataKeys Class Reference	2640
6.1063.1 Member Function Documentation	2641
6.1063.1.1 getDbValue()	2641
6.1064 Digikam::MetadataListView Class Reference	2642
6.1065 Digikam::MetadataListViewItem Class Reference	2643
6.1066 Digikam::MetadataOption Class Reference	2644
6.1066.1 Member Function Documentation	2645
6.1066.1.1 parseOperation()	2645
6.1067 Digikam::MetadataOptionDialog Class Reference	2646
6.1068 Digikam::MetadataPage Class Reference	2647
6.1069 Digikam::MetadataPanel Class Reference	2648
6.1070 Digikam::MetadataRemover Class Reference	2650
6.1070.1 Constructor & Destructor Documentation	2653
6.1070.1.1 MetadataRemover() [1/2]	2653
6.1070.1.2 MetadataRemover() [2/2]	2653
6.1070.2 Member Function Documentation	2653
6.1070.2.1 setUseMultiCoreCPU()	2653
6.1071 Digikam::MetadataRemoveTask Class Reference	2654
6.1072 Digikam::MetadataSelector Class Reference	2655
6.1073 Digikam::MetadataSelectorItem Class Reference	2656
6.1074 Digikam::MetadataSelectorView Class Reference	2657
6.1075 Digikam::MetadataStatusBar Class Reference	2658
6.1076 Digikam::MetadataSynchronizer Class Reference	2659
6.1076.1 Constructor & Destructor Documentation	2662
6.1076.1.1 MetadataSynchronizer() [1/2]	2662
6.1076.1.2 MetadataSynchronizer() [2/2]	2662
6.1076.2 Member Function Documentation	2662
6.1076.2.1 setUseMultiCoreCPU()	2662
6.1077 Digikam::MetadataSyncTask Class Reference	2663
6.1078 Digikam::MetadataWidget Class Reference	2665
6.1078.1 Member Function Documentation	2666
6.1078.1.1 setup()	2666
6.1079 Digikam::MetaEngine Class Reference	2667
6.1079.1 Member Typedef Documentation	2672
6.1079.1.1 AltLangMap	2672
6.1079.1.2 MetaDataMap	2672
6.1079.1.3 TagsMap	2672
6.1079.2 Member Enumeration Documentation	2673
6.1079.2.1 Backend	2673
6.1079.2.2 ImageColorWorkSpace	2673



---

6.1079.2.3 ImageOrientation	2673
6.1079.2.4 MetadataWritingMode	2673
6.1079.2.5 XmpTagType	2674
6.1079.3 Constructor & Destructor Documentation	2674
6.1079.3.1 MetaEngine() [1/3]	2674
6.1079.3.2 MetaEngine() [2/3]	2674
6.1079.3.3 MetaEngine() [3/3]	2674
6.1079.3.4 ~MetaEngine()	2674
6.1079.4 Member Function Documentation	2674
6.1079.4.1 addToXmpTagStringBag()	2674
6.1079.4.2 applyChanges()	2675
6.1079.4.3 backendName()	2675
6.1079.4.4 canWriteComment()	2675
6.1079.4.5 canWriteExif()	2675
6.1079.4.6 canWriteIptc()	2675
6.1079.4.7 canWriteXmp()	2675
6.1079.4.8 clearComments()	2675
6.1079.4.9 clearExif()	2676
6.1079.4.10 clearIptc()	2676
6.1079.4.11 clearXmp()	2676
6.1079.4.12 convertDegreeAngleToDouble()	2676
6.1079.4.13 convertFromGPSCoordinateString() [1/2]	2676
6.1079.4.14 convertFromGPSCoordinateString() [2/2]	2676
6.1079.4.15 convertToGPSCoordinateString() [1/2]	2677
6.1079.4.16 convertToGPSCoordinateString() [2/2]	2677
6.1079.4.17 convertToRational()	2677
6.1079.4.18 convertToRationalSmallDenominator()	2677
6.1079.4.19 convertToUserPresentableNumbers() [1/2]	2678
6.1079.4.20 convertToUserPresentableNumbers() [2/2]	2678
6.1079.4.21 createExifUserStringFromValue()	2678
6.1079.4.22 detectLanguageAlt()	2678
6.1079.4.23 Exiv2Version()	2678
6.1079.4.24 exportChanges()	2679
6.1079.4.25 getComments()	2679
6.1079.4.26 getCommentsDecoded()	2679
6.1079.4.27 getDigitizationDateTime()	2679
6.1079.4.28 getExifComment()	2679
6.1079.4.29 getExifEncoded()	2679
6.1079.4.30 getExifTagComment()	2680
6.1079.4.31 getExifTagData()	2680
6.1079.4.32 getExifTagDescription()	2680
6.1079.4.33 getExifTagLong() [1/2]	2680

---

6.1079.4.34 getExifTagLong() [2/2]	2680
6.1079.4.35 getExifTagRational()	2680
6.1079.4.36 getExifTagsDataList()	2681
6.1079.4.37 getExifTagString()	2681
6.1079.4.38 getExifTagTitle()	2681
6.1079.4.39 getExifTagVariant()	2681
6.1079.4.40 getExifThumbnail()	2682
6.1079.4.41 getFilePath()	2682
6.1079.4.42 getGPSAltitude()	2682
6.1079.4.43 getGPSInfo()	2682
6.1079.4.44 getGPSLatitudeNumber()	2682
6.1079.4.45 getGPSLatitudeString()	2682
6.1079.4.46 getIptc()	2683
6.1079.4.47 getIptcKeywords()	2683
6.1079.4.48 getIptcSubCategories()	2683
6.1079.4.49 getIptcSubjects()	2683
6.1079.4.50 getIptcTagData()	2683
6.1079.4.51 getIptcTagDescription()	2683
6.1079.4.52 getIptcTagsDataList()	2684
6.1079.4.53 getIptcTagsList()	2684
6.1079.4.54 getIptcTagsStringList()	2684
6.1079.4.55 getIptcTagString()	2684
6.1079.4.56 getIptcTagTitle()	2684
6.1079.4.57 getItemColorWorkSpace()	2685
6.1079.4.58 getItemDateTime()	2685
6.1079.4.59 getItemDimensions()	2685
6.1079.4.60 getItemIccProfile()	2685
6.1079.4.61 getItemOrientation()	2685
6.1079.4.62 getItemPreview()	2685
6.1079.4.63 getMakernoteTagsList()	2685
6.1079.4.64 getMimeType()	2686
6.1079.4.65 getPixelSize()	2686
6.1079.4.66 getStdExifTagsList()	2686
6.1079.4.67 getXmp()	2686
6.1079.4.68 getXmpKeywords()	2686
6.1079.4.69 getXmpSubCategories()	2686
6.1079.4.70 getXmpSubjects()	2686
6.1079.4.71 getXmpTagDescription()	2687
6.1079.4.72 getXmpTagsDataList()	2687
6.1079.4.73 getXmpTagsList()	2687
6.1079.4.74 getXmpTagString()	2687
6.1079.4.75 getXmpTagStringBag()	2687

---

6.1079.4.76 getXmpTagStringLangAlt()	2688
6.1079.4.77 getXmpTagStringListLangAlt()	2688
6.1079.4.78 getXmpTagStringSeq()	2688
6.1079.4.79 getXmpTagTitle()	2688
6.1079.4.80 getXmpTagVariant()	2688
6.1079.4.81 hasComments()	2689
6.1079.4.82 hasExif()	2689
6.1079.4.83 hasIptc()	2689
6.1079.4.84 hasSidecar()	2689
6.1079.4.85 hasXmp()	2689
6.1079.4.86 initializeExiv2()	2689
6.1079.4.87 initializeGPSInfo()	2689
6.1079.4.88 isEmpty()	2690
6.1079.4.89 load()	2690
6.1079.4.90 loadFromData()	2690
6.1079.4.91 loadFromDataAndMerge()	2690
6.1079.4.92 loadFromSidecarAndMerge()	2690
6.1079.4.93 metadataWritingMode()	2690
6.1079.4.94 readWithExifTool()	2691
6.1079.4.95 registerXmpNameSpace()	2691
6.1079.4.96 removeExifTag()	2691
6.1079.4.97 removeExifThumbnail()	2691
6.1079.4.98 removeFromXmpTagStringBag()	2691
6.1079.4.99 removeGPSInfo()	2691
6.1079.4.100 removeIptcTag()	2692
6.1079.4.101 removeXmpKeywords()	2692
6.1079.4.102 removeXmpSubCategories()	2692
6.1079.4.103 removeXmpSubjects()	2692
6.1079.4.104 removeXmpTag()	2692
6.1079.4.105 rotateExifQImage()	2692
6.1079.4.106 save()	2693
6.1079.4.107 setComments()	2693
6.1079.4.108 setExif()	2693
6.1079.4.109 setExifComment()	2693
6.1079.4.110 setExifTagData()	2693
6.1079.4.111 setExifTagLong()	2693
6.1079.4.112 setExifTagRational()	2694
6.1079.4.113 setExifTagString()	2694
6.1079.4.114 setExifTagURational()	2694
6.1079.4.115 setExifTagUShort()	2694
6.1079.4.116 setExifTagVariant()	2694
6.1079.4.117 setExifThumbnail()	2695

6.1079.4.118 setFilePath()	2695
6.1079.4.119 setGPSInfo() [1/3]	2695
6.1079.4.120 setGPSInfo() [2/3]	2695
6.1079.4.121 setGPSInfo() [3/3]	2695
6.1079.4.122 setImageDateTime()	2695
6.1079.4.123 setIptc()	2696
6.1079.4.124 setIptcKeywords()	2696
6.1079.4.125 setIptcSubCategories()	2696
6.1079.4.126 setIptcSubjects()	2696
6.1079.4.127 setIptcTagData()	2696
6.1079.4.128 setIptcTagsStringList()	2697
6.1079.4.129 setIptcTagString()	2697
6.1079.4.130 setItemColorWorkSpace()	2697
6.1079.4.131 setItemDimensions()	2697
6.1079.4.132 setItemIccProfile()	2697
6.1079.4.133 setItemOrientation()	2697
6.1079.4.134 setItemPreview()	2698
6.1079.4.135 setItemProgramId()	2698
6.1079.4.136 setMetadataWritingMode()	2698
6.1079.4.137 setProgramId()	2698
6.1079.4.138 setReadWithExifTool()	2698
6.1079.4.139 setTiffThumbnail()	2699
6.1079.4.140 setUpdateFileTimeStamp()	2699
6.1079.4.141 setUseCompatibleFileName()	2699
6.1079.4.142 setUseXMPSidecar4Reading()	2699
6.1079.4.143 setWriteDngFiles()	2699
6.1079.4.144 setWriteRawFiles()	2699
6.1079.4.145 setWriteWithExifTool()	2699
6.1079.4.146 setXmp()	2700
6.1079.4.147 setXmpKeywords()	2700
6.1079.4.148 setXmpSubCategories()	2700
6.1079.4.149 setXmpSubjects()	2700
6.1079.4.150 setXmpTagString() [1/2]	2700
6.1079.4.151 setXmpTagString() [2/2]	2700
6.1079.4.152 setXmpTagStringBag()	2701
6.1079.4.153 setXmpTagStringLangAlt()	2701
6.1079.4.154 setXmpTagStringListLangAlt()	2701
6.1079.4.155 setXmpTagStringSeq()	2701
6.1079.4.156 sidecarFilePathForFile()	2701
6.1079.4.157 sidecarPath()	2701
6.1079.4.158 sidecarUrl() [1/2]	2702
6.1079.4.159 sidecarUrl() [2/2]	2702

6.1079.4.160 supportBmff()	2702
6.1079.4.161 supportJpegXL()	2702
6.1079.4.162 supportMetadataWriting()	2702
6.1079.4.163 supportXmp()	2702
6.1079.4.164 unregisterXmpNameSpace()	2702
6.1079.4.165 updateFileTimeStamp()	2703
6.1079.4.166 useCompatibleFileName()	2703
6.1079.4.167 useXMPSidecar4Reading()	2703
6.1079.4.168 writeDngFiles()	2703
6.1079.4.169 writeRawFiles()	2703
6.1079.4.170 writeWithExifTool()	2703
6.1080 Digikam::MetaEngine::Private Class Reference	2704
6.1080.1 Member Function Documentation	2705
6.1080.1.1 convertCommentValue()	2705
6.1080.1.2 decodeGPSCoordinate()	2705
6.1080.1.3 detectEncodingAndDecode()	2705
6.1080.1.4 extractIptcTagString()	2705
6.1080.1.5 isUtf8()	2705
6.1080.1.6 printExiv2ExceptionError()	2705
6.1080.1.7 printExiv2MessageHandler()	2706
6.1081 Digikam::MetaEngineData Class Reference	2706
6.1082 Digikam::MetaEngineData::Private Class Reference	2706
6.1083 Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList > Class Template Reference	2707
6.1083.1 Member Function Documentation	2707
6.1083.1.1 exclusiveMerge()	2707
6.1083.1.2 mergeAll()	2707
6.1083.1.3 mergeFields()	2708
6.1084 Digikam::MetaEnginePreviews Class Reference	2708
6.1084.1 Constructor & Destructor Documentation	2708
6.1084.1.1 MetaEnginePreviews() [1/2]	2708
6.1084.1.2 MetaEnginePreviews() [2/2]	2708
6.1084.2 Member Function Documentation	2709
6.1084.2.1 data()	2709
6.1084.2.2 dataSize()	2709
6.1084.2.3 image()	2709
6.1085 Digikam::MetaEngineRotation Class Reference	2709
6.1085.1 Member Enumeration Documentation	2710
6.1085.1.1 TransformationAction	2710
6.1085.2 Constructor & Destructor Documentation	2710
6.1085.2.1 MetaEngineRotation() [1/3]	2710
6.1085.2.2 MetaEngineRotation() [2/3]	2710

---

6.1085.2.3 MetaEngineRotation() [3/3]	2711
6.1085.3 Member Function Documentation	2711
6.1085.3.1 exifOrientation()	2711
6.1085.3.2 isNoTransform()	2711
6.1085.3.3 operator*=() [1/3]	2711
6.1085.3.4 operator*=() [2/3]	2711
6.1085.3.5 operator*=() [3/3]	2711
6.1085.3.6 toTransform() [1/2]	2711
6.1085.3.7 toTransform() [2/2]	2712
6.1085.3.8 transformations()	2712
6.1086 Digikam::MetaEngineSettings Class Reference	2712
6.1086.1 Member Function Documentation	2713
6.1086.1.1 exifRotate()	2713
6.1086.1.2 instance()	2713
6.1086.1.3 setSettings()	2713
6.1086.1.4 settings()	2713
6.1087 Digikam::MetaEngineSettingsContainer Class Reference	2713
6.1087.1 Detailed Description	2714
6.1087.2 Member Enumeration Documentation	2715
6.1087.2.1 RotationBehaviorFlag	2715
6.1088 Digikam::MigrateFromDigikam4Page Class Reference	2716
6.1088.1 Member Function Documentation	2717
6.1088.1.1 checkForMigration()	2717
6.1088.1.2 isMigrationChecked()	2717
6.1089 Digikam::MimeFilter Class Reference	2718
6.1089.1 Member Enumeration Documentation	2718
6.1089.1.1 TypeMimeFilter	2718
6.1090 Digikam::MixerContainer Class Reference	2719
6.1091 Digikam::MixerFilter Class Reference	2720
6.1091.1 Member Function Documentation	2723
6.1091.1.1 filterAction()	2723
6.1091.1.2 filterIdentifier()	2723
6.1091.1.3 readParameters()	2723
6.1092 Digikam::MixerSettings Class Reference	2724
6.1093 Digikam::MLClassifierFoundation Class Reference	2725
6.1094 Digikam::MLClassifierFoundation::VotingGroups Class Reference	2726
6.1095 Digikam::MLClassifierFoundation::VotingGroups::VoteTally Struct Reference	2726
6.1096 Digikam::MLPipelineFoundation Class Reference	2727
6.1096.1 Member Enumeration Documentation	2729
6.1096.1.1 MLPipelineStage	2729
6.1096.2 Member Function Documentation	2729
6.1096.2.1 cancel()	2729

---

6.1097 Digikam::MLPipelineFoundation::_MLPipelinePerformanceProfile Struct Reference . . . . .	2730
6.1098 Digikam::MLPipelinePackageFoundation Class Reference . . . . .	2731
6.1099 Digikam::MLPipelinePackageNotify Class Reference . . . . .	2732
6.1100 Digikam::ModelCompleter Class Reference . . . . .	2733
6.1100.1 Member Function Documentation . . . . .	2733
6.1100.1.1 setItemModel() . . . . .	2733
6.1101 Digikam::ModelIndexBasedComboBox Class Reference . . . . .	2735
6.1101.1 Constructor & Destructor Documentation . . . . .	2736
6.1101.1.1 ModelIndexedComboBox() . . . . .	2736
6.1102 Digikam::ModelMenu Class Reference . . . . .	2737
6.1102.1 Detailed Description . . . . .	2738
6.1102.2 Member Function Documentation . . . . .	2738
6.1102.2.1 prePopulated() . . . . .	2738
6.1103 Digikam::Modifier Class Reference . . . . .	2739
6.1103.1 Member Function Documentation . . . . .	2740
6.1103.1.1 parseOperation() . . . . .	2740
6.1104 Digikam::MonthWidget Class Reference . . . . .	2741
6.1105 Digikam::MysqlAdminBinary Class Reference . . . . .	2742
6.1106 Digikam::MysqlInitBinary Class Reference . . . . .	2745
6.1107 Digikam::MysqlServerBinary Class Reference . . . . .	2748
6.1108 Digikam::MysqlUpgradeBinary Class Reference . . . . .	2751
6.1109 Digikam::NamespaceEditDlg Class Reference . . . . .	2753
6.1110 Digikam::NamespaceEntry Class Reference . . . . .	2754
6.1110.1 Member Data Documentation . . . . .	2755
6.1110.1.1 convertRatio . . . . .	2755
6.1110.1.2 namespaceName . . . . .	2755
6.1111 Digikam::NamespaceListView Class Reference . . . . .	2756
6.1111.1 Member Function Documentation . . . . .	2757
6.1111.1.1 startDrag() . . . . .	2757
6.1112 Digikam::NetworkManager Class Reference . . . . .	2757
6.1112.1 Member Function Documentation . . . . .	2758
6.1112.1.1 getNetworkManager() . . . . .	2758
6.1112.1.2 instance() . . . . .	2758
6.1113 Digikam::NewItemFinder Class Reference . . . . .	2759
6.1113.1 Member Enumeration Documentation . . . . .	2761
6.1113.1.1 FinderMode . . . . .	2761
6.1114 Digikam::NewlyAppearedFile Class Reference . . . . .	2762
6.1115 Digikam::NoDuplicatesImportFilterModel Class Reference . . . . .	2763
6.1116 Digikam::NoDuplicatesItemFilterModel Class Reference . . . . .	2766
6.1117 Digikam::NoiseDetector Class Reference . . . . .	2768
6.1117.1 Member Function Documentation . . . . .	2769
6.1117.1.1 detect() . . . . .	2769

6.1118 Digikam::NonDeterministicRandomData Class Reference	2769
6.1118.1 Constructor & Destructor Documentation	2770
6.1118.1.1 NonDeterministicRandomData()	2770
6.1119 Digikam::NormalizeFilter Class Reference	2771
6.1119.1 Member Function Documentation	2774
6.1119.1.1 filterAction()	2774
6.1119.1.2 filterIdentifier()	2774
6.1119.1.3 readParameters()	2774
6.1120 Digikam::NormalSearchTreeView Class Reference	2775
6.1120.1 Detailed Description	2779
6.1120.2 Constructor & Destructor Documentation	2779
6.1120.2.1 NormalSearchTreeView()	2779
6.1120.2.2 ~NormalSearchTreeView()	2780
6.1120.3 Member Function Documentation	2780
6.1120.3.1 addCustomContextMenuActions()	2780
6.1120.3.2 copySearch	2780
6.1120.3.3 editSearch	2780
6.1120.3.4 handleCustomContextMenuAction()	2781
6.1120.3.5 newSearch	2781
6.1121 Digikam::NRContainer Class Reference	2781
6.1121.1 Member Data Documentation	2781
6.1121.1.1 thresholds	2781
6.1122 Digikam::NREstimate Class Reference	2782
6.1122.1 Constructor & Destructor Documentation	2785
6.1122.1.1 NREstimate()	2785
6.1122.2 Member Function Documentation	2785
6.1122.2.1 setLogFilesPath()	2785
6.1122.2.2 settings()	2785
6.1122.2.3 startAnalyse()	2785
6.1123 Digikam::NRFilter Class Reference	2786
6.1123.1 Member Function Documentation	2789
6.1123.1.1 filterAction()	2789
6.1123.1.2 filterIdentifier()	2789
6.1123.1.3 readParameters()	2789
6.1124 Digikam::NRSettings Class Reference	2790
6.1125 Digikam::OilPaintFilter Class Reference	2791
6.1125.1 Member Function Documentation	2794
6.1125.1.1 filterAction()	2794
6.1125.1.2 filterIdentifier()	2794
6.1125.1.3 readParameters()	2794
6.1126 Digikam::OnlineVersionChecker Class Reference	2794
6.1126.1 Member Function Documentation	2795



---

6.1126.1.1 bundleProperties()	2795
6.1126.1.2 lastCheckDate()	2795
6.1127 Digikam::OnlineVersionDlg Class Reference	2796
6.1128 Digikam::OnlineVersionDwnl Class Reference	2797
6.1129 Digikam::OpenCVDNNFaceDetector Class Reference	2797
6.1129.1 Member Function Documentation	2798
6.1129.1.1 detectFaces()	2798
6.1129.1.2 recommendedImageSizeForDetection()	2799
6.1130 Digikam::OpenCVDNNFaceRecognizer Class Reference	2799
6.1130.1 Member Enumeration Documentation	2799
6.1130.1.1 Classifier	2799
6.1130.2 Member Function Documentation	2800
6.1130.2.1 clearTraining()	2800
6.1130.2.2 prepareForRecognition() [1/2]	2800
6.1130.2.3 prepareForRecognition() [2/2]	2800
6.1130.2.4 recognize() [1/2]	2800
6.1130.2.5 recognize() [2/2]	2800
6.1130.2.6 remove()	2801
6.1130.2.7 setNbNeighbors()	2801
6.1130.2.8 setThreshold()	2801
6.1130.2.9 train()	2801
6.1131 Digikam::OpenCVDNNFaceRecognizer::Private Class Reference	2801
6.1132 Digikam::OpenCVDNNFaceRecognizer::Private::ParallelRecognizer Class Reference	2802
6.1133 Digikam::OpenCVDNNFaceRecognizer::Private::ParallelTrainer Class Reference	2803
6.1134 Digikam::OpenfacePreprocessor Class Reference	2803
6.1134.1 Member Function Documentation	2803
6.1134.1.1 loadModels()	2803
6.1135 Digikam::OpenFilePage Class Reference	2804
6.1136 Digikam::Option Class Reference	2805
6.1136.1 Member Function Documentation	2806
6.1136.1.1 parseOperation()	2806
6.1137 Digikam::OverlayWidget Class Reference	2808
6.1137.1 Detailed Description	2809
6.1138 Digikam::PackageLoadingDescriptionList Class Reference	2809
6.1139 Digikam::PageItem Class Reference	2810
6.1140 Digikam::PALbum Class Reference	2811
6.1140.1 Detailed Description	2813
6.1140.2 Member Function Documentation	2813
6.1140.2.1 databaseUrl()	2813
6.1141 Digikam::PALbumPath Class Reference	2813
6.1142 Digikam::PanIconFrame Class Reference	2814
6.1142.1 Detailed Description	2815

6.1142.2 Member Function Documentation	2815
6.1142.2.1 close	2815
6.1142.2.2 exec() [1/2]	2815
6.1142.2.3 exec() [2/2]	2815
6.1142.2.4 keyPressEvent()	2815
6.1142.2.5 popup()	2816
6.1142.2.6 resizeEvent()	2816
6.1142.2.7 setMainWidget()	2816
6.1143 Digikam::PanIconWidget Class Reference	2817
6.1143.1 Member Function Documentation	2818
6.1143.1.1 regionSelectionMoved()	2818
6.1143.1.2 signalSelectionMoved	2818
6.1144 Digikam::ParallelAdapter< A > Class Template Reference	2819
6.1144.1 Constructor & Destructor Documentation	2821
6.1144.1.1 ParallelAdapter()	2821
6.1144.2 Member Function Documentation	2821
6.1144.2.1 asQObject()	2821
6.1144.2.2 mocMetaObject()	2821
6.1144.2.3 staticMetacallPointer()	2821
6.1144.2.4 WorkerObjectQtMetacall()	2822
6.1145 Digikam::ParallelPipes Class Reference	2822
6.1146 Digikam::ParallelWorkers Class Reference	2824
6.1146.1 Constructor & Destructor Documentation	2825
6.1146.1.1 ParallelWorkers()	2825
6.1146.2 Member Function Documentation	2825
6.1146.2.1 asQObject()	2825
6.1146.2.2 mocMetaObject()	2826
6.1146.2.3 optimalWorkerCount()	2826
6.1146.2.4 optimalWorkerCountReached()	2826
6.1146.2.5 replacementQtMetacall()	2826
6.1146.2.6 schedule()	2826
6.1146.2.7 WorkerObjectQtMetacall()	2826
6.1147 Digikam::Parser Class Reference	2827
6.1147.1 Member Function Documentation	2828
6.1147.1.1 parseStringsValid()	2828
6.1148 Digikam::ParseResults Class Reference	2828
6.1149 Digikam::ParseSettings Class Reference	2829
6.1150 Digikam::PeopleSideBarWidget Class Reference	2830
6.1150.1 Member Function Documentation	2832
6.1150.1.1 applySettings()	2832
6.1150.1.2 changeAlbumFromHistory()	2832
6.1150.1.3 doLoadState()	2832

---

6.1150.1.4 doSaveState()	2832
6.1150.1.5 getCaption()	2832
6.1150.1.6 getIcon()	2833
6.1150.1.7 setActive()	2833
6.1151 Digikam::PersistentWidgetDelegateOverlay Class Reference	2834
6.1151.1 Constructor & Destructor Documentation	2836
6.1151.1.1 PersistentWidgetDelegateOverlay()	2836
6.1151.2 Member Function Documentation	2836
6.1151.2.1 hide()	2836
6.1151.2.2 setActive()	2837
6.1151.2.3 setFocusOnWidget()	2837
6.1151.2.4 setPersistent	2837
6.1151.2.5 showOnIndex()	2837
6.1151.2.6 slotEntered()	2837
6.1151.2.7 slotLayoutChanged()	2837
6.1151.2.8 slotReset()	2838
6.1151.2.9 slotRowsRemoved()	2838
6.1151.2.10 slotViewportEntered()	2838
6.1151.2.11 viewportLeaveEvent()	2838
6.1152 Digikam::PhotoInfoContainer Class Reference	2838
6.1153 Digikam::PickLabelFilter Class Reference	2840
6.1154 Digikam::PickLabelMenuAction Class Reference	2842
6.1155 Digikam::PickLabelSelector Class Reference	2843
6.1156 Digikam::PickLabelWidget Class Reference	2844
6.1156.1 Member Function Documentation	2845
6.1156.1.1 pickLabels()	2845
6.1156.1.2 setButtonsExclusive()	2846
6.1156.1.3 setDescriptionBoxVisible()	2846
6.1156.1.4 setPickLabels()	2846
6.1157 Digikam::PlaceholderWidget Class Reference	2846
6.1158 Digikam::PointTransformAffine Class Reference	2847
6.1159 Digikam::PositionKeys Class Reference	2847
6.1159.1 Constructor & Destructor Documentation	2848
6.1159.1.1 PositionKeys()	2848
6.1159.2 Member Function Documentation	2848
6.1159.2.1 getDbValue()	2848
6.1160 Digikam::PreviewList Class Reference	2849
6.1161 Digikam::PreviewListItem Class Reference	2850
6.1162 Digikam::PreviewLoadingTask Class Reference	2851
6.1162.1 Member Function Documentation	2853
6.1162.1.1 execute()	2853
6.1163 Digikam::PreviewLoadThread Class Reference	2854

6.1163.1 Constructor & Destructor Documentation	2858
6.1163.1.1 PreviewLoadThread()	2858
6.1163.2 Member Function Documentation	2858
6.1163.2.1 load() [1/2]	2858
6.1163.2.2 load() [2/2]	2858
6.1163.2.3 loadFast()	2859
6.1163.2.4 loadFastButLarge()	2859
6.1163.2.5 loadFastSynchronously()	2859
6.1163.2.6 loadHighQuality()	2859
6.1163.2.7 setDisplayingWidget()	2859
6.1164 Digikam::PreviewPage Class Reference	2860
6.1165 Digikam::PreviewSettings Class Reference	2861
6.1165.1 Member Enumeration Documentation	2861
6.1165.1.1 Quality	2861
6.1166 Digikam::PreviewThreadWrapper Class Reference	2862
6.1167 Digikam::PreviewToolBar Class Reference	2863
6.1167.1 Member Enumeration Documentation	2864
6.1167.1.1 PreviewMode	2864
6.1168 Digikam::PrivateProgressItemCreator Class Reference	2864
6.1168.1 Member Function Documentation	2865
6.1168.1.1 addProgressItem()	2865
6.1168.1.2 createProgressItem()	2865
6.1169 Digikam::ProcessLauncher Class Reference	2866
6.1169.1 Member Function Documentation	2867
6.1169.1.1 elapsedTime()	2867
6.1169.1.2 exitCode()	2867
6.1169.1.3 output()	2867
6.1169.1.4 setConsoleTraces()	2867
6.1169.1.5 startProcess()	2867
6.1169.1.6 success()	2868
6.1170 Digikam::ProgressEntry Class Reference	2868
6.1171 Digikam::ProgressItem Class Reference	2868
6.1171.1 Member Function Documentation	2870
6.1171.1.1 advance()	2870
6.1171.1.2 canBeCanceled()	2870
6.1171.1.3 hasThumbnail()	2870
6.1171.1.4 id()	2870
6.1171.1.5 label()	2870
6.1171.1.6 parent()	2871
6.1171.1.7 progress()	2871
6.1171.1.8 progressItemAdded	2871
6.1171.1.9 progressItemCanceled	2871

6.1171.1.10	<a href="#">progressItemCompleted</a>	2871
6.1171.1.11	<a href="#">progressItemLabel</a>	2872
6.1171.1.12	<a href="#">progressItemProgress</a>	2872
6.1171.1.13	<a href="#">progressItemStatus</a>	2872
6.1171.1.14	<a href="#">progressItemThumbnail</a>	2872
6.1171.1.15	<a href="#">progressItemUsesBusyIndicator</a>	2873
6.1171.1.16	<a href="#">setLabel()</a>	2873
6.1171.1.17	<a href="#">setProgress()</a>	2873
6.1171.1.18	<a href="#">setShowAtStart()</a>	2873
6.1171.1.19	<a href="#">setStatus()</a>	2873
6.1171.1.20	<a href="#">setThumbnail()</a>	2874
6.1171.1.21	<a href="#">setUsesBusyIndicator()</a>	2874
6.1171.1.22	<a href="#">showAtStart()</a>	2874
6.1171.1.23	<a href="#">status()</a>	2874
6.1171.1.24	<a href="#">usesBusyIndicator()</a>	2874
6.1172	<a href="#">Digikam::ProgressManager Class Reference</a>	2875
6.1172.1	<a href="#">Detailed Description</a>	2877
6.1172.2	<a href="#">Member Function Documentation</a>	2877
6.1172.2.1	<a href="#">addProgressItem()</a>	2877
6.1172.2.2	<a href="#">createProgressItem()</a> [1/4]	2877
6.1172.2.3	<a href="#">createProgressItem()</a> [2/4]	2878
6.1172.2.4	<a href="#">createProgressItem()</a> [3/4]	2878
6.1172.2.5	<a href="#">createProgressItem()</a> [4/4]	2878
6.1172.2.6	<a href="#">findItemById()</a>	2879
6.1172.2.7	<a href="#">getUniqueID()</a>	2879
6.1172.2.8	<a href="#">instance()</a>	2879
6.1172.2.9	<a href="#">isEmpty()</a>	2880
6.1172.2.10	<a href="#">progressItemAdded</a>	2880
6.1172.2.11	<a href="#">progressItemCanceled</a>	2880
6.1172.2.12	<a href="#">progressItemCompleted</a>	2880
6.1172.2.13	<a href="#">progressItemLabel</a>	2880
6.1172.2.14	<a href="#">progressItemProgress</a>	2881
6.1172.2.15	<a href="#">progressItemStatus</a>	2881
6.1172.2.16	<a href="#">progressItemThumbnail</a>	2881
6.1172.2.17	<a href="#">progressItemUsesBusyIndicator</a>	2881
6.1172.2.18	<a href="#">singleItem()</a>	2881
6.1172.2.19	<a href="#">slotStandardCancelHandler</a>	2881
6.1173	<a href="#">Digikam::ProgressView Class Reference</a>	2882
6.1174	<a href="#">Digikam::ProxyClickLineEdit Class Reference</a>	2885
6.1174.1	<a href="#">Constructor &amp; Destructor Documentation</a>	2886
6.1174.1.1	<a href="#">ProxyClickLineEdit()</a>	2886
6.1175	<a href="#">Digikam::ProxyLineEdit Class Reference</a>	2887

6.1175.1 Constructor & Destructor Documentation	2888
6.1175.1.1 ProxyLineEdit()	2888
6.1175.2 Member Function Documentation	2888
6.1175.2.1 mouseMoveEvent()	2888
6.1175.2.2 mousePressEvent()	2888
6.1175.2.3 setWidget()	2889
6.1176 Digikam::QListImageListProvider Class Reference	2889
6.1176.1 Detailed Description	2890
6.1176.2 Member Function Documentation	2890
6.1176.2.1 atEnd()	2890
6.1176.2.2 image()	2890
6.1176.2.3 images()	2890
6.1176.2.4 proceed()	2890
6.1176.2.5 setImages()	2891
6.1176.2.6 setUnpairedImages()	2891
6.1176.2.7 size()	2891
6.1177 Digikam::QMapForAdaptors< Key, Value > Class Template Reference	2891
6.1177.1 Detailed Description	2892
6.1178 Digikam::QueueListView Class Reference	2892
6.1178.1 Member Enumeration Documentation	2893
6.1178.1.1 ItemListType	2893
6.1179 Digikam::QueueListViewItem Class Reference	2894
6.1180 Digikam::QueueMgrWindow Class Reference	2895
6.1180.1 Member Function Documentation	2897
6.1180.1.1 infolface()	2897
6.1180.1.2 queuesMap()	2898
6.1181 Digikam::QueueMgrWindow::Private Class Reference	2898
6.1182 Digikam::QueuePool Class Reference	2899
6.1182.1 Member Function Documentation	2900
6.1182.1.1 applySettings()	2900
6.1183 Digikam::QueuePoolBar Class Reference	2901
6.1184 Digikam::QueueSettings Class Reference	2901
6.1184.1 Detailed Description	2902
6.1185 Digikam::QueueSettingsView Class Reference	2902
6.1186 Digikam::QueueToolTip Class Reference	2903
6.1187 Digikam::RadioButtonHBox Class Reference	2904
6.1188 Digikam::RainDropFilter Class Reference	2905
6.1188.1 Member Function Documentation	2908
6.1188.1.1 filterAction()	2908
6.1188.1.2 filterIdentifier()	2908
6.1188.1.3 readParameters()	2908
6.1189 Digikam::RandomNumberGenerator Class Reference	2908

---

6.1189.1 Detailed Description . . . . .	2909
6.1189.2 Constructor & Destructor Documentation . . . . .	2909
6.1189.2.1 RandomNumberGenerator() . . . . .	2909
6.1189.3 Member Function Documentation . . . . .	2909
6.1189.3.1 currentSeed() . . . . .	2909
6.1189.3.2 nonDeterministicSeed() . . . . .	2909
6.1189.3.3 number() [1/2] . . . . .	2909
6.1189.3.4 number() [2/2] . . . . .	2910
6.1189.3.5 reseed() . . . . .	2910
6.1189.3.6 seed() . . . . .	2910
6.1189.3.7 seedByTime() . . . . .	2910
6.1189.3.8 seedNonDeterministic() . . . . .	2910
6.1189.3.9 timeSeed() . . . . .	2910
6.1189.3.10 yesOrNo() . . . . .	2911
6.1190 Digikam::RangeDialog Class Reference . . . . .	2911
6.1191 Digikam::RangeModifier Class Reference . . . . .	2912
6.1191.1 Member Function Documentation . . . . .	2914
6.1191.1.1 parseOperation() . . . . .	2914
6.1192 Digikam::RatingBox Class Reference . . . . .	2915
6.1193 Digikam::RatingComboBox Class Reference . . . . .	2917
6.1193.1 Member Enumeration Documentation . . . . .	2918
6.1193.1.1 RatingValue . . . . .	2918
6.1194 Digikam::RatingComboBoxDelegate Class Reference . . . . .	2919
6.1195 Digikam::RatingComboBoxModel Class Reference . . . . .	2920
6.1196 Digikam::RatingComboBoxWidget Class Reference . . . . .	2921
6.1196.1 Constructor & Destructor Documentation . . . . .	2923
6.1196.1.1 RatingComboBoxWidget() . . . . .	2923
6.1197 Digikam::RatingFilter Class Reference . . . . .	2924
6.1198 Digikam::RatingFilterWidget Class Reference . . . . .	2926
6.1199 Digikam::RatingMenuAction Class Reference . . . . .	2928
6.1200 Digikam::RatingStarDrawer Class Reference . . . . .	2929
6.1201 Digikam::RatingWidget Class Reference . . . . .	2930
6.1201.1 Member Function Documentation . . . . .	2931
6.1201.1.1 starPolygon() . . . . .	2931
6.1202 Digikam::RawCameraDlg Class Reference . . . . .	2932
6.1203 Digikam::RawPage Class Reference . . . . .	2933
6.1204 Digikam::RawProcessingFilter Class Reference . . . . .	2935
6.1204.1 Detailed Description . . . . .	2938
6.1204.2 Constructor & Destructor Documentation . . . . .	2938
6.1204.2.1 RawProcessingFilter() [1/3] . . . . .	2938
6.1204.2.2 RawProcessingFilter() [2/3] . . . . .	2938
6.1204.2.3 RawProcessingFilter() [3/3] . . . . .	2939

---

6.1204.3 Member Function Documentation	2939
6.1204.3.1 filterAction()	2939
6.1204.3.2 filterIdentifier()	2939
6.1204.3.3 filterImage()	2939
6.1204.3.4 readParameters()	2939
6.1204.3.5 setObserver()	2940
6.1204.3.6 setOutputProfile()	2940
6.1204.3.7 setSettings()	2940
6.1205 Digikam::RecognitionBenchmarker Class Reference	2941
6.1205.1 Member Function Documentation	2943
6.1205.1.1 result()	2943
6.1206 Digikam::RecognitionBenchmarker::Statistics Class Reference	2943
6.1207 Digikam::RecognitionPreprocessor Class Reference	2944
6.1207.1 Member Function Documentation	2944
6.1207.1.1 preprocess()	2944
6.1208 Digikam::RecognitionTrainingProvider Class Reference	2945
6.1208.1 Detailed Description	2945
6.1208.2 Member Function Documentation	2946
6.1208.2.1 images()	2946
6.1208.2.2 newImages()	2946
6.1209 Digikam::RecognitionTrainingUpdateQueue Class Reference	2946
6.1210 Digikam::RecognitionWorker Class Reference	2947
6.1210.1 Member Function Documentation	2949
6.1210.1.1 aboutToDeactivate()	2949
6.1210.1.2 process	2949
6.1211 Digikam::RedEye::RegressionTree Struct Reference	2949
6.1211.1 Member Function Documentation	2950
6.1211.1.1 operator>()	2950
6.1212 Digikam::RedEye::ShapePredictor Class Reference	2950
6.1213 Digikam::RedEye::SplitFeature Struct Reference	2951
6.1214 Digikam::RedEyeCorrectionContainer Class Reference	2951
6.1215 Digikam::RedEyeCorrectionFilter Class Reference	2952
6.1215.1 Constructor & Destructor Documentation	2955
6.1215.1.1 RedEyeCorrectionFilter()	2955
6.1215.2 Member Function Documentation	2955
6.1215.2.1 filterAction()	2955
6.1215.2.2 filterIdentifier()	2955
6.1216 Digikam::RedEyeCorrectionSettings Class Reference	2956
6.1217 Digikam::RefocusFilter Class Reference	2957
6.1217.1 Member Function Documentation	2960
6.1217.1.1 filterAction()	2960
6.1217.1.2 filterIdentifier()	2960



6.1217.1.3 readParameters() . . . . .	2960
6.1218 Digikam::RefocusMatrix Class Reference . . . . .	2960
6.1219 Digikam::RegionFrameItem Class Reference . . . . .	2961
6.1219.1 Member Function Documentation . . . . .	2963
6.1219.1.1 setHudWidget() . . . . .	2963
6.1219.1.2 setViewportRect . . . . .	2964
6.1220 Digikam::RemoveBookmarksCommand Class Reference . . . . .	2964
6.1221 Digikam::RemoveDoublesModifier Class Reference . . . . .	2966
6.1221.1 Member Function Documentation . . . . .	2967
6.1221.1.1 parseOperation() . . . . .	2967
6.1222 Digikam::RemoveFilterAction Class Reference . . . . .	2968
6.1223 Digikam::RenameCustomizer Class Reference . . . . .	2969
6.1224 Digikam::RenameFileJob Class Reference . . . . .	2970
6.1225 Digikam::ReplaceDialog Class Reference . . . . .	2972
6.1226 Digikam::ReplaceModifier Class Reference . . . . .	2973
6.1226.1 Member Function Documentation . . . . .	2974
6.1226.1.1 parseOperation() . . . . .	2974
6.1227 Digikam::RestoreDTrashItemsJob Class Reference . . . . .	2975
6.1228 Digikam::RGBBackend Class Reference . . . . .	2977
6.1228.1 Constructor & Destructor Documentation . . . . .	2977
6.1228.1.1 RGBBackend() . . . . .	2977
6.1228.2 Member Function Documentation . . . . .	2978
6.1228.2.1 backendName() . . . . .	2978
6.1228.2.2 callRGBBackend() . . . . .	2978
6.1228.2.3 getErrorMessage() . . . . .	2978
6.1229 Digikam::RGInfo Class Reference . . . . .	2978
6.1229.1 Constructor & Destructor Documentation . . . . .	2978
6.1229.1.1 RGInfo() . . . . .	2978
6.1229.1.2 ~RGInfo() . . . . .	2979
6.1229.2 Member Data Documentation . . . . .	2979
6.1229.2.1 coordinates . . . . .	2979
6.1229.2.2 id . . . . .	2979
6.1229.2.3 rgData . . . . .	2979
6.1230 Digikam::RGTagModel Class Reference . . . . .	2979
6.1230.1 Detailed Description . . . . .	2982
6.1230.2 Constructor & Destructor Documentation . . . . .	2982
6.1230.2.1 RGTagModel() . . . . .	2982
6.1230.3 Member Function Documentation . . . . .	2983
6.1230.3.1 addDataInTree() . . . . .	2983
6.1230.3.2 addExternalTags() . . . . .	2983
6.1230.3.3 addNewData() . . . . .	2983
6.1230.3.4 addNewTag() . . . . .	2983

6.1230.3.5 addSpacerTag()	2984
6.1230.3.6 branchFromIndex()	2984
6.1230.3.7 climbTreeAndGetSpacers()	2984
6.1230.3.8 deleteAllSpacersOrNewTags()	2985
6.1230.3.9 deleteTag()	2985
6.1230.3.10 findAndDeleteSpacersOrNewTags()	2985
6.1230.3.11 fromSourceIndex()	2985
6.1230.3.12 getSpacerAddress()	2986
6.1230.3.13 getSpacers()	2987
6.1230.3.14 getTagType()	2987
6.1230.3.15 readdNewTags()	2987
6.1230.3.16 readdTag()	2987
6.1230.3.17 toSourceIndex()	2988
6.1231 Digikam::RGWidget Class Reference	2988
6.1231.1 Constructor & Destructor Documentation	2989
6.1231.1.1 RGWidget()	2989
6.1231.1.2 ~RGWidget()	2989
6.1231.2 Member Function Documentation	2990
6.1231.2.1 eventFilter()	2990
6.1231.2.2 readSettingsFromGroup()	2990
6.1231.2.3 saveSettingsToGroup()	2990
6.1231.2.4 setUIEnabled()	2990
6.1231.2.5 signalProgressChanged	2990
6.1231.2.6 signalProgressSetup	2991
6.1231.2.7 signalSetUIEnabled	2991
6.1231.2.8 signalUndoCommand	2991
6.1232 Digikam::RubberItem Class Reference	2992
6.1233 Digikam::Rule Class Reference	2995
6.1233.1 Member Function Documentation	2996
6.1233.1.1 addToken()	2996
6.1233.1.2 escapeToken()	2996
6.1233.1.3 isValid()	2996
6.1233.1.4 parseOperation()	2997
6.1233.1.5 regExp()	2997
6.1233.1.6 registerButton()	2998
6.1233.1.7 registerMenu()	2999
6.1233.1.8 reset()	2999
6.1233.1.9 setUseTokenMenu()	2999
6.1233.1.10 tokens()	3000
6.1233.1.11 useTokenMenu()	3000
6.1234 Digikam::RuleDialog Class Reference	3000
6.1235 Digikam::RuleType Class Reference	3000

6.1236 Digikam::RuleTypeForConversion Class Reference . . . . .	3000
6.1237 Digikam::SafeTemporaryFile Class Reference . . . . .	3001
6.1238 Digikam::SAAlbum Class Reference . . . . .	3002
6.1238.1 Detailed Description . . . . .	3004
6.1238.2 Member Function Documentation . . . . .	3004
6.1238.2.1 databaseUrl() . . . . .	3004
6.1238.2.2 getTemporaryHaarTitle() . . . . .	3004
6.1238.2.3 getTemporaryTitle() . . . . .	3005
6.1238.2.4 isTemporarySearch() . . . . .	3005
6.1239 Digikam::SaveProperties Class Reference . . . . .	3005
6.1240 Digikam::SavingContext Class Reference . . . . .	3005
6.1241 Digikam::SavingTask Class Reference . . . . .	3006
6.1241.1 Member Function Documentation . . . . .	3007
6.1241.1.1 continueQuery() . . . . .	3007
6.1241.1.2 execute() . . . . .	3007
6.1241.1.3 progressInfo() . . . . .	3008
6.1241.1.4 type() . . . . .	3008
6.1242 Digikam::ScanController Class Reference . . . . .	3009
6.1242.1 Member Function Documentation . . . . .	3011
6.1242.1.1 abortInitialization() . . . . .	3011
6.1242.1.2 beginFileMetadataWrite() . . . . .	3011
6.1242.1.3 cancelAllAndSuspendCollectionScan() . . . . .	3011
6.1242.1.4 cancelCompleteScan() . . . . .	3011
6.1242.1.5 completeCollectionScan() . . . . .	3012
6.1242.1.6 completeCollectionScanInBackground() . . . . .	3012
6.1242.1.7 databaseInitialization() . . . . .	3012
6.1242.1.8 finishFileMetadataWrite() . . . . .	3012
6.1242.1.9 getNewIdsList() . . . . .	3012
6.1242.1.10 hintAtModificationOfItems() . . . . .	3012
6.1242.1.11 hintAtMoveOrCopyOfAlbum() . . . . .	3013
6.1242.1.12 hintAtMoveOrCopyOfItems() . . . . .	3013
6.1242.1.13 restart() . . . . .	3013
6.1242.1.14 restartCollectionScan() . . . . .	3013
6.1242.1.15 resumeCollectionScan() . . . . .	3013
6.1242.1.16 scannedInfo() . . . . .	3013
6.1242.1.17 scheduleCollectionScan() . . . . .	3014
6.1242.1.18 scheduleCollectionScanExternal() . . . . .	3014
6.1242.1.19 scheduleCollectionScanRelaxed() . . . . .	3014
6.1242.1.20 shutDown() . . . . .	3014
6.1242.1.21 suspendCollectionScan() . . . . .	3014
6.1242.1.22 updateUniqueHash() . . . . .	3014
6.1243 Digikam::ScanController::FileMetadataWrite Class Reference . . . . .	3015

---

6.1243.1 Detailed Description . . . . .	3015
6.1244 Digikam::ScanController::Private Class Reference . . . . .	3015
6.1245 Digikam::ScanControllerCreator Class Reference . . . . .	3016
6.1246 Digikam::ScanControllerLoadingCacheFileWatch Class Reference . . . . .	3017
6.1247 Digikam::ScanStateFilter Class Reference . . . . .	3018
6.1247.1 Member Function Documentation . . . . .	3020
6.1247.1.1 run() . . . . .	3020
6.1248 Digikam::SchemeManager Class Reference . . . . .	3020
6.1248.1 Detailed Description . . . . .	3021
6.1248.2 Member Enumeration Documentation . . . . .	3022
6.1248.2.1 BackgroundRole . . . . .	3022
6.1248.2.2 ColorSet . . . . .	3022
6.1248.2.3 DecorationRole . . . . .	3023
6.1248.2.4 ForegroundRole . . . . .	3023
6.1248.2.5 ShadeRole . . . . .	3024
6.1248.3 Constructor & Destructor Documentation . . . . .	3024
6.1248.3.1 SchemeManager() [1/2] . . . . .	3024
6.1248.3.2 ~SchemeManager() . . . . .	3024
6.1248.3.3 SchemeManager() [2/2] . . . . .	3025
6.1248.4 Member Function Documentation . . . . .	3025
6.1248.4.1 adjustBackground() . . . . .	3025
6.1248.4.2 adjustForeground() . . . . .	3025
6.1248.4.3 background() . . . . .	3026
6.1248.4.4 contrast() . . . . .	3026
6.1248.4.5 contrastF() . . . . .	3026
6.1248.4.6 createApplicationPalette() . . . . .	3026
6.1248.4.7 decoration() . . . . .	3027
6.1248.4.8 foreground() . . . . .	3027
6.1248.4.9 operator=() . . . . .	3027
6.1248.4.10 shade() [1/3] . . . . .	3027
6.1248.4.11 shade() [2/3] . . . . .	3027
6.1248.4.12 shade() [3/3] . . . . .	3028
6.1249 Digikam::ScriptingSettings Class Reference . . . . .	3028
6.1250 Digikam::SearchChangeset Class Reference . . . . .	3029
6.1251 Digikam::SearchesDBJobInfo Class Reference . . . . .	3030
6.1252 Digikam::SearchesDBJobsThread Class Reference . . . . .	3032
6.1252.1 Member Function Documentation . . . . .	3034
6.1252.1.1 searchesListing() . . . . .	3034
6.1253 Digikam::SearchesJob Class Reference . . . . .	3035
6.1254 Digikam::SearchField Class Reference . . . . .	3036
6.1254.1 Member Function Documentation . . . . .	3038
6.1254.1.1 createField() . . . . .	3038

---

6.1254.1.2 isVisible()	3038
6.1254.1.3 setVisible()	3038
6.1254.1.4 write()	3038
6.1255 Digikam::SearchFieldAlbum Class Reference	3039
6.1255.1 Member Function Documentation	3041
6.1255.1.1 read()	3041
6.1255.1.2 reset()	3041
6.1255.1.3 setupValueWidgets()	3041
6.1255.1.4 setValueWidgetsVisible()	3041
6.1255.1.5 valueWidgetRects()	3042
6.1255.1.6 write()	3042
6.1256 Digikam::SearchFieldCheckBox Class Reference	3043
6.1256.1 Member Function Documentation	3045
6.1256.1.1 read()	3045
6.1256.1.2 reset()	3045
6.1256.1.3 setupValueWidgets()	3045
6.1256.1.4 setValueWidgetsVisible()	3045
6.1256.1.5 valueWidgetRects()	3046
6.1256.1.6 write()	3046
6.1257 Digikam::SearchFieldChoice Class Reference	3047
6.1257.1 Member Function Documentation	3049
6.1257.1.1 read()	3049
6.1257.1.2 reset()	3049
6.1257.1.3 setupValueWidgets()	3049
6.1257.1.4 setValueWidgetsVisible()	3050
6.1257.1.5 valueWidgetRects()	3050
6.1257.1.6 write()	3050
6.1258 Digikam::SearchFieldColorDepth Class Reference	3051
6.1258.1 Member Function Documentation	3053
6.1258.1.1 read()	3053
6.1258.1.2 setupValueWidgets()	3053
6.1259 Digikam::SearchFieldComboBox Class Reference	3054
6.1259.1 Member Function Documentation	3056
6.1259.1.1 reset()	3056
6.1259.1.2 setupValueWidgets()	3056
6.1259.1.3 setValueWidgetsVisible()	3056
6.1259.1.4 valueWidgetRects()	3056
6.1259.1.5 write()	3057
6.1260 Digikam::SearchFieldGroup Class Reference	3057
6.1261 Digikam::SearchFieldGroupLabel Class Reference	3058
6.1262 Digikam::SearchFieldKeyword Class Reference	3060
6.1262.1 Member Function Documentation	3062

---

6.1262.1.1 read()	3062
6.1262.1.2 write()	3062
6.1263 Digikam::SearchFieldLabels Class Reference	3063
6.1263.1 Member Function Documentation	3065
6.1263.1.1 read()	3065
6.1263.1.2 reset()	3065
6.1263.1.3 setValueWidgets()	3065
6.1263.1.4 setValueWidgetsVisible()	3065
6.1263.1.5 valueWidgetRects()	3066
6.1263.1.6 write()	3066
6.1264 Digikam::SearchFieldMonthDay Class Reference	3067
6.1264.1 Member Function Documentation	3069
6.1264.1.1 read()	3069
6.1264.1.2 reset()	3069
6.1264.1.3 setValueWidgets()	3069
6.1264.1.4 setValueWidgetsVisible()	3069
6.1264.1.5 valueWidgetRects()	3070
6.1264.1.6 write()	3070
6.1265 Digikam::SearchFieldPageOrientation Class Reference	3071
6.1265.1 Member Function Documentation	3073
6.1265.1.1 read()	3073
6.1265.1.2 setValueWidgets()	3073
6.1266 Digikam::SearchFieldRangeDate Class Reference	3074
6.1266.1 Member Function Documentation	3076
6.1266.1.1 read()	3076
6.1266.1.2 reset()	3076
6.1266.1.3 setValueWidgets()	3076
6.1266.1.4 setValueWidgetsVisible()	3077
6.1266.1.5 valueWidgetRects()	3077
6.1266.1.6 write()	3077
6.1267 Digikam::SearchFieldRangeDouble Class Reference	3078
6.1267.1 Member Function Documentation	3080
6.1267.1.1 read()	3080
6.1267.1.2 reset()	3080
6.1267.1.3 setValueWidgets()	3080
6.1267.1.4 setValueWidgetsVisible()	3081
6.1267.1.5 valueWidgetRects()	3081
6.1267.1.6 write()	3081
6.1268 Digikam::SearchFieldRangeInt Class Reference	3082
6.1268.1 Member Function Documentation	3084
6.1268.1.1 read()	3084
6.1268.1.2 reset()	3084

---

6.1268.1.3	setupValueWidgets()	3084
6.1268.1.4	setValueWidgetsVisible()	3085
6.1268.1.5	valueWidgetRects()	3085
6.1268.1.6	write()	3085
6.1269	Digikam::SearchFieldRangeTime Class Reference	3086
6.1269.1	Member Function Documentation	3088
6.1269.1.1	read()	3088
6.1269.1.2	reset()	3088
6.1269.1.3	setupValueWidgets()	3088
6.1269.1.4	setValueWidgetsVisible()	3088
6.1269.1.5	valueWidgetRects()	3089
6.1269.1.6	write()	3089
6.1270	Digikam::SearchFieldRating Class Reference	3090
6.1270.1	Member Function Documentation	3092
6.1270.1.1	read()	3092
6.1270.1.2	reset()	3092
6.1270.1.3	setupValueWidgets()	3092
6.1270.1.4	setValueWidgetsVisible()	3092
6.1270.1.5	valueWidgetRects()	3093
6.1270.1.6	write()	3093
6.1271	Digikam::SearchFieldText Class Reference	3094
6.1271.1	Member Function Documentation	3096
6.1271.1.1	read()	3096
6.1271.1.2	reset()	3096
6.1271.1.3	setupValueWidgets()	3096
6.1271.1.4	setValueWidgetsVisible()	3096
6.1271.1.5	valueWidgetRects()	3097
6.1271.1.6	write()	3097
6.1272	Digikam::SearchFilterModel Class Reference	3098
6.1272.1	Detailed Description	3101
6.1272.2	Member Function Documentation	3101
6.1272.2.1	isFiltering()	3101
6.1272.2.2	matches()	3101
6.1272.2.3	setFilterSearchType()	3101
6.1272.2.4	setListTemporarySearches()	3101
6.1273	Digikam::SearchGroup Class Reference	3102
6.1273.1	Member Function Documentation	3104
6.1273.1.1	addGroupToLayout()	3104
6.1273.1.2	createSearchGroup()	3104
6.1274	Digikam::SearchGroupLabel Class Reference	3105
6.1275	Digikam::SearchInfo Class Reference	3106
6.1275.1	Detailed Description	3106

---

6.1275.2 Member Function Documentation . . . . .	3106
6.1275.2.1 operator<() . . . . .	3106
6.1276 Digikam::SearchModel Class Reference . . . . .	3107
6.1276.1 Constructor & Destructor Documentation . . . . .	3112
6.1276.1.1 SearchModel() . . . . .	3112
6.1276.2 Member Function Documentation . . . . .	3112
6.1276.2.1 albumData() . . . . .	3112
6.1276.2.2 albumForId() . . . . .	3113
6.1276.2.3 setPixmapForNormalSearches() . . . . .	3113
6.1276.2.4 setReplaceNames() . . . . .	3113
6.1277 Digikam::SearchModificationHelper Class Reference . . . . .	3114
6.1277.1 Detailed Description . . . . .	3115
6.1277.2 Constructor & Destructor Documentation . . . . .	3115
6.1277.2.1 SearchModificationHelper() . . . . .	3115
6.1277.2.2 ~SearchModificationHelper() . . . . .	3115
6.1277.3 Member Function Documentation . . . . .	3115
6.1277.3.1 createFuzzySearchFromDropped() . . . . .	3115
6.1277.3.2 createFuzzySearchFromImage() . . . . .	3116
6.1277.3.3 createFuzzySearchFromSketch() . . . . .	3116
6.1277.3.4 slotCreateFuzzySearchFromDropped . . . . .	3116
6.1277.3.5 slotCreateFuzzySearchFromImage . . . . .	3117
6.1277.3.6 slotCreateFuzzySearchFromSketch . . . . .	3117
6.1277.3.7 slotCreateTimeLineSearch . . . . .	3118
6.1277.3.8 slotSearchDelete . . . . .	3118
6.1277.3.9 slotSearchRename . . . . .	3118
6.1278 Digikam::SearchSideBarWidget Class Reference . . . . .	3119
6.1278.1 Member Function Documentation . . . . .	3120
6.1278.1.1 applySettings() . . . . .	3120
6.1278.1.2 changeAlbumFromHistory() . . . . .	3121
6.1278.1.3 doLoadState() . . . . .	3121
6.1278.1.4 doSaveState() . . . . .	3121
6.1278.1.5 getCaption() . . . . .	3121
6.1278.1.6 getIcon() . . . . .	3121
6.1278.1.7 setActive() . . . . .	3121
6.1279 Digikam::SearchTabHeader Class Reference . . . . .	3122
6.1280 Digikam::SearchTextBar Class Reference . . . . .	3123
6.1280.1 Detailed Description . . . . .	3125
6.1280.2 Member Enumeration Documentation . . . . .	3125
6.1280.2.1 HighlightState . . . . .	3125
6.1280.3 Member Function Documentation . . . . .	3125
6.1280.3.1 doLoadState() . . . . .	3125
6.1280.3.2 doSaveState() . . . . .	3125



---

6.1280.3.3	getCurrentHighlightState()	3125
6.1280.3.4	setCaseSensitive()	3125
6.1280.3.5	setHighlightOnResult()	3126
6.1281	Digikam::SearchTextBarDb Class Reference	3127
6.1281.1	Detailed Description	3129
6.1281.2	Member Function Documentation	3129
6.1281.2.1	setFilterModel()	3129
6.1281.2.2	setModel() [1/2]	3129
6.1281.2.3	setModel() [2/2]	3129
6.1282	Digikam::SearchTextFilterSettings Class Reference	3130
6.1283	Digikam::SearchTextSettings Class Reference	3131
6.1284	Digikam::SearchTreeView Class Reference	3132
6.1285	Digikam::SearchView Class Reference	3137
6.1285.1	Member Function Documentation	3139
6.1285.1.1	addGroupToLayout()	3139
6.1285.1.2	bottomBarPixmap()	3139
6.1285.1.3	createSearchGroup()	3139
6.1285.1.4	groupLabelPixmap()	3139
6.1286	Digikam::SearchViewBottomBar Class Reference	3140
6.1287	Digikam::SearchViewThemedPartsCache Class Reference	3141
6.1288	Digikam::SearchWindow Class Reference	3142
6.1288.1	Constructor & Destructor Documentation	3143
6.1288.1.1	SearchWindow()	3143
6.1288.2	Member Function Documentation	3143
6.1288.2.1	readSearch()	3143
6.1288.2.2	reset()	3143
6.1288.2.3	search()	3143
6.1288.2.4	searchEdited	3143
6.1289	Digikam::SearchXmlCachingReader Class Reference	3144
6.1289.1	Constructor & Destructor Documentation	3146
6.1289.1.1	SearchXmlCachingReader()	3146
6.1290	Digikam::SearchXmlReader Class Reference	3147
6.1290.1	Member Function Documentation	3148
6.1290.1.1	defaultFieldOperator()	3148
6.1290.1.2	fieldOperator()	3148
6.1290.1.3	groupCaption()	3149
6.1290.1.4	groupOperator()	3149
6.1290.1.5	isFieldElement()	3149
6.1290.1.6	isGroupElement()	3149
6.1290.1.7	readNext()	3149
6.1290.1.8	readToEndOfElement()	3149
6.1290.1.9	readToFirstField()	3149

---

6.1290.1.10 readToStartOfElement()	3150
6.1290.1.11 value()	3150
6.1291 Digikam::SearchXmlWriter Class Reference	3151
6.1291.1 Constructor & Destructor Documentation	3152
6.1291.1.1 SearchXmlWriter()	3152
6.1291.2 Member Function Documentation	3152
6.1291.2.1 finish()	3152
6.1291.2.2 finishField()	3153
6.1291.2.3 finishGroup()	3153
6.1291.2.4 keywordSearch()	3153
6.1291.2.5 setDefaultFieldOperator()	3153
6.1291.2.6 setFieldOperator()	3153
6.1291.2.7 setGroupCaption()	3153
6.1291.2.8 setGroupOperator()	3153
6.1291.2.9 writeField()	3154
6.1291.2.10 writeGroup()	3154
6.1291.2.11 writeValue()	3154
6.1291.2.12 xml()	3154
6.1292 Digikam::SequenceNumberDialog Class Reference	3155
6.1293 Digikam::SequenceNumberOption Class Reference	3156
6.1293.1 Member Function Documentation	3157
6.1293.1.1 parseOperation()	3157
6.1294 Digikam::Setup Class Reference	3159
6.1294.1 Member Function Documentation	3161
6.1294.1.1 execDialog()	3161
6.1294.1.2 execSinglePage()	3161
6.1295 Digikam::SetupAlbumView Class Reference	3162
6.1296 Digikam::SetupCamera Class Reference	3163
6.1297 Digikam::SetupCategory Class Reference	3164
6.1298 Digikam::SetupCollectionDelegate Class Reference	3165
6.1298.1 Member Function Documentation	3167
6.1298.1.1 createItemWidgets()	3167
6.1298.1.2 updateItemWidgets()	3167
6.1299 Digikam::SetupCollectionModel Class Reference	3169
6.1299.1 Member Enumeration Documentation	3171
6.1299.1.1 SetupCollectionDataRole	3171
6.1299.2 Constructor & Destructor Documentation	3172
6.1299.2.1 SetupCollectionModel()	3172
6.1299.3 Member Function Documentation	3172
6.1299.3.1 slotAppendPressed	3172
6.1299.3.2 slotCategoryButtonPressed	3172
6.1300 Digikam::SetupCollectionModel::Item Class Reference	3172

---

6.1301 Digikam::SetupCollections Class Reference	3173
6.1302 Digikam::SetupCollectionTreeView Class Reference	3174
6.1303 Digikam::SetupDatabase Class Reference	3175
6.1304 Digikam::SetupEditor Class Reference	3176
6.1305 Digikam::SetupEditorIface Class Reference	3177
6.1306 Digikam::SetupGeolocation Class Reference	3178
6.1307 Digikam::SetupICC Class Reference	3179
6.1307.1 Constructor & Destructor Documentation	3179
6.1307.1.1 SetupICC()	3179
6.1308 Digikam::SetupImageQualitySorter Class Reference	3180
6.1309 Digikam::SetupIOFiles Class Reference	3181
6.1310 Digikam::SetupLightTable Class Reference	3181
6.1311 Digikam::SetupMetadata Class Reference	3182
6.1312 Digikam::SetupMetadata::Private Class Reference	3183
6.1313 Digikam::SetupMime Class Reference	3184
6.1314 Digikam::SetupMisc Class Reference	3185
6.1315 Digikam::SetupMisc::Private Class Reference	3186
6.1316 Digikam::SetupPlugins Class Reference	3187
6.1317 Digikam::SetupRaw Class Reference	3188
6.1318 Digikam::SetupTemplate Class Reference	3189
6.1319 Digikam::SetupToolTip Class Reference	3190
6.1320 Digikam::SetupVersioning Class Reference	3191
6.1321 Digikam::SharedLoadingTask Class Reference	3192
6.1321.1 Member Function Documentation	3194
6.1321.1.1 accessMode()	3194
6.1321.1.2 addListener()	3194
6.1321.1.3 cacheKey()	3194
6.1321.1.4 completed()	3194
6.1321.1.5 execute()	3194
6.1321.1.6 loadSaveNotifier()	3195
6.1321.1.7 notifyNewLoadingProcess()	3195
6.1321.1.8 progressInfo()	3195
6.1321.1.9 querySendNotifyEvent()	3195
6.1321.1.10 removeListener()	3195
6.1321.1.11 setResult()	3195
6.1322 Digikam::SharedLoadSaveThread Class Reference	3196
6.1323 Digikam::SharedQueue< T > Class Template Reference	3199
6.1324 Digikam::SharpContainer Class Reference	3199
6.1325 Digikam::SharpenFilter Class Reference	3201
6.1325.1 Constructor & Destructor Documentation	3204
6.1325.1.1 SharpenFilter()	3204
6.1325.2 Member Function Documentation	3204

---

6.1325.2.1 filterAction()	3204
6.1325.2.2 filterIdentifier()	3204
6.1325.2.3 readParameters()	3204
6.1326 Digikam::SharpSettings Class Reference	3205
6.1327 Digikam::ShearFilter Class Reference	3206
6.1327.1 Member Function Documentation	3209
6.1327.1.1 filterAction()	3209
6.1327.1.2 filterIdentifier()	3209
6.1327.1.3 readParameters()	3209
6.1328 Digikam::ShowHideVersionsOverlay Class Reference	3210
6.1328.1 Member Function Documentation	3213
6.1328.1.1 checkIndex()	3213
6.1328.1.2 createButton()	3213
6.1328.1.3 setActive()	3213
6.1328.1.4 updateButton()	3213
6.1329 Digikam::Sidebar Class Reference	3214
6.1329.1 Detailed Description	3216
6.1329.2 Constructor & Destructor Documentation	3216
6.1329.2.1 Sidebar()	3216
6.1329.3 Member Function Documentation	3217
6.1329.3.1 activeNextTab()	3217
6.1329.3.2 activePreviousTab()	3217
6.1329.3.3 appendTab()	3217
6.1329.3.4 backup() [1/2]	3217
6.1329.3.5 backup() [2/2]	3218
6.1329.3.6 deleteTab()	3218
6.1329.3.7 doLoadState()	3218
6.1329.3.8 doSaveState()	3218
6.1329.3.9 expand()	3218
6.1329.3.10 getActiveTab()	3218
6.1329.3.11 isExpanded()	3218
6.1329.3.12 restore() [1/2]	3219
6.1329.3.13 restore() [2/2]	3219
6.1329.3.14 setActiveTab()	3219
6.1329.3.15 shrink()	3219
6.1329.3.16 signalChangedTab	3219
6.1329.3.17 signalViewChanged	3219
6.1330 Digikam::Sidebar::Private Class Reference	3220
6.1330.1 Member Data Documentation	3220
6.1330.1.1 isMinimized	3220
6.1331 Digikam::SidebarSplitter Class Reference	3221
6.1331.1 Constructor & Destructor Documentation	3222

6.1331.1.1 SidebarSplitter()	3222
6.1331.2 Member Function Documentation	3222
6.1331.2.1 restoreState() [1/2]	3222
6.1331.2.2 restoreState() [2/2]	3222
6.1331.2.3 saveState() [1/2]	3223
6.1331.2.4 saveState() [2/2]	3223
6.1331.2.5 setSize()	3223
6.1331.2.6 size()	3223
6.1332 Digikam::SidebarSplitter::Private Class Reference	3223
6.1333 Digikam::SidebarState Class Reference	3223
6.1334 Digikam::SidebarWidget Class Reference	3224
6.1334.1 Detailed Description	3225
6.1334.2 Constructor & Destructor Documentation	3225
6.1334.2.1 SidebarWidget()	3225
6.1334.2.2 ~SidebarWidget()	3225
6.1334.3 Member Function Documentation	3225
6.1334.3.1 applySettings()	3225
6.1334.3.2 changeAlbumFromHistory()	3226
6.1334.3.3 getCaption()	3226
6.1334.3.4 getIcon()	3226
6.1334.3.5 requestActiveTab	3226
6.1334.3.6 setActive()	3226
6.1334.3.7 signalNotificationError	3227
6.1335 Digikam::SidecarFinder Class Reference	3227
6.1336 Digikam::SimilarityDb Class Reference	3227
6.1336.1 Member Function Documentation	3228
6.1336.1.1 clearImageSimilarity()	3228
6.1336.1.2 copySimilarityAttributes()	3228
6.1336.1.3 getDirtyOrMissingFingerprints()	3228
6.1336.1.4 getDirtyOrMissingFingerprintURLs()	3229
6.1336.1.5 getImageSimilarity()	3229
6.1336.1.6 getImageSimilarityAlgorithms()	3229
6.1336.1.7 getLegacySetting()	3230
6.1336.1.8 getSetting()	3230
6.1336.1.9 hasDirtyOrMissingFingerprint()	3230
6.1336.1.10 hasFingerprint()	3231
6.1336.1.11 hasFingerprints() [1/2]	3231
6.1336.1.12 hasFingerprints() [2/2]	3231
6.1336.1.13 integrityCheck()	3231
6.1336.1.14 registeredImageIds()	3232
6.1336.1.15 removeImageFingerprint()	3232
6.1336.1.16 removeImageSimilarity() [1/2]	3232

---

6.1336.1.17 removeImageSimilarity() [2/2]	3232
6.1336.1.18 setSetting()	3233
6.1336.1.19 vacuum()	3233
6.1337 Digikam::SimilarityDbAccess Class Reference	3233
6.1337.1 Constructor & Destructor Documentation	3234
6.1337.1.1 SimilarityDbAccess()	3234
6.1337.2 Member Function Documentation	3234
6.1337.2.1 checkReadyForUse()	3234
6.1337.2.2 cleanUpDatabase()	3234
6.1337.2.3 initDbEngineErrorHandler()	3234
6.1337.2.4 isInitialized()	3234
6.1337.2.5 parameters()	3235
6.1337.2.6 setLastError()	3235
6.1337.2.7 setParameters()	3235
6.1338 Digikam::SimilarityDbBackend Class Reference	3236
6.1338.1 Member Function Documentation	3239
6.1338.1.1 initSchema()	3239
6.1339 Digikam::SimilarityDbSchemaUpdater Class Reference	3239
6.1340 Digikam::SimpleCollectionScannerObserver Class Reference	3239
6.1340.1 Member Function Documentation	3240
6.1340.1.1 continueQuery()	3240
6.1341 Digikam::SimpleTreeModel Class Reference	3240
6.1342 Digikam::SimpleTreeModel::Item Class Reference	3241
6.1343 Digikam::SinglePhotoPreviewLayout Class Reference	3242
6.1343.1 Member Function Documentation	3243
6.1343.1.1 addItem()	3243
6.1343.1.2 maxZoomFactor()	3244
6.1343.1.3 setGraphicsView()	3244
6.1343.1.4 setScaleFitToWindow()	3244
6.1344 Digikam::SketchWidget Class Reference	3245
6.1344.1 Member Function Documentation	3246
6.1344.1.1 setSketchImageFromXML()	3246
6.1344.1.2 sketchImageToXML()	3246
6.1345 Digikam::SlideVideo Class Reference	3247
6.1346 Digikam::SoftProofDialog Class Reference	3248
6.1347 Digikam::SolidHardwareDlg Class Reference	3249
6.1348 Digikam::SolidVolumeInfo Class Reference	3250
6.1349 Digikam::SparseModelIndexVector Class Reference	3250
6.1350 Digikam::SpellCheckConfig Class Reference	3251
6.1351 Digikam::SqueezedComboBox Class Reference	3252
6.1351.1 Detailed Description	3253
6.1351.2 Constructor & Destructor Documentation	3253

6.1351.2.1 SqueezedComboBox()	3253
6.1351.2.2 ~SqueezedComboBox()	3253
6.1351.3 Member Function Documentation	3253
6.1351.3.1 addSqueezedItem()	3253
6.1351.3.2 contains()	3254
6.1351.3.3 findOriginalText()	3254
6.1351.3.4 insertSqueezedItem()	3254
6.1351.3.5 insertSqueezedList()	3255
6.1351.3.6 item()	3255
6.1351.3.7 itemHighlighted()	3255
6.1351.3.8 setCurrent()	3255
6.1351.3.9 sizeHint()	3256
6.1352 Digikam::StackedView Class Reference	3256
6.1352.1 Member Function Documentation	3257
6.1352.1.1 isInSingleFileMode()	3257
6.1353 Digikam::StartScanPage Class Reference	3258
6.1354 Digikam::State Struct Reference	3259
6.1355 Digikam::StateSavingObject Class Reference	3259
6.1355.1 Detailed Description	3260
6.1355.2 Member Enumeration Documentation	3260
6.1355.2.1 StateSavingDepth	3260
6.1355.3 Constructor & Destructor Documentation	3261
6.1355.3.1 StateSavingObject()	3261
6.1355.3.2 ~StateSavingObject()	3261
6.1355.4 Member Function Documentation	3261
6.1355.4.1 doLoadState()	3261
6.1355.4.2 doSaveState()	3262
6.1355.4.3 entryName()	3262
6.1355.4.4 getConfigGroup()	3262
6.1355.4.5 getStateSavingDepth()	3262
6.1355.4.6 loadState()	3263
6.1355.4.7 saveState()	3263
6.1355.4.8 setConfigGroup()	3263
6.1355.4.9 setEntryPrefix()	3263
6.1355.4.10 setStateSavingDepth()	3263
6.1356 Digikam::StatusBarProgressWidget Class Reference	3264
6.1357 Digikam::StatusProgressBar Class Reference	3265
6.1358 Digikam::StayPoppedUpComboBox Class Reference	3267
6.1358.1 Constructor & Destructor Documentation	3268
6.1358.1.1 StayPoppedUpComboBox()	3268
6.1358.2 Member Function Documentation	3268
6.1358.2.1 installView()	3268

---

6.1358.2.2 sendViewportEventToView()	3269
6.1359 Digikam::StretchFilter Class Reference	3270
6.1359.1 Member Function Documentation	3273
6.1359.1.1 filterAction()	3273
6.1359.1.2 filterIdentifier()	3273
6.1359.1.3 readParameters()	3273
6.1360 Digikam::StyleSheetDebugger Class Reference	3273
6.1360.1 Constructor & Destructor Documentation	3274
6.1360.1.1 StyleSheetDebugger()	3274
6.1361 Digikam::SubjectData Class Reference	3274
6.1362 Digikam::SubjectEdit Class Reference	3275
6.1363 Digikam::SubjectWidget Class Reference	3277
6.1364 Digikam::SubQueryBuilder Class Reference	3278
6.1365 Digikam::SyncJob Class Reference	3279
6.1365.1 Member Function Documentation	3279
6.1365.1.1 getTagThumbnail()	3279
6.1366 Digikam::SystemSettings Class Reference	3279
6.1366.1 Member Enumeration Documentation	3280
6.1366.1.1 ProxyType	3280
6.1367 Digikam::SystemSettingsWidget Class Reference	3281
6.1368 Digikam::TableView Class Reference	3282
6.1368.1 Member Function Documentation	3284
6.1368.1.1 doLoadState()	3284
6.1368.1.2 doSaveState()	3284
6.1368.1.3 invertSelection()	3284
6.1368.1.4 selectAll()	3285
6.1368.1.5 slotAwayFromSelection	3285
6.1368.1.6 slotDeleteSelected	3285
6.1368.1.7 slotSetCurrentWhenAvailable	3285
6.1369 Digikam::TableViewColumn Class Reference	3285
6.1369.1 Member Function Documentation	3286
6.1369.1.1 columnAffectedByChangeset()	3286
6.1369.1.2 compare()	3287
6.1369.1.3 data()	3287
6.1369.1.4 getColumnFlags()	3287
6.1369.1.5 paint()	3287
6.1369.1.6 sizeHint()	3287
6.1369.1.7 updateThumbnailSize()	3287
6.1370 Digikam::TableViewColumnConfiguration Class Reference	3288
6.1371 Digikam::TableViewColumnConfigurationWidget Class Reference	3288
6.1372 Digikam::TableViewColumnDescription Class Reference	3289
6.1373 Digikam::TableViewColumnFactory Class Reference	3289



---

6.1374 Digikam::TableViewColumnProfile Class Reference	3290
6.1374.1 Member Function Documentation	3290
6.1374.1.1 loadSettings()	3290
6.1375 Digikam::TableViewColumns::ColumnAudioVideoProperties Class Reference	3291
6.1375.1 Member Function Documentation	3293
6.1375.1.1 compare()	3293
6.1375.1.2 data()	3293
6.1375.1.3 getColumnFlags()	3293
6.1375.1.4 getTitle()	3293
6.1375.1.5 setConfiguration()	3294
6.1376 Digikam::TableViewColumns::ColumnDigikamProperties Class Reference	3295
6.1376.1 Member Function Documentation	3297
6.1376.1.1 columnAffectedByChangeset()	3297
6.1376.1.2 compare()	3297
6.1376.1.3 data()	3297
6.1376.1.4 getColumnFlags()	3297
6.1376.1.5 getDescription()	3298
6.1376.1.6 getTitle()	3298
6.1377 Digikam::TableViewColumns::ColumnFileConfigurationWidget Class Reference	3298
6.1377.1 Member Function Documentation	3299
6.1377.1.1 getNewConfiguration()	3299
6.1378 Digikam::TableViewColumns::ColumnFileProperties Class Reference	3300
6.1378.1 Member Function Documentation	3302
6.1378.1.1 compare()	3302
6.1378.1.2 data()	3302
6.1378.1.3 getColumnFlags()	3302
6.1378.1.4 getConfigurationWidget()	3302
6.1378.1.5 getTitle()	3303
6.1378.1.6 setConfiguration()	3303
6.1379 Digikam::TableViewColumns::ColumnGeoConfigurationWidget Class Reference	3303
6.1379.1 Member Function Documentation	3304
6.1379.1.1 getNewConfiguration()	3304
6.1380 Digikam::TableViewColumns::ColumnGeoProperties Class Reference	3305
6.1380.1 Member Function Documentation	3307
6.1380.1.1 compare()	3307
6.1380.1.2 data()	3307
6.1380.1.3 getColumnFlags()	3307
6.1380.1.4 getConfigurationWidget()	3307
6.1380.1.5 getTitle()	3308
6.1380.1.6 setConfiguration()	3308
6.1381 Digikam::TableViewColumns::ColumnItemProperties Class Reference	3309
6.1381.1 Member Function Documentation	3311

---

6.1381.1.1 compare()	3311
6.1381.1.2 data()	3311
6.1381.1.3 getColumnFlags()	3311
6.1381.1.4 getTitle()	3311
6.1382 Digikam::TableViewColumns::ColumnPhotoConfigurationWidget Class Reference	3312
6.1382.1 Member Function Documentation	3313
6.1382.1.1 getNewConfiguration()	3313
6.1383 Digikam::TableViewColumns::ColumnPhotoProperties Class Reference	3314
6.1383.1 Member Function Documentation	3316
6.1383.1.1 compare()	3316
6.1383.1.2 data()	3316
6.1383.1.3 getColumnFlags()	3316
6.1383.1.4 getConfigurationWidget()	3316
6.1383.1.5 getTitle()	3317
6.1383.1.6 setConfiguration()	3317
6.1384 Digikam::TableViewColumns::ColumnThumbnail Class Reference	3318
6.1384.1 Member Function Documentation	3320
6.1384.1.1 data()	3320
6.1384.1.2 getColumnFlags()	3320
6.1384.1.3 getTitle()	3320
6.1384.1.4 paint()	3320
6.1384.1.5 sizeHint()	3320
6.1384.1.6 updateThumbnailSize()	3321
6.1385 Digikam::TableViewConfigurationDialog Class Reference	3321
6.1386 Digikam::TableViewItemDelegate Class Reference	3322
6.1386.1 Member Function Documentation	3322
6.1386.1.1 sizeHint()	3322
6.1387 Digikam::TableViewModel Class Reference	3323
6.1387.1 Member Function Documentation	3325
6.1387.1.1 addColumnAt()	3325
6.1387.1.2 flags()	3325
6.1387.1.3 indexFromImageId()	3325
6.1387.1.4 infoFromItem()	3325
6.1387.1.5 loadColumnProfile()	3325
6.1387.1.6 parent()	3325
6.1387.1.7 sort()	3325
6.1388 Digikam::TableViewModel::Item Class Reference	3326
6.1389 Digikam::TableViewSelectionModelSyncer Class Reference	3326
6.1389.1 Constructor & Destructor Documentation	3327
6.1389.1.1 TableViewSelectionModelSyncer()	3327
6.1390 Digikam::TableViewShared Class Reference	3327
6.1391 Digikam::TableViewTreeView Class Reference	3328

---

6.1391.1 Detailed Description . . . . .	3329
6.1391.2 Member Function Documentation . . . . .	3329
6.1391.2.1 dragDropHandler() . . . . .	3329
6.1391.2.2 hasHiddenGroupedImages() . . . . .	3329
6.1391.2.3 mapIndexForDragDrop() . . . . .	3330
6.1391.2.4 pixmapForDrag() . . . . .	3330
6.1392 Digikam::TagChangeset Class Reference . . . . .	3330
6.1392.1 Member Enumeration Documentation . . . . .	3330
6.1392.1.1 Operation . . . . .	3330
6.1393 Digikam::TagCheckView Class Reference . . . . .	3331
6.1393.1 Member Function Documentation . . . . .	3336
6.1393.1.1 addCustomContextMenuActions() . . . . .	3336
6.1393.1.2 checkedTagsChanged . . . . .	3336
6.1393.1.3 doLoadState() . . . . .	3337
6.1393.1.4 doSaveState() . . . . .	3337
6.1393.1.5 setCheckNewTags() . . . . .	3337
6.1394 Digikam::TagCompleter Class Reference . . . . .	3338
6.1394.1 Constructor & Destructor Documentation . . . . .	3338
6.1394.1.1 TagCompleter() . . . . .	3338
6.1394.2 Member Function Documentation . . . . .	3339
6.1394.2.1 setContextParentTag() . . . . .	3339
6.1394.2.2 setSupportingTagModel() . . . . .	3339
6.1394.2.3 update() . . . . .	3339
6.1395 Digikam::TagData Struct Reference . . . . .	3339
6.1396 Digikam::TagDragDropHandler Class Reference . . . . .	3340
6.1396.1 Member Function Documentation . . . . .	3341
6.1396.1.1 accepts() . . . . .	3341
6.1396.1.2 createMimeData() . . . . .	3341
6.1396.1.3 dropEvent() . . . . .	3341
6.1396.1.4 mimeTypes() . . . . .	3342
6.1397 Digikam::TagEditDlg Class Reference . . . . .	3342
6.1397.1 Member Function Documentation . . . . .	3343
6.1397.1.1 createTAlbum() . . . . .	3343
6.1398 Digikam::TagFilterView Class Reference . . . . .	3344
6.1398.1 Detailed Description . . . . .	3349
6.1398.2 Constructor & Destructor Documentation . . . . .	3349
6.1398.2.1 TagFilterView() . . . . .	3349
6.1398.2.2 ~TagFilterView() . . . . .	3350
6.1398.3 Member Function Documentation . . . . .	3350
6.1398.3.1 addCustomContextMenuActions() . . . . .	3350
6.1398.3.2 handleCustomContextMenuAction() . . . . .	3350
6.1399 Digikam::TagFolderView Class Reference . . . . .	3351

---

6.1399.1 Constructor & Destructor Documentation	3355
6.1399.1.1 TagFolderView()	3355
6.1399.1.2 ~TagFolderView()	3355
6.1399.2 Member Function Documentation	3356
6.1399.2.1 addCustomContextMenuActions()	3356
6.1399.2.2 contextMenuEvent()	3356
6.1399.2.3 contextMenuTitle()	3356
6.1399.2.4 handleCustomContextMenuAction()	3357
6.1399.2.5 setContextMenuItems()	3357
6.1399.2.6 setShowDeleteFaceTagsAction()	3357
6.1399.2.7 setShowFindDuplicateAction()	3357
6.1399.2.8 tagPropsEdit()	3358
6.1400 Digikam::TaggingAction Class Reference	3358
6.1400.1 Member Enumeration Documentation	3358
6.1400.1.1 Type	3358
6.1400.2 Constructor & Destructor Documentation	3359
6.1400.2.1 TaggingAction() [1/3]	3359
6.1400.2.2 TaggingAction() [2/3]	3359
6.1400.2.3 TaggingAction() [3/3]	3359
6.1401 Digikam::TaggingActionFactory Class Reference	3359
6.1401.1 Member Enumeration Documentation	3360
6.1401.1.1 NameMatchMode	3360
6.1401.2 Member Function Documentation	3360
6.1401.2.1 setConstraintInterface()	3360
6.1402 Digikam::TaggingActionFactory::ConstraintInterface Class Reference	3361
6.1403 Digikam::TagInfo Class Reference	3361
6.1403.1 Detailed Description	3362
6.1404 Digikam::TagList Class Reference	3362
6.1404.1 Member Function Documentation	3363
6.1404.1.1 restoreSettings()	3363
6.1405 Digikam::TagMgrListModel Class Reference	3363
6.1405.1 Member Function Documentation	3364
6.1405.1.1 addItem()	3364
6.1405.1.2 data()	3364
6.1405.1.3 dropMimeData()	3365
6.1405.1.4 supportedDropActions()	3365
6.1406 Digikam::TagMgrListView Class Reference	3365
6.1406.1 Member Function Documentation	3366
6.1406.1.1 startDrag()	3366
6.1407 Digikam::TagMgrTreeView Class Reference	3367
6.1407.1 Member Function Documentation	3372
6.1407.1.1 contextMenuEvent()	3372

---

6.1407.1.2 setContextMenuItems()	3372
6.1408 Digikam::TagModel Class Reference	3373
6.1408.1 Constructor & Destructor Documentation	3378
6.1408.1.1 TagModel()	3378
6.1408.2 Member Function Documentation	3378
6.1408.2.1 albumData()	3378
6.1408.2.2 albumForId()	3378
6.1408.2.3 decorationRoleData()	3378
6.1408.2.4 fontRoleData()	3378
6.1409 Digikam::TagModificationHelper Class Reference	3379
6.1409.1 Detailed Description	3380
6.1409.2 Constructor & Destructor Documentation	3380
6.1409.2.1 TagModificationHelper()	3380
6.1409.2.2 ~TagModificationHelper()	3381
6.1409.3 Member Function Documentation	3381
6.1409.3.1 bindMultipleTags()	3381
6.1409.3.2 bindTag()	3381
6.1409.3.3 boundMultipleTags()	3381
6.1409.3.4 boundTag()	3381
6.1409.3.5 slotFaceTagDelete [1/2]	3382
6.1409.3.6 slotFaceTagDelete [2/2]	3382
6.1409.3.7 slotMultipleFaceTagDel [1/2]	3382
6.1409.3.8 slotMultipleFaceTagDel [2/2]	3382
6.1409.3.9 slotMultipleTagDel [1/2]	3382
6.1409.3.10 slotMultipleTagDel [2/2]	3382
6.1409.3.11 slotMultipleTagsToFaceTags [1/2]	3383
6.1409.3.12 slotMultipleTagsToFaceTags [2/2]	3383
6.1409.3.13 slotTagDelete [1/2]	3383
6.1409.3.14 slotTagDelete [2/2]	3383
6.1409.3.15 slotTagEdit [1/2]	3384
6.1409.3.16 slotTagEdit [2/2]	3384
6.1409.3.17 slotTagNew [1/2]	3384
6.1409.3.18 slotTagNew [2/2]	3384
6.1409.3.19 slotTagToFaceTag [1/2]	3385
6.1409.3.20 slotTagToFaceTag [2/2]	3385
6.1410 Digikam::TagProperties Class Reference	3385
6.1410.1 Constructor & Destructor Documentation	3386
6.1410.1.1 TagProperties() [1/2]	3386
6.1410.1.2 TagProperties() [2/2]	3386
6.1410.2 Member Function Documentation	3386
6.1410.2.1 addProperty()	3386
6.1410.2.2 getOrCreate()	3386

---

6.1410.2.3 hasProperty() [1/2]	3386
6.1410.2.4 hasProperty() [2/2]	3387
6.1410.2.5 value()	3387
6.1411 Digikam::TagPropertiesFilterModel Class Reference	3388
6.1411.1 Detailed Description	3391
6.1411.2 Member Function Documentation	3391
6.1411.2.1 isFiltering()	3391
6.1411.2.2 matches()	3391
6.1412 Digikam::TagProperty Class Reference	3391
6.1413 Digikam::TagPropertyName Class Reference	3392
6.1414 Digikam::TagPropWidget Class Reference	3392
6.1415 Digikam::TagRegion Class Reference	3393
6.1415.1 Constructor & Destructor Documentation	3394
6.1415.1.1 TagRegion() [1/3]	3394
6.1415.1.2 TagRegion() [2/3]	3394
6.1415.1.3 TagRegion() [3/3]	3394
6.1415.2 Member Function Documentation	3394
6.1415.2.1 absoluteToRelative()	3394
6.1415.2.2 adjustToOrientation()	3394
6.1415.2.3 intersects()	3395
6.1415.2.4 mapToOriginalSize() [1/2]	3395
6.1415.2.5 mapToOriginalSize() [2/2]	3395
6.1415.2.6 relativeToAbsolute() [1/2]	3395
6.1415.2.7 relativeToAbsolute() [2/2]	3395
6.1415.2.8 reverseToOrientation()	3395
6.1415.2.9 toRect()	3396
6.1415.2.10 toVariant()	3396
6.1415.2.11 toXml()	3396
6.1416 Digikam::TagsActionMngr Class Reference	3396
6.1416.1 Member Function Documentation	3397
6.1416.1.1 actionCollections()	3397
6.1416.1.2 registerLabelsActions()	3397
6.1416.1.3 registerTagsActionCollections()	3397
6.1416.1.4 updateTagShortcut()	3398
6.1417 Digikam::TagsCache Class Reference	3398
6.1417.1 Member Enumeration Documentation	3400
6.1417.1.1 LeadingSlashPolicy	3400
6.1417.2 Member Function Documentation	3400
6.1417.2.1 canBeWrittenToMetadata()	3400
6.1417.2.2 colorLabelForTag()	3400
6.1417.2.3 colorLabelFromTags()	3400
6.1417.2.4 colorLabelTags()	3401

---

6.1417.2.5 containsPublicTags()	3401
6.1417.2.6 createTag()	3401
6.1417.2.7 getOrCreateInternalTag()	3401
6.1417.2.8 getOrCreateTag()	3401
6.1417.2.9 getOrCreateTagWithProperty()	3401
6.1417.2.10 hasProperty()	3402
6.1417.2.11 hasTag()	3402
6.1417.2.12 isInternalTag()	3402
6.1417.2.13 parentTag()	3402
6.1417.2.14 parentTags()	3402
6.1417.2.15 pickLabelForTag()	3402
6.1417.2.16 pickLabelFromTags()	3403
6.1417.2.17 pickLabelTags()	3403
6.1417.2.18 properties()	3403
6.1417.2.19 propertyValue()	3403
6.1417.2.20 publicTags()	3403
6.1417.2.21 shortenedTagPaths()	3403
6.1417.2.22 tagAdded	3404
6.1417.2.23 tagForColorLabel()	3404
6.1417.2.24 tagForName()	3404
6.1417.2.25 tagForPath()	3404
6.1417.2.26 tagForPickLabel()	3404
6.1417.2.27 tagName()	3404
6.1417.2.28 tagPath()	3405
6.1417.2.29 tagsContaining()	3405
6.1417.2.30 tagsForName()	3405
6.1417.2.31 tagsWithProperty()	3405
6.1417.2.32 tagsWithPropertyCached()	3405
6.1418 Digikam::TagsDBJobInfo Class Reference	3406
6.1419 Digikam::TagsDBJobsThread Class Reference	3407
6.1419.1 Member Function Documentation	3408
6.1419.1.1 tagsListing()	3408
6.1420 Digikam::TagsEdit Class Reference	3409
6.1421 Digikam::TagShortInfo Class Reference	3410
6.1422 Digikam::TagsJob Class Reference	3410
6.1423 Digikam::TagsLineEditOverlay Class Reference	3412
6.1423.1 Member Function Documentation	3414
6.1423.1.1 createWidget()	3414
6.1423.1.2 hide()	3415
6.1423.1.3 setActive()	3415
6.1423.1.4 slotEntered()	3415
6.1423.1.5 visualChange()	3415

---

6.1424 Digikam::TagsManager Class Reference . . . . .	3416
6.1424.1 Member Function Documentation . . . . .	3417
6.1424.1.1 doLoadState() . . . . .	3417
6.1424.1.2 doSaveState() . . . . .	3418
6.1425 Digikam::TagsManagerFilterModel Class Reference . . . . .	3419
6.1425.1 Member Function Documentation . . . . .	3422
6.1425.1.1 matches() . . . . .	3422
6.1426 Digikam::TagsPopupMenu Class Reference . . . . .	3422
6.1426.1 Member Enumeration Documentation . . . . .	3423
6.1426.1.1 Mode . . . . .	3423
6.1427 Digikam::TagTreeView Class Reference . . . . .	3424
6.1427.1 Member Function Documentation . . . . .	3428
6.1427.1.1 filteredModel() . . . . .	3428
6.1428 Digikam::TagTreeViewSelectComboBox Class Reference . . . . .	3429
6.1429 Digikam::TagViewSideBarWidget Class Reference . . . . .	3432
6.1429.1 Member Function Documentation . . . . .	3434
6.1429.1.1 applySettings() . . . . .	3434
6.1429.1.2 changeAlbumFromHistory() . . . . .	3434
6.1429.1.3 doLoadState() . . . . .	3434
6.1429.1.4 doSaveState() . . . . .	3434
6.1429.1.5 getCaption() . . . . .	3434
6.1429.1.6 getIcon() . . . . .	3435
6.1429.1.7 setActive() . . . . .	3435
6.1430 Digikam::TAlbum Class Reference . . . . .	3436
6.1430.1 Detailed Description . . . . .	3438
6.1430.2 Member Function Documentation . . . . .	3438
6.1430.2.1 databaseUrl() . . . . .	3438
6.1430.2.2 tagPath() . . . . .	3438
6.1431 Digikam::Template Class Reference . . . . .	3439
6.1431.1 Member Function Documentation . . . . .	3440
6.1431.1.1 isEmpty() . . . . .	3440
6.1431.1.2 isNull() . . . . .	3440
6.1431.1.3 merge() . . . . .	3440
6.1431.1.4 operator==( ) . . . . .	3440
6.1431.2 Member Data Documentation . . . . .	3440
6.1431.2.1 m_authors . . . . .	3440
6.1431.2.2 m_authorsPosition . . . . .	3440
6.1431.2.3 m_contactInfo . . . . .	3440
6.1431.2.4 m_copyright . . . . .	3441
6.1431.2.5 m_credit . . . . .	3441
6.1431.2.6 m_instructions . . . . .	3441
6.1431.2.7 m_locationInfo . . . . .	3441



---

6.1431.2.8 m_rightUsageTerms	3441
6.1431.2.9 m_source	3441
6.1431.2.10 m_subjects	3441
6.1431.2.11 m_templateTitle	3441
6.1432 Digikam::TemplateList Class Reference	3442
6.1433 Digikam::TemplateListItem Class Reference	3443
6.1434 Digikam::TemplateManager Class Reference	3444
6.1435 Digikam::TemplatePanel Class Reference	3445
6.1436 Digikam::TemplateSelector Class Reference	3446
6.1437 Digikam::TemplateViewer Class Reference	3448
6.1438 Digikam::TextFilter Class Reference	3450
6.1439 Digikam::TextureContainer Class Reference	3451
6.1440 Digikam::TextureFilter Class Reference	3452
6.1440.1 Member Function Documentation	3455
6.1440.1.1 filterAction()	3455
6.1440.1.2 filterIdentifier()	3455
6.1440.1.3 readParameters()	3455
6.1441 Digikam::TextureSettings Class Reference	3455
6.1442 Digikam::ThemeManager Class Reference	3456
6.1443 Digikam::ThemeManager::Private Class Reference	3457
6.1444 Digikam::ThreadManager Class Reference	3458
6.1445 Digikam::ThumbBarDock Class Reference	3459
6.1445.1 Detailed Description	3460
6.1445.2 Member Function Documentation	3460
6.1445.2.1 getToggleAction()	3460
6.1445.2.2 reinitialize()	3460
6.1445.2.3 shouldBeVisible()	3460
6.1446 Digikam::ThumbnailAligningDelegate Class Reference	3461
6.1447 Digikam::ThumbnailCreator Class Reference	3461
6.1447.1 Constructor & Destructor Documentation	3462
6.1447.1.1 ThumbnailCreator() [1/2]	3462
6.1447.1.2 ThumbnailCreator() [2/2]	3462
6.1447.2 Member Function Documentation	3462
6.1447.2.1 deleteThumbnailsFromDisk()	3462
6.1447.2.2 errorString()	3462
6.1447.2.3 fileThumbnailInfo()	3463
6.1447.2.4 identifierForDetail()	3463
6.1447.2.5 load()	3463
6.1447.2.6 loadDetail()	3463
6.1447.2.7 pregenerate()	3463
6.1447.2.8 setExifRotate()	3463
6.1447.2.9 setLoadingProperties()	3464

---

6.1447.2.10 setOnlyLargeThumbnails()	3464
6.1447.2.11 setRemoveAlphaChannel()	3464
6.1447.2.12 setThumbnailInfoProvider()	3464
6.1447.2.13 setThumbnailSize()	3464
6.1447.2.14 store()	3464
6.1447.2.15 storedSize()	3465
6.1447.2.16 thumbnailSize()	3465
6.1448 Digikam::ThumbnailCreator::Private Class Reference	3465
6.1449 Digikam::ThumbnailIdentifier Class Reference	3466
6.1449.1 Member Data Documentation	3466
6.1449.1.1 filePath	3466
6.1449.1.2 id	3467
6.1450 Digikam::ThumbnailImage Class Reference	3467
6.1451 Digikam::ThumbnailImageCatcher Class Reference	3467
6.1451.1 Constructor & Destructor Documentation	3468
6.1451.1.1 ThumbnailImageCatcher()	3468
6.1451.2 Member Function Documentation	3468
6.1451.2.1 cancel	3468
6.1451.2.2 enqueue()	3469
6.1451.2.3 setActive	3469
6.1452 Digikam::ThumbnailImageCatcher::Private Class Reference	3469
6.1453 Digikam::ThumbnailImageCatcher::Private::CatcherResult Class Reference	3469
6.1454 Digikam::ThumbnailInfo Class Reference	3470
6.1454.1 Member Data Documentation	3471
6.1454.1.1 customIdentifier	3471
6.1454.1.2 fileName	3471
6.1454.1.3 isAccessible	3471
6.1454.1.4 mimeType	3471
6.1454.1.5 modificationDate	3471
6.1454.1.6 orientationHint	3472
6.1454.1.7 uniqueHash	3472
6.1455 Digikam::ThumbnailInfoProvider Class Reference	3472
6.1456 Digikam::ThumbnailLoadingTask Class Reference	3473
6.1456.1 Member Function Documentation	3475
6.1456.1.1 execute()	3475
6.1456.1.2 postProcess()	3475
6.1457 Digikam::ThumbnailLoadThread Class Reference	3476
6.1457.1 Member Function Documentation	3480
6.1457.1.1 defaultThread()	3480
6.1457.1.2 deleteThumbnail()	3481
6.1457.1.3 find() [1/5]	3481
6.1457.1.4 find() [2/5]	3481

6.1457.1.5 find() [3/5]	3481
6.1457.1.6 find() [4/5]	3481
6.1457.1.7 find() [5/5]	3482
6.1457.1.8 findBuffered()	3482
6.1457.1.9 findGroup()	3482
6.1457.1.10 initializeNoThumbnailStorage()	3482
6.1457.1.11 initializeThumbnailDatabase()	3482
6.1457.1.12 lastDescriptions()	3482
6.1457.1.13 load()	3483
6.1457.1.14 maximumThumbnailSize()	3483
6.1457.1.15 pixmapToThumbnailSize()	3483
6.1457.1.16 pregenerateGroup()	3483
6.1457.1.17 preload()	3483
6.1457.1.18 setDisplayingWidget()	3483
6.1457.1.19 setHighlightPixmap()	3484
6.1457.1.20 setPixmapRequested()	3484
6.1457.1.21 setSendSurrogatePixmap()	3484
6.1457.1.22 setThumbnailSize()	3484
6.1457.1.23 storeDetailThumbnail()	3484
6.1457.1.24 thumbnailLoaded()	3485
6.1457.1.25 thumbnailToPixmapSize()	3485
6.1458 Digikam::ThumbnailLoadThread::Private Class Reference	3485
6.1459 Digikam::ThumbnailLoadThreadStaticPriv Class Reference	3485
6.1460 Digikam::ThumbnailResult Class Reference	3486
6.1461 Digikam::ThumbnailSize Class Reference	3486
6.1461.1 Member Enumeration Documentation	3486
6.1461.1.1 Size	3486
6.1462 Digikam::ThumbsDb Class Reference	3487
6.1462.1 Member Function Documentation	3487
6.1462.1.1 findAll()	3487
6.1462.1.2 findByFilePath()	3488
6.1462.1.3 integrityCheck()	3488
6.1462.1.4 removeByFilePath()	3488
6.1462.1.5 removeByUniqueHash()	3488
6.1462.1.6 vacuum()	3488
6.1463 Digikam::ThumbsDbAccess Class Reference	3488
6.1463.1 Constructor & Destructor Documentation	3489
6.1463.1.1 ThumbsDbAccess()	3489
6.1463.2 Member Function Documentation	3489
6.1463.2.1 setLastError()	3489
6.1464 Digikam::ThumbsDbBackend Class Reference	3490
6.1464.1 Member Function Documentation	3493

---

6.1464.1.1 initSchema() . . . . .	3493
6.1465 Digikam::ThumbsDbInfo Class Reference . . . . .	3493
6.1466 Digikam::ThumbsDbInfoProvider Class Reference . . . . .	3493
6.1466.1 Member Function Documentation . . . . .	3494
6.1466.1.1 thumbnailInfo() . . . . .	3494
6.1467 Digikam::ThumbsDbSchemaUpdater Class Reference . . . . .	3494
6.1468 Digikam::ThumbsGenerator Class Reference . . . . .	3495
6.1468.1 Constructor & Destructor Documentation . . . . .	3497
6.1468.1.1 ThumbsGenerator() [1/2] . . . . .	3497
6.1468.1.2 ThumbsGenerator() [2/2] . . . . .	3498
6.1468.2 Member Function Documentation . . . . .	3498
6.1468.2.1 setUseMultiCoreCPU() . . . . .	3498
6.1469 Digikam::ThumbsTask Class Reference . . . . .	3498
6.1470 Digikam::TileGrouper Class Reference . . . . .	3500
6.1470.1 Member Function Documentation . . . . .	3500
6.1470.1.1 updateClusters() . . . . .	3500
6.1471 Digikam::TileIndex Class Reference . . . . .	3500
6.1472 Digikam::TimeAdjustContainer Class Reference . . . . .	3501
6.1472.1 Detailed Description . . . . .	3502
6.1472.2 Member Function Documentation . . . . .	3502
6.1472.2.1 atLeastOneUpdateToProcess() . . . . .	3502
6.1472.3 Member Data Documentation . . . . .	3502
6.1472.3.1 enableExifTool . . . . .	3502
6.1473 Digikam::TimeAdjustSettings Class Reference . . . . .	3503
6.1473.1 Member Function Documentation . . . . .	3503
6.1473.1.1 detAdjustmentByClockPhotoUrl() . . . . .	3503
6.1474 Digikam::TimelineSideBarWidget Class Reference . . . . .	3504
6.1474.1 Member Function Documentation . . . . .	3505
6.1474.1.1 applySettings() . . . . .	3505
6.1474.1.2 changeAlbumFromHistory() . . . . .	3506
6.1474.1.3 doLoadState() . . . . .	3506
6.1474.1.4 doSaveState() . . . . .	3506
6.1474.1.5 getCaption() . . . . .	3506
6.1474.1.6 getIcon() . . . . .	3506
6.1474.1.7 setActive() . . . . .	3506
6.1475 Digikam::TimeLineWidget Class Reference . . . . .	3507
6.1475.1 Member Enumeration Documentation . . . . .	3508
6.1475.1.1 ScaleMode . . . . .	3508
6.1475.1.2 SelectionMode . . . . .	3509
6.1475.2 Member Function Documentation . . . . .	3509
6.1475.2.1 selectedDateRange() . . . . .	3509
6.1476 Digikam::TimeZoneComboBox Class Reference . . . . .	3509

---

6.1477 Digikam::Token Class Reference	3510
6.1477.1 Detailed Description	3511
6.1477.2 Member Function Documentation	3511
6.1477.2.1 action()	3511
6.1477.2.2 description()	3511
6.1477.2.3 id()	3511
6.1477.2.4 signalTokenTriggered	3512
6.1478 Digikam::TonalityContainer Class Reference	3512
6.1479 Digikam::TonalityFilter Class Reference	3513
6.1479.1 Member Function Documentation	3516
6.1479.1.1 filterAction()	3516
6.1479.1.2 filterIdentifier()	3516
6.1479.1.3 readParameters()	3516
6.1480 Digikam::ToolListViewGroup Class Reference	3516
6.1481 Digikam::ToolListViewItem Class Reference	3517
6.1482 Digikam::ToolSettingsView Class Reference	3518
6.1483 Digikam::ToolsListView Class Reference	3519
6.1484 Digikam::ToolsView Class Reference	3520
6.1485 Digikam::TooltipCreator Class Reference	3521
6.1486 Digikam::TooltipDialog Class Reference	3521
6.1487 Digikam::TooltipsPage Class Reference	3522
6.1488 Digikam::TrackCorrelator Class Reference	3523
6.1489 Digikam::TrackCorrelator::Correlation Class Reference	3524
6.1490 Digikam::TrackCorrelator::CorrelationOptions Class Reference	3524
6.1491 Digikam::TrackCorrelatorThread Class Reference	3525
6.1492 Digikam::TrackListModel Class Reference	3526
6.1492.1 Member Function Documentation	3527
6.1492.1.1 headerData()	3527
6.1492.1.2 index()	3527
6.1493 Digikam::TrackManager Class Reference	3527
6.1493.1 Member Function Documentation	3528
6.1493.1.1 clear()	3528
6.1494 Digikam::TrackManager::Track Class Reference	3528
6.1495 Digikam::TrackManager::TrackPoint Class Reference	3529
6.1496 Digikam::TrackReader Class Reference	3529
6.1497 Digikam::TrackReader::TrackReadResult Class Reference	3530
6.1498 Digikam::TrainerWorker Class Reference	3531
6.1498.1 Member Function Documentation	3533
6.1498.1.1 aboutToDeactivate()	3533
6.1498.1.2 process	3533
6.1499 Digikam::TrainingDataProvider Class Reference	3534
6.1499.1 Detailed Description	3534

---

6.1499.2 Member Function Documentation . . . . .	3534
6.1499.2.1 images() . . . . .	3534
6.1499.2.2 newImages() . . . . .	3535
6.1500 Digikam::TransactionItem Class Reference . . . . .	3536
6.1500.1 Member Function Documentation . . . . .	3537
6.1500.1.1 setStatus() . . . . .	3537
6.1501 Digikam::TransactionItemView Class Reference . . . . .	3538
6.1502 Digikam::TransitionMngr Class Reference . . . . .	3539
6.1503 Digikam::TransitionMngr::Private Class Reference . . . . .	3539
6.1504 Digikam::TransitionPreview Class Reference . . . . .	3541
6.1505 Digikam::TrashView Class Reference . . . . .	3542
6.1505.1 Member Function Documentation . . . . .	3543
6.1505.1.1 getThumbnailSize() . . . . .	3543
6.1505.1.2 lastSelectedItemUrl() . . . . .	3543
6.1505.1.3 model() . . . . .	3543
6.1505.1.4 setThumbnailSize() . . . . .	3543
6.1505.1.5 statusBarText() . . . . .	3543
6.1506 Digikam::TreeBranch Class Reference . . . . .	3544
6.1507 Digikam::TreeProxyModel Class Reference . . . . .	3544
6.1508 Digikam::TreeViewComboBox Class Reference . . . . .	3545
6.1508.1 Constructor & Destructor Documentation . . . . .	3546
6.1508.1.1 TreeViewComboBox() . . . . .	3546
6.1508.2 Member Function Documentation . . . . .	3547
6.1508.2.1 installView() . . . . .	3547
6.1508.2.2 sendViewportEventToView() . . . . .	3547
6.1508.2.3 view() . . . . .	3547
6.1509 Digikam::TreeViewLineEditComboBox Class Reference . . . . .	3548
6.1509.1 Constructor & Destructor Documentation . . . . .	3550
6.1509.1.1 TreeViewLineEditComboBox() . . . . .	3550
6.1509.2 Member Function Documentation . . . . .	3550
6.1509.2.1 installLineEdit() . . . . .	3550
6.1509.2.2 installView() . . . . .	3550
6.1509.2.3 setLineEditText() . . . . .	3550
6.1510 Digikam::TrimmedModifier Class Reference . . . . .	3551
6.1510.1 Member Function Documentation . . . . .	3552
6.1510.1.1 parseOperation() . . . . .	3552
6.1511 Digikam::TwoProgressItemsContainer Class Reference . . . . .	3553
6.1512 Digikam::UMSCamera Class Reference . . . . .	3555
6.1512.1 Detailed Description . . . . .	3557
6.1512.2 Member Function Documentation . . . . .	3557
6.1512.2.1 cameraAbout() . . . . .	3557
6.1512.2.2 cameraDriverType() . . . . .	3557

---

6.1512.2.3 cameraManual()	3557
6.1512.2.4 cameraMD5ID()	3558
6.1512.2.5 cameraSummary()	3558
6.1512.2.6 cancel()	3558
6.1512.2.7 capture()	3558
6.1512.2.8 deleteItem()	3558
6.1512.2.9 doConnect()	3558
6.1512.2.10 downloadItem()	3558
6.1512.2.11 getFolders()	3559
6.1512.2.12 getFreeSpace()	3559
6.1512.2.13 getItemInfo()	3559
6.1512.2.14 getItemsInfoList()	3559
6.1512.2.15 getMetadata()	3559
6.1512.2.16 getPreview()	3559
6.1512.2.17 getThumbnail()	3560
6.1512.2.18 setLockItem()	3560
6.1512.2.19 uploadItem()	3560
6.1513 Digikam::UndoAction Class Reference	3561
6.1514 Digikam::UndoActionIrreversible Class Reference	3562
6.1515 Digikam::UndoActionReversible Class Reference	3563
6.1516 Digikam::UndoCache Class Reference	3564
6.1516.1 Member Function Documentation	3564
6.1516.1.1 clear()	3564
6.1516.1.2 clearFrom()	3564
6.1516.1.3 getData()	3564
6.1516.1.4 putData()	3564
6.1517 Digikam::UndoManager Class Reference	3565
6.1518 Digikam::UndoMetadataContainer Class Reference	3565
6.1518.1 Member Function Documentation	3565
6.1518.1.1 fromImage()	3565
6.1518.1.2 toImage()	3566
6.1519 Digikam::UndoState Class Reference	3566
6.1520 Digikam::UniqueModifier Class Reference	3567
6.1520.1 Member Function Documentation	3568
6.1520.1.1 parseOperation()	3568
6.1520.1.2 reset()	3569
6.1521 Digikam::UnsharpMaskFilter Class Reference	3570
6.1521.1 Member Function Documentation	3573
6.1521.1.1 filterAction()	3573
6.1521.1.2 filterIdentifier()	3573
6.1521.1.3 readParameters()	3573
6.1522 Digikam::VersionFileInfo Class Reference	3573

6.1523 Digikam::VersionFileOperation Class Reference	3573
6.1523.1 Member Enumeration Documentation	3574
6.1523.1.1 Task	3574
6.1523.2 Constructor & Destructor Documentation	3574
6.1523.2.1 VersionFileOperation()	3574
6.1523.3 Member Function Documentation	3574
6.1523.3.1 allFilePaths()	3574
6.1524 Digikam::VersioningPromptUserSaveDialog Class Reference	3575
6.1525 Digikam::VersionItemFilterSettings Class Reference	3575
6.1525.1 Member Function Documentation	3576
6.1525.1.1 matches()	3576
6.1525.1.2 setExceptionList()	3576
6.1526 Digikam::VersionManager Class Reference	3577
6.1527 Digikam::VersionManagerSettings Class Reference	3578
6.1528 Digikam::VersionNamingScheme Class Reference	3579
6.1528.1 Constructor & Destructor Documentation	3580
6.1528.1.1 VersionNamingScheme()	3580
6.1528.2 Member Function Documentation	3580
6.1528.2.1 baseName()	3580
6.1528.2.2 directory()	3580
6.1528.2.3 incrementedCounter()	3580
6.1528.2.4 initialCounter()	3581
6.1528.2.5 intermediateFileName()	3581
6.1528.2.6 versionFileName()	3581
6.1529 Digikam::VersionsDelegate Class Reference	3582
6.1529.1 Member Function Documentation	3584
6.1529.1.1 asDelegate()	3584
6.1530 Digikam::VersionsTreeView Class Reference	3585
6.1530.1 Member Function Documentation	3586
6.1530.1.1 dragDropHandler()	3586
6.1530.1.2 mapIndexForDragDrop()	3587
6.1530.1.3 pixmapForDrag()	3587
6.1531 Digikam::VersionsWidget Class Reference	3587
6.1532 Digikam::VideoFrame Class Reference	3588
6.1533 Digikam::VideoInfoContainer Class Reference	3588
6.1534 Digikam::VideoMetadataContainer Class Reference	3589
6.1535 Digikam::VideoStripFilter Class Reference	3589
6.1536 Digikam::VideoThumbDecoder Class Reference	3589
6.1537 Digikam::VideoThumbDecoder::Private Class Reference	3590
6.1538 Digikam::VideoThumbnailer Class Reference	3590
6.1539 Digikam::VideoThumbWriter Class Reference	3591
6.1540 Digikam::VidPlayerDlg Class Reference	3591



---

6.1541 Digikam::VidSlideSettings Class Reference . . . . .	3591
6.1541.1 Member Enumeration Documentation . . . . .	3594
6.1541.1.1 Selection . . . . .	3594
6.1541.1.2 VidBitRate . . . . .	3594
6.1541.1.3 VidCodec . . . . .	3594
6.1541.1.4 VidFormat . . . . .	3595
6.1541.1.5 VidPlayer . . . . .	3595
6.1541.1.6 VidStd . . . . .	3595
6.1541.1.7 VidType . . . . .	3595
6.1541.2 Member Function Documentation . . . . .	3597
6.1541.2.1 readSettings() . . . . .	3597
6.1541.2.2 videoTypeNames() . . . . .	3597
6.1541.3 Member Data Documentation . . . . .	3597
6.1541.3.1 conflictRule . . . . .	3597
6.1541.3.2 outputDir . . . . .	3597
6.1542 Digikam::VidSlideTask Class Reference . . . . .	3598
6.1543 Digikam::VidSlideThread Class Reference . . . . .	3600
6.1543.1 Member Function Documentation . . . . .	3601
6.1543.1.1 prepareFrames() . . . . .	3601
6.1544 Digikam::VisibilityController Class Reference . . . . .	3602
6.1544.1 Member Function Documentation . . . . .	3603
6.1544.1.1 addObject() . . . . .	3603
6.1544.1.2 addWidget() . . . . .	3603
6.1544.1.3 isVisible() . . . . .	3603
6.1544.1.4 setContainerWidget() . . . . .	3603
6.1545 Digikam::VisibilityObject Class Reference . . . . .	3604
6.1546 Digikam::WBContainer Class Reference . . . . .	3604
6.1546.1 Member Data Documentation . . . . .	3605
6.1546.1.1 black . . . . .	3605
6.1547 Digikam::WBFilter Class Reference . . . . .	3606
6.1547.1 Member Function Documentation . . . . .	3609
6.1547.1.1 autoWBAdjustmentFromColor() . . . . .	3609
6.1547.1.2 filterAction() . . . . .	3609
6.1547.1.3 filterIdentifier() . . . . .	3609
6.1547.1.4 filterImage() . . . . .	3610
6.1547.1.5 readParameters() . . . . .	3610
6.1548 Digikam::WBSettings Class Reference . . . . .	3610
6.1549 Digikam::WebBrowserDlg Class Reference . . . . .	3611
6.1550 Digikam::WebWidget Class Reference . . . . .	3612
6.1551 Digikam::WelcomePage Class Reference . . . . .	3613
6.1552 Digikam::WelcomePageView Class Reference . . . . .	3614
6.1553 Digikam::WelcomePageViewPage Class Reference . . . . .	3615

---

6.1554 Digikam::WorkerObject Class Reference	3616
6.1554.1 Member Enumeration Documentation	3617
6.1554.1.1 DeactivatingMode	3617
6.1554.2 Constructor & Destructor Documentation	3617
6.1554.2.1 WorkerObject()	3617
6.1554.3 Member Function Documentation	3618
6.1554.3.1 aboutToDeactivate()	3618
6.1554.3.2 aboutToQuitLoop()	3618
6.1554.3.3 connectAndSchedule()	3618
6.1554.3.4 deactivate	3618
6.1554.3.5 schedule	3618
6.1554.3.6 setPriority()	3619
6.1554.3.7 shutDown()	3619
6.1555 Digikam::Workflow Class Reference	3619
6.1555.1 Detailed Description	3619
6.1556 Digikam::WorkflowDlg Class Reference	3620
6.1557 Digikam::WorkflowItem Class Reference	3621
6.1558 Digikam::WorkflowList Class Reference	3622
6.1559 Digikam::WorkflowManager Class Reference	3623
6.1559.1 Member Function Documentation	3624
6.1559.1.1 load()	3624
6.1559.1.2 save()	3624
6.1560 Digikam::WorkingWidget Class Reference	3624
6.1561 Digikam::WSAlbum Class Reference	3625
6.1561.1 Member Function Documentation	3625
6.1561.1.1 setBaseAlbum()	3625
6.1562 Digikam::WSComboBoxIntermediate Class Reference	3626
6.1562.1 Constructor & Destructor Documentation	3626
6.1562.1.1 WSComboBoxIntermediate()	3626
6.1562.2 Member Function Documentation	3626
6.1562.2.1 setIntermediate()	3626
6.1563 Digikam::WSLoginDialog Class Reference	3627
6.1564 Digikam::WSNewAlbumDialog Class Reference	3628
6.1565 Digikam::WSSelectUserDlg Class Reference	3629
6.1566 Digikam::WSSettings Class Reference	3630
6.1567 Digikam::WSSettingsWidget Class Reference	3632
6.1568 Digikam::WSToolDialog Class Reference	3634
6.1569 Digikam::WSToolUtils Class Reference	3635
6.1569.1 Member Function Documentation	3635
6.1569.1.1 randomString()	3635
6.1570 Digikam::XbelReader Class Reference	3636
6.1571 Digikam::XbelWriter Class Reference	3636

---

6.1572 Digikam::XmpWidget Class Reference	3637
6.1572.1 Member Function Documentation	3639
6.1572.1.1 getMetadataTitle()	3639
6.1572.1.2 getTagDescription()	3639
6.1572.1.3 getTagTitle()	3639
6.1572.1.4 loadFromURL()	3639
6.1573 ShowFoto::NoDuplicatesShowfotoFilterModel Class Reference	3640
6.1574 ShowFoto::Showfoto Class Reference	3643
6.1574.1 Member Function Documentation	3648
6.1574.1.1 infoface()	3648
6.1575 ShowFoto::Showfoto::Private Class Reference	3648
6.1576 ShowFoto::ShowfotoCategorizedView Class Reference	3649
6.1576.1 Member Function Documentation	3654
6.1576.1.1 addOverlay()	3654
6.1576.1.2 deselected	3654
6.1576.1.3 dragDropHandler()	3654
6.1576.1.4 filterModel()	3654
6.1576.1.5 hintAt	3654
6.1576.1.6 indexActivated()	3654
6.1576.1.7 modelChanged	3655
6.1576.1.8 nextIndexHint()	3655
6.1576.1.9 nextInOrder()	3655
6.1576.1.10 selected	3655
6.1576.1.11 setCurrentInfo	3655
6.1576.1.12 setCurrentUrl	3655
6.1576.1.13 setCurrentWhenAvailable	3656
6.1576.1.14 setSelectedShowfotoItemInfos	3656
6.1576.1.15 setSelectedUrls	3656
6.1576.1.16 showContextMenuOnIndex()	3656
6.1576.1.17 showfotoFilterModel()	3656
6.1576.1.18 showfotoItemInfoActivated	3656
6.1576.1.19 showfotoThumbnailModel()	3656
6.1576.1.20 toIndex()	3657
6.1577 ShowFoto::ShowfotoCoordinatesOverlay Class Reference	3657
6.1577.1 Member Function Documentation	3659
6.1577.1.1 checkIndex()	3659
6.1577.1.2 createWidget()	3659
6.1577.1.3 setActive()	3660
6.1577.1.4 slotEntered()	3660
6.1577.1.5 visualChange()	3660
6.1578 ShowFoto::ShowfotoCoordinatesOverlayWidget Class Reference	3660
6.1579 ShowFoto::ShowfotoDelegate Class Reference	3662

---

6.1579.1 Member Function Documentation	3665
6.1579.1.1 acceptsActivation()	3665
6.1579.1.2 acceptsToolTip()	3666
6.1579.1.3 clearCaches()	3666
6.1579.1.4 clearModelDataCaches()	3666
6.1579.1.5 imageInformationRect()	3666
6.1579.1.6 pixmapForDrag()	3666
6.1579.1.7 pixmapRect()	3666
6.1579.1.8 retrieveThumbnailPixmap()	3667
6.1579.1.9 setDefaultViewOptions()	3667
6.1579.1.10 updateContentWidth()	3667
6.1579.1.11 updateRects()	3667
6.1579.1.12 updateSizeRectsAndPixmapes()	3667
6.1580 ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate Class Reference	3668
6.1580.1 Member Function Documentation	3669
6.1580.1.1 clearRects()	3669
6.1581 ShowFoto::ShowfotoDragDropHandler Class Reference	3670
6.1581.1 Member Function Documentation	3671
6.1581.1.1 accepts()	3671
6.1581.1.2 createMimeData()	3671
6.1581.1.3 dropEvent()	3671
6.1581.1.4 mimeTypes()	3672
6.1582 ShowFoto::ShowfotoFilterModel Class Reference	3673
6.1582.1 Member Enumeration Documentation	3675
6.1582.1.1 ShowfotoFilterModelRoles	3675
6.1582.2 Member Function Documentation	3676
6.1582.2.1 categoryIdentifier()	3676
6.1582.2.2 compareCategories()	3676
6.1582.2.3 compareInfosCategories()	3677
6.1582.2.4 infosLessThan()	3677
6.1582.2.5 setDirectSourceShowfotoModel()	3677
6.1582.2.6 showfotoFilterModel()	3677
6.1582.2.7 showfotoItemInfosAdded	3677
6.1582.2.8 subSortLessThan()	3677
6.1583 ShowFoto::ShowfotoFolderViewBar Class Reference	3678
6.1584 ShowFoto::ShowfotoFolderViewBookmarkDlg Class Reference	3680
6.1585 ShowFoto::ShowfotoFolderViewBookmarkItem Class Reference	3681
6.1586 ShowFoto::ShowfotoFolderViewBookmarkList Class Reference	3682
6.1587 ShowFoto::ShowfotoFolderViewBookmarks Class Reference	3683
6.1588 ShowFoto::ShowfotoFolderViewList Class Reference	3684
6.1588.1 Member Enumeration Documentation	3684
6.1588.1.1 FolderViewRole	3684

6.1589 ShowFoto::ShowfotoFolderViewModel Class Reference	3685
6.1589.1 Member Function Documentation	3685
6.1589.1.1 currentFilePath()	3685
6.1590 ShowFoto::ShowfotoFolderViewSideBar Class Reference	3686
6.1590.1 Member Function Documentation	3688
6.1590.1.1 doLoadState()	3688
6.1590.1.2 doSaveState()	3688
6.1591 ShowFoto::ShowfotoFolderViewToolTip Class Reference	3689
6.1592 ShowFoto::ShowfotoFolderViewUndo Class Reference	3690
6.1593 ShowFoto::ShowfotoInfolface Class Reference	3691
6.1593.1 Member Function Documentation	3693
6.1593.1.1 openSetupPage()	3693
6.1594 ShowFoto::ShowfotoItemInfo Class Reference	3693
6.1594.1 Member Function Documentation	3694
6.1594.1.1 isNull()	3694
6.1594.1.2 operator==( )	3694
6.1594.2 Member Data Documentation	3694
6.1594.2.1 size	3694
6.1595 ShowFoto::ShowfotoItemModel Class Reference	3695
6.1595.1 Member Enumeration Documentation	3697
6.1595.1.1 ShowfotoItemModelRoles	3697
6.1595.2 Member Function Documentation	3698
6.1595.2.1 addShowfotoItemInfoSynchronously()	3698
6.1595.2.2 allRefreshingFinished	3698
6.1595.2.3 clearShowfotoItemInfos()	3698
6.1595.2.4 indexForShowfotoItemInfo()	3698
6.1595.2.5 indexForUrl()	3698
6.1595.2.6 itemInfosAboutToBeAdded	3699
6.1595.2.7 itemInfosAboutToBeRemoved	3699
6.1595.2.8 itemInfosAdded	3699
6.1595.2.9 itemInfosRemoved	3699
6.1595.2.10 preprocess	3699
6.1595.2.11 readyForIncrementalRefresh	3699
6.1595.2.12 removeIndex()	3700
6.1595.2.13 requestIncrementalRefresh()	3700
6.1595.2.14 retrieveShowfotoItemInfo()	3700
6.1595.2.15 rowCount()	3700
6.1595.2.16 setKeepsFileUrlCache()	3700
6.1595.2.17 setSendRemovalSignals()	3700
6.1595.2.18 setShowfotoItemInfos()	3700
6.1595.2.19 showfotoItemInfo() [1/2]	3701
6.1595.2.20 showfotoItemInfo() [2/2]	3701

---

6.1595.2.21 showfotoItemInfosAboutToBeRemoved()	3701
6.1595.2.22 showfotoItemInfosCleared()	3701
6.1595.2.23 startIncrementalRefresh()	3701
6.1596 ShowFoto::ShowfotoItemSortSettings Class Reference	3701
6.1596.1 Member Enumeration Documentation	3702
6.1596.1.1 SortOrder	3702
6.1596.2 Member Function Documentation	3703
6.1596.2.1 compare()	3703
6.1596.2.2 compareByOrder()	3703
6.1596.2.3 compareCategories()	3703
6.1596.2.4 compareValue()	3703
6.1596.2.5 lessThan() [1/2]	3703
6.1596.2.6 lessThan() [2/2]	3704
6.1596.2.7 lessThanByOrder()	3704
6.1596.2.8 naturalCompare()	3704
6.1596.3 Member Data Documentation	3704
6.1596.3.1 currentCategorizationSortOrder	3704
6.1597 ShowFoto::ShowfotoItemViewDelegate Class Reference	3705
6.1597.1 Member Function Documentation	3708
6.1597.1.1 acceptsActivation()	3708
6.1597.1.2 acceptsToolTip()	3708
6.1597.1.3 asDelegate()	3708
6.1597.1.4 gridSize()	3708
6.1597.1.5 imageInformationRect()	3708
6.1597.1.6 mouseMoved()	3709
6.1597.1.7 pixmapRect()	3709
6.1597.1.8 setDefaultViewOptions()	3709
6.1597.1.9 setSpacing()	3709
6.1597.1.10 setThumbnailSize()	3709
6.1598 ShowFoto::ShowfotoItemViewDelegatePrivate Class Reference	3710
6.1598.1 Member Function Documentation	3711
6.1598.1.1 clearRects()	3711
6.1599 ShowFoto::ShowfotoKineticScroller Class Reference	3712
6.1599.1 Detailed Description	3712
6.1599.2 Member Function Documentation	3712
6.1599.2.1 enableKineticScrollFor()	3712
6.1600 ShowFoto::ShowfotoNormalDelegate Class Reference	3713
6.1600.1 Member Function Documentation	3717
6.1600.1.1 updateRects()	3717
6.1601 ShowFoto::ShowfotoNormalDelegatePrivate Class Reference	3718
6.1602 ShowFoto::ShowfotoSettings Class Reference	3720
6.1603 ShowFoto::ShowfotoSetup Class Reference	3722

---

6.1603.1 Member Function Documentation . . . . .	3724
6.1603.1.1 execSinglePage() . . . . .	3724
6.1604 ShowFoto::ShowfotoSetupMetadata Class Reference . . . . .	3725
6.1605 ShowFoto::ShowfotoSetupMisc Class Reference . . . . .	3726
6.1606 ShowFoto::ShowfotoSetupPlugins Class Reference . . . . .	3727
6.1607 ShowFoto::ShowfotoSetupRaw Class Reference . . . . .	3728
6.1608 ShowFoto::ShowfotoSetupToolTip Class Reference . . . . .	3729
6.1609 ShowFoto::ShowfotoSortFilterModel Class Reference . . . . .	3730
6.1609.1 Member Function Documentation . . . . .	3732
6.1609.1.1 mapToSourceShowfotoModel() . . . . .	3732
6.1609.1.2 setDirectSourceShowfotoModel() . . . . .	3732
6.1609.1.3 showfotoFilterModel() . . . . .	3732
6.1609.1.4 showfotoItemInfosSorted() . . . . .	3732
6.1610 ShowFoto::ShowfotoStackViewFavoriteItem Class Reference . . . . .	3733
6.1610.1 Member Enumeration Documentation . . . . .	3734
6.1610.1.1 FavoriteType . . . . .	3734
6.1610.2 Member Function Documentation . . . . .	3734
6.1610.2.1 hierarchyFromParent() . . . . .	3734
6.1610.2.2 urlsToPaths() . . . . .	3734
6.1611 ShowFoto::ShowfotoStackViewFavoriteItemDlg Class Reference . . . . .	3735
6.1612 ShowFoto::ShowfotoStackViewFavoriteList Class Reference . . . . .	3736
6.1612.1 Member Function Documentation . . . . .	3737
6.1612.1.1 filter() . . . . .	3737
6.1612.1.2 setFilter() . . . . .	3737
6.1612.1.3 signalSearchResult . . . . .	3737
6.1613 ShowFoto::ShowfotoStackViewFavorites Class Reference . . . . .	3738
6.1614 ShowFoto::ShowfotoStackViewItem Class Reference . . . . .	3739
6.1615 ShowFoto::ShowfotoStackViewList Class Reference . . . . .	3740
6.1615.1 Member Enumeration Documentation . . . . .	3741
6.1615.1.1 StackViewRole . . . . .	3741
6.1616 ShowFoto::ShowfotoStackViewSideBar Class Reference . . . . .	3742
6.1616.1 Member Function Documentation . . . . .	3743
6.1616.1.1 doLoadState() . . . . .	3743
6.1616.1.2 doSaveState() . . . . .	3744
6.1617 ShowFoto::ShowfotoStackViewToolTip Class Reference . . . . .	3744
6.1618 ShowFoto::ShowfotoThumbnailBar Class Reference . . . . .	3746
6.1618.1 Member Function Documentation . . . . .	3751
6.1618.1.1 setModelsFiltered() . . . . .	3751
6.1618.1.2 setScrollBarPolicy() . . . . .	3751
6.1619 ShowFoto::ShowfotoThumbnailDelegate Class Reference . . . . .	3752
6.1619.1 Member Function Documentation . . . . .	3756
6.1619.1.1 acceptsActivation() . . . . .	3756

---

6.1619.1.2 maximumSize()	3756
6.1619.1.3 setDefaultViewOptions()	3756
6.1619.1.4 updateContentWidth()	3756
6.1619.1.5 updateRects()	3756
6.1620 ShowFoto::ShowfotoThumbnailDelegatePrivate Class Reference	3757
6.1621 ShowFoto::ShowfotoThumbnailModel Class Reference	3759
6.1621.1 Constructor & Destructor Documentation	3762
6.1621.1.1 ShowfotoThumbnailModel()	3762
6.1621.2 Member Function Documentation	3763
6.1621.2.1 data()	3763
6.1621.2.2 setData()	3763
6.1621.2.3 setEmitDataChanged()	3763
6.1621.2.4 setPreloadThumbnails()	3763
6.1621.2.5 setPreloadThumbnailSize()	3763
6.1621.2.6 setThumbnailLoadThread()	3763
6.1621.2.7 setThumbnailSize()	3764
6.1621.2.8 showfotoItemInfosCleared()	3764
<b>Index</b>	<b>3765</b>



# Chapter 1

## digikam project API reference.

digikam is an advanced open-source digital photo management application that runs on Linux, Windows, and macOS.

digikam is an advanced open-source digital photo management application that runs on Linux, Windows, and macOS.

### Author

(c) 2001-2025 digiKam team.

## 1.1 Source Code Directories

digikam is split into a number of components, each ones located to a dedicated directory. The main namespace is [Digikam](#) for the digiKam application and all sub components. A second namespace is ShowFoto for the stand alone version of digiKam image editor.

See below the complete list of directories used by the project:

SOURCE TREE-VIEW	DETAILS
. AUTHORS	List of developers and contributors to the project
. bootstrap.api	Script to build API documentation (HTML + PDF)
. bootstrap.linux	Configuration script to compile under Linux
. bootstrap.local	Configuration script to compile a local version under Linux
. bootstrap.macports	Configuration script to compile under macOS with Macports
. bootstrap.homebrew	Configuration script to compile under macOS with HomeBrew
. bootstrap.vcpkg	Configuration script to compile under Windows with VPCKG
. bootstrap.tarball	Script to build the release tarball
. build	Temporary directory created by bootstrap script to host compiled files
. ChangeLog	Note about how to list source code changes since the project origin
. CMakeLists.txt	Main Cmake script including lead compilation rules for the project
. COPYING	Link to main project license
. LICENSES	All licenses used in the project
. Mainpage.dox	API documentation main page based on Doxygen
. Messages.sh	Script to extract strings for translators
. NEWS	Notice to resume all project changes done at release time

SOURCE TREE-VIEW	DETAILS
. README.DEVEL	Read me file for developers
. README.md	First start helper documentation
. README.BUNDLES	Read me for Linux, macOS, and Windows bundles support
. build	Directory to store compiled files and binary targets
. core	All source code are hosted in this directory
.. app	Lead application component
... date	All date relevant views
... dragdrop	Drag and drop helper classes
... filters	Tags filter widgets
... items	Item management classes
... delegate	Item view delegate
... overlays	Item overlays
... thumbbar	Item thumbbar widget
... utils	Item utility classes
... views	Item view classes
... main	Main digiKam application
... utils	Generic utility classes
... views	Views classes
... preview	Item preview classes
... sidebar	Left sidebar contents
... stack	Stacked-view show in central place of main digiKam window
... tableview	Table-view classes
... utils	View utility classes
.. cmake	Extra Cmake scripts will be hosted here
... modules	Cmake scripts to find extra dependencies
... templates	Cmake template files used at configuration time
.. data	Application data files will be hosted here
... about	Welcome page files (HTML + CSS)
... colorschemes	GUI Color scheme files
... database	Database XML configuration files
... facesengine	Face detection and recognition data files
... filters	Image filters data files
... geolocation	Geolocation tool data files
... hotplug	Hotplug Linux integration files
... htmlgallery	HTML gallery tool data files
... icons	Application icons
... metadata	Metadata tool data files
... pics	Application pictures
... printcreator	Print Creator tool data files
... profiles	Basis open source ICC color profiles
... scripts	Miscs maintenance scripts
.. dplugins	All digiKam plugins will be hosted in this directory
... bqm	All Batch Queue Manager plugins
... colors	All color adjustments plugins
... convert	All file convert plugins
... custom	All user-custom processing plugins
... decorate	All decorate item plugins
... enhance	All enhance item plugins
... filters	All filter item plugins

SOURCE TREE-VIEW	DETAILS
... metadata	All metadata edit plugins
... transform	All transform item plugins
... editor	All Image Editor plugins
... colors	All color adjustments plugins
... decorate	All decorate item plugins
... enhance	All enhance item plugins
... file	All file processing plugins
... filters	All filter item plugins
... transform	All transform item plugins
... generic	All generic plugins
... import	Tools to import items
... metadata	Plugins to change items metadata
... tools	Plugins hosted in Tools main menu
... view	Plugins to display items
... webservices	All plugins to import and export items to remote web-services
... rawimport	All Raw import plugins
... dimg	All DImg image loader plugins
.. libs	digiKam core sub-components (few are shared with Showfoto)
... album	All classes use to manage digiKam albums operations and properties
... database	All low level database interface is here
... collection	All classes relevant of collections management
... coredb	The core database interface used to host all image properties
... dbjobs	All database multi-threaded jobs
... engine	The low level database engine classes
... haar	The similarity low level algorithms to compute image finger-prints
... history	The item history classes for the database
... item	The database item classes, including containers, lister, and scanner
... models	The database model classes
... server	The Mysql internal server
... similaritydb	The similarity database
... tags	The database tags management classes
... thumbsdb	The thumbnails database
... utils	Miscs tools and widgets used with database
... dialogs	Common dialogs
... dimg	The Qt digiKam image data container support ICC and 16 bits color depth
... filters	All image filters will be hosted here. All support 16 bits color depth
... auto	Auto colors correction filters
... bcg	Brightness-Contrast-Gamma filter
... bw	Black and White image converter, including infrared filter
... cb	Colors balance filter
... curves	Colors curves filter
... decorate	Decorate filters
... film	Analog film emulation filters
... fx	Special effect filters
... greycstoration	Cimg based restoration filter
... hsl	Hue-Saturation-Lightness filter
... icc	Icc color profile filters
... imgqsort	The image quality sort algorithms
... lc	Local contrast filter (pseudo HDR)

SOURCE TREE-VIEW	DETAILS
. . . . . lens	Lens corrections filters, including Qt Lensfun interface
. . . . . levels	Color levels filter
. . . . . nr	Wavelets noise reduction filter
. . . . . redeye	Red-eyes parser and fixer
. . . . . sharp	Image sharp filter, including Unsharped-mask and Refocus
. . . . . transform	All image transformation filters
. . . . . wb	White balance filter
. . . . imagehistory	Image history interface for image container
. . . . loaders	All DImg image loaders interface
. . . . metadataengine	The metadata wrapper based on Exiv2 for image and FFMpeg for video
. . . . dngwriter	Qt classes to convert RAW files to DNG format
. . . . extra	DNG and XMP sdks from Adobe
. . . . dplugins	All shared dplugins classes are hosted here
. . . . core	Low level classes for plugins definitions
. . . . iface	Low level classes for host interface definitions
. . . . setup	Classes to setup plugins in configuration panel
. . . . webservices	Common classes for Webservices tools
. . . . widgets	Common widget sfor plugins
. . . . dtrash	digiKam trash manager full independent of desktop trash
. . . . facesengine	Face detection and recognition engine + Faces database implementations
. . . . alignment-congealing	Face alignment based on congealing method
. . . . alignment-flandmark	Face alignment based on flandmark method
. . . . detection	Face detection algorithms
. . . . dnnface	Deep-learning face algorithms
. . . . facedb	Faces database classes
. . . . opencv3-face	OpenCV version 3 face management classes
. . . . preprocessing-tantriggs	Face pre-processing based on tantriggs method
. . . . recognition-dlib-dnn	Deep-learning faces recognition module
. . . . recognition-opencv-eigenfaces	Eigen faces recognition module
. . . . recognition-opencv-fisherfaces	Fisher faces recognition module
. . . . recognition-opencv-lbph	LBPH bases faces recognition module
. . . . shape-predictor	Shape predictor algorithms
. . . . fileactionmanager	Classes to connect database and metadata actions to file operations
. . . . filters	Widgets to filter items by metadata properties
. . . . imageproperties	All widgets used in right side-bar from all main views
. . . . iojobs	Multithreaded jobs manager used with files operations
. . . . jpegutils	Utilities to process JPEG files
. . . . libjpeg	JPEG loss-less transform private implementations from libjpeg
. . . . kmemoryinfo	Qt backend to analyze system memory information
. . . . models	Qt models used with item views
. . . . notificationmanager	Multi-desktop notifications wrapper
. . . . pgfutils	Qt Classes to work with PGF image format
. . . . progressmanager	Multi-level operations progress widget
. . . . rawengine	Qt classes to work with libraw decoder
. . . . libraw	Internal Libraw sdk
. . . . settings	digiKam settings manager
. . . . tags	Classes to play with tags
. . . . tagsmanager	Tags manager view
. . . . template	Metadata template support

SOURCE TREE-VIEW	DETAILS
. . . threadimageio	Classes to process thumbs and preview extraction including video support
. . . threads	Classes to manage and chain threads using multi-core
. . . timeadjust	Common classes time adjustments tools
. . . transitionmngr	Frames transitions manager
. . . versionmanager	Classes to manage versioning operations
. . . video	Classes to play with video contents
. . . widgets	To host plenty of widgets used everywhere
. . . . colors	Colors relevant views
. . . . combo	Combo-box helper classes
. . . . common	Uncategorized widgets
. . . . files	File operation classes
. . . . fonts	Font management classes
. . . . graphicsview	Graphics-view implementation (model-view)
. . . . iccprofiles	ICC color profiles widgets
. . . . imagehistory	Image history widgets
. . . . itemview	Item-view implementations (model-view)
. . . . layout	Layout helper classes
. . . . mainview	Common top-level view implementations
. . . . metadata	Metadata widgets
. . . . range	Range helper classes
. . showfoto	Stand alone image editor
. . . main	Main Showfoto application
. . . setup	Showfoto Setup views
. . . thumbbar	Showfoto thumb-bar views
. . tests	Unit tests
. . utilities	digiKam utilities and advanced tools (few are shared with showfoto)
. . . advancedrename	Advance rename tool
. . . extrasupport	Extra desktop features support as Baloo search engine
. . . facemanagement	Face management classes and tools
. . . firstrun	First-run assistant to configure lead digiKam settings
. . . fuzzysearch	Similarity search tools
. . . geolocation	All geo-location tools are located here
. . . . editor	Tool to edit items geo-location
. . . . geoiface	All shared classes used by geo-location tools
. . . . geomapwrapper	Legacy helper classes for geo-location support
. . . . mapsearches	Tool to perform map searches
. . . imageeditor	The famous digiKam image editor, a lots of classes shared with Showfoto
. . . . core	Core implementation including canvas and tools interface
. . . . dialogs	Image editor dialogs
. . . . editor	The core image editors classes
. . . . main	The main digiKam image editor view, not shared with Showfoto
. . . . widgets	All common widgets
. . . import	The import tools, including USB Mass Storage and Gphoto2 support
. . . . backend	Camera backends
. . . . dialogs	Import tools dialogs
. . . . items	Import item classes
. . . . main	Import tool main view
. . . . models	Import model classes
. . . . views	Import view classes

SOURCE TREE-VIEW	DETAILS
. . . . widgets	Import common widgets
. . . lighttable	The Light-table tool to compare images side by side
. . . maintenance	The digiKam tool to maintain the database contents
. . . queuemanager	The famous Batch Queue Manager tool
. . . . main	The main BQM view
. . . . manager	The multi-core manager to run tools in background
. . . . tools	All BQM tools classed by functions
. . . . views	The BQM internal views
. . . searchwindow	The powerful advanced search tool
. . . setup	All digiKam setup panel, with few ones shared with Showfoto
. . . . album	Album configuration views
. . . . camera	Camera configuration views
. . . . collections	Collection configuration views
. . . . editor	Image Editor configuration views
. . . . metadata	Metadata configuration views
. . . slideshow	The simple slideshow tool
. po	Program translations
. project	Extra project parts
. . bundles	Bundles build scripts
. . . 3rdparty	External components required to build bundles
. . . CD	Continuous deployment configurations
. . . appimage	Linux AppImage
. . . homebrew	macOS package (HomeBrew version)
. . . macports	macOS package (Macports version)
. . . vcpkg	Windows installer
. . documents	Project documentations
. . reports	Static analyzers report scripts for Continuous Integration
. . scripts	3rdparty source code management scripts

## 1.2 External Dependencies

### 1.2.1 Dependencies To Checkout All Source Code

- Git <http://git-scm.com>

### 1.2.2 Dependencies To Process Translations Files (optional)

- Gettext <https://www.gnu.org/software/gettext> (including Msgfmt to compile po files to mo files)

### 1.2.3 Dependencies To Compile And Link Source Code

The full list of mandatory (X) and (optional) external dependencies required to compile and link digiKam source code is listed below.

Dependency	Requirement	Qt5 Version	Qt6 Version	External Links	Remarks	Notes
CMake	X	>= 3.16.0	>= 3.22.0	<a href="#">url</a>		
ECM	X	>= 5.55.0	>= 5.240.0	<a href="#">url</a>	Qt6 support implemented in KDE framework >= 5.91.0	
Qt::Core	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::Gui	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::Widgets	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::Network	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::NetworkAuth	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::Sql	X	>= 5.14	>= 6.4	<a href="#">url</a>	Including Qt::Sqlite and Qt::Mysql plugins	
Qt::Xml	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::Concurrent	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::PrintSupport	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::Svg	X	>= 5.14	>= 6.4	<a href="#">url</a>		
Qt::WebEngine	X	>= 5.14	>= 6.4	<a href="#">url</a>	To render web contents (ENABLE_QWEBENGINE=on)	
Qt::XmlPatterns	optional	>= 5.14	—	<a href="#">url</a>	To parse and validate Xml	Used by Rajce plugin. Module removed with Qt6.
Qt::X11Extras	optional	>= 5.14	—	<a href="#">url</a>	For color management support under Linux	Module removed with Qt6.
Qt::DBus	optional	>= 5.14	>= 6.4	<a href="#">url</a>	Optional: only for Linux Desktop	
Qt::OpenGL	optional	>= 5.14	>= 6.4	<a href="#">url</a>	For Presentation tool	
Qt::OpenGLWidgets	optional	—	>= 6.4	<a href="#">url</a>	For Presentation tool	With Qt6, OpenGL is separated in 2 modules: core and widgets.
Qt::Multimedia	optional	—	>= 6.4	<a href="#">url</a>	For Presentation tool	With Qt6, OpenGL is separated in 2 modules: core and widgets.

Dependency	Requirement	Qt5 Version	Qt6 Version	External Links	Remarks	Notes
Qt::Test	optional	>= 5.14	>= 6.4	<a href="#">url</a>	To compile test codes (BUILD_↔ TESTING=on)	
Qt::Qml	optional	>= 5.14	>= 6.4	<a href="#">url</a>	To compile test codes (BUILD_↔ TESTING=on) O2 unit tests	
Qt::WebView	optional	>= 5.14	>= 6.4	<a href="#">url</a>	To compile test codes (BUILD_↔ TESTING=on) O2 unit tests	
KF::Config	X	>= 5.95.0	>= 5.240.0	<a href="#">url</a>		
KF::XmlGui	X	>= 5.95.0	>= 5.240.0	<a href="#">url</a>		
KF::I18n	X	>= 5.95.0	>= 5.240.0	<a href="#">url</a>		
KF::↔ Window↔ System	X	>= 5.95.0	>= 5.240.0	<a href="#">url</a>		
KF::Service	X	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	TODO: make optional for Linux desktop (DFile↔ Operations)	
KF::Solid	X	>= 5.95.0	>= 5.240.0	<a href="#">url</a>		
KF::Core↔ Addons	X	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	Needs for KAbout↔ Data and KMemory↔ Info	
KF::Notify↔ Config	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	For Linux desktop application notify configuration	
KF::↔ Notifications	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	For Linux desktop notifications integrations	
KF::↔ Thread↔ Weaver	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	For panorama tool	
KF::Icon↔ Themes	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	Optional: only for Linux Desktop (KIcon↔ Dialog)	
KF::File↔ MetaData	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	Plasma desktop files indexer support	(ENABLE_↔ KFILEMETADATASUPPORT=ON Disabled by default.



Dependency	Requirement	Qt5 Version	Qt6 Version	External Links	Remarks	Notes
KF::CalendarCore	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	For calendar tool to setup special events	
KF::KIO	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	Optional: only for Linux Desktop	
KF::Sonnet	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	To perform spell-checking in text widget (aka caption)	
KF::AkonadiContact	optional	>= 5.95.0	>= 5.240.0	<a href="#">url</a>	Plasma desktop address-book support	(ENABLE_AKONADICONTACTSUPPORT= Disabled by default.
libopencv	X	>= 3.3		<a href="#">url</a>	OpenCV 4 recommended	DNN module required for face management
libtiff	X	>= 4.0		<a href="#">url</a>	For DImg TIFF image loader	
libpng	X	>= 1.6		<a href="#">url</a>	For DImg PNG image loader	
libjpeg	X	>= 8		<a href="#">url</a>	jpeglib >= 8.0 required by RawEngine for DNG support	
libboost	X	>= 1.55.0		<a href="#">url</a>	For Versioning support	
liblcms	X	>= 2.x		<a href="#">url</a>	For Color Management support	
libexpat	X	>= 2.1.0		<a href="#">url</a>	For RAW to DNG converter	
libexiv2	X	>= 0.27.0		<a href="#">url</a>	Metadata low level management.	
libjpegxl	optional	>= 0.7		<a href="#">url</a>	For DNG-Writer and RawEngine	To decode and encode DNG files
libheif	optional	>= 1.6.0		<a href="#">url</a>	For HEIF file format support.	Library must be compiled with libde265 (read) and optionally with libx265 (write).

Depen- dency	Require- ment	Qt5 Version	Qt6 Version	External Links	Remarks	Notes
libx265	optional	>= 2.2		<a href="#">url</a>	For HEIC en- coding sup- port	
libxml2	optional	>= 2.7.0		<a href="#">url</a>	For Html↔ Gallery tool	
libxslt	optional	>= 1.1.0		<a href="#">url</a>	For Html↔ Gallery tool	
Flex	optional	>= 2.5.0		<a href="#">url</a>	For Panorama tool	
Bison	optional	>= 2.5.0		<a href="#">url</a>	For Panorama tool	
libmesa	optional	>= 11.0		<a href="#">url</a>	For Presen- tation tools (Linux only)	
libksane	optional	>= 21.12.0	22.04.2	<a href="#">url</a>	Digital scan- ner support	
libjasper	optional	>= 1.900.1		<a href="#">url</a>	For JPEG- 2000 support	
libeigen3	optional	>= 3.2		<a href="#">url</a>	For Refocus tool	See if Cla- pack from OpenCV can be used instead
liblensfun	optional	>= 0.2.8		<a href="#">url</a>	For Lens↔ Correction tool	
libglib2	optional	>= 2.0.0		<a href="#">url</a>	For Liquid rescale tool	
libgphoto2	optional	>= 2.5		<a href="#">url</a>	Digital cam- era drivers support. Need libusb- 1	
libgomp	optional	>= 5.0		<a href="#">url</a>	OpenMP support for RawEngine	
libimagemagick	optional	>= 6.7.0		<a href="#">url</a>	Image↔ Magick codecs sup- port for DImg image loader	Version >= 7.0 recom- mended
libffmpeg	optional	>= 4.4.x		<a href="#">url</a>	To play video and audio (ENABLE_↔ MEDIAPLAYER=on)	libavformat, libavutil, libavcodec used to ex- tract video metadata. QtAVPlayer
libvaapi	optional	>= 2.4		<a href="#">url</a>	To play video and audio (ENABLE_↔ MEDIAPLAYER=on)	Intel Video support in QtAVPlayer

## 1.3 Get Source Code

### 1.3.1 Software Components

digikam project use a single git repository from GitLab to host whole source code base. The project page is given below:

<https://invent.kde.org/graphics/digikam>

The digikam handbook source code is hosted in a separate GitLab repository:

<https://invent.kde.org/documentation/digikam-doc>

## 1.4 Development Environment

If you are a developer with push access to the git repositories, it is strongly recommended to use the "kde:" prefix and let git use the read-only mirrors for pulling.

If you did not clone this repository from "kde:", do it again:

```
git config --global url.git://anongit.kde.org/.insteadof kde:
git config --global url.ssh://git@git.kde.org/.pushinsteadof kde:
git clone kde:digikam
```

See below an example of .gitconfig file working with a developer account:

```
[url "git://anongit.kde.org/"]
    insteadof = kde://

[url "git@git.kde.org:"]
    pushinsteadof = kde://

[url "ssh://git@git.kde.org/"]
    pushinsteadof = kde://

[alias]
    up = pull --rebase -v --stat
    ci = commit -a -v

[core]
    editor = mcedit

[user]
    name = my name
    email = my email

[push]
    default = tracking

[color]
    # turn on color
    diff = auto
    status = auto
    branch = auto
    interactive = auto
    ui = auto

[color "branch"]
    current = green bold
    local = green
    remote = red bold

[color "diff"]
    meta = yellow bold
    frag = magenta bold
    old = red bold
    new = green bold

[color "status"]
    added = green bold
    changed = yellow bold
    untracked = red

[color "sh"]
    branch = yellow
    [color "sh"]
```

## 1.5 Cmake Configuration Options

To configure the project with CMake, use dedicated "bootstrap" script for your platform where all available configuration options are present with default values.

There are two configuration sections : the top level and the core.

### 1.5.1 Top Level Configuration

- Packaging options:
  - **DIGIKAMSC\_COMPILE\_DIGIKAM** : Build digiKam core (default=ON).
  - **DIGIKAMSC\_COMPILE\_PO** : Build application translations files. (default=OFF).
- Developers only options:
  - **BUILD\_TESTING=ON** : Build tests code (default=ON).
  - **BUILD\_WITH\_QT6=ON** : Build with Qt6 framework, else Qt5 (default=OFF).

### 1.5.2 Core Configuration

- Extra feature support options:
  - **ENABLE\_KFILEMETADATASUPPORT** : Build digiKam with KDE files indexer support (default=OFF).
  - **ENABLE\_AKONADICONTACTSUPPORT** : Build digiKam with KDE Mail Contacts support (default=OFF).
  - **ENABLE\_GEOLOCATION** : Build digiKam with Geolocation support (default=ON).
  - **ENABLE\_MEDIAPLAYER** : Build digiKam with Media Player support (default=ON).
  - **ENABLE\_DBUS** : Build digiKam with DBUS support (default=ON).
  - **ENABLE\_APPSTYLES** : Build digiKam with support for changing the widget application style (default=OFF).
  - **ENABLE\_KIO** : Build digiKam with KIO support (default=ON).
- Database options
  - **ENABLE\_MYSQLSUPPORT** : Build digiKam with MySQL database support (default=ON).
  - **ENABLE\_INTERNALMYSQL** : Build digiKam with internal MySQL server executable (default=ON).
- Showfoto application options
  - **ENABLE\_SHOWFOTO** : Build Showfoto stand-alone image editor application (default=ON).
- Developers only options:
  - **ENABLE\_DIGIKAM\_MODELTEST** : Enable ModelTest on some models for debugging (default=OFF).
  - **ENABLE\_SANITIZERS** : Enable ASAN and UBSAN sanitizers when available (default=OFF).
  - **BUILD\_WITH\_CCACHE** : Use ccache to speed up compilations (default=OFF)

## 1.6 Setup Local Compilation and Run-Time

This section describes how to install digiKam from the git repository, while keeping a system-wide digiKam install.

This procedure is based on the configure script **bootstrap.local**

1. Set the root directory for your git install in bootstrap.local (DIGIKAM\_INSTALL\_PREFIX variable)
2. If you want a clean build directory, set CLEANROOT to 1
3. Type the following command in your terminal:

```
$ ./bootstrap.local # or "./bootstrap.local --eclipse" if you intend to use Eclipse
$ cd build
$ make
$ make install
$ KDESYCOCA="/your/root/directory/var/tmp/kde-$USER/ksycoca5" kbuildsycoca5
```

To run digikam, use the following commands:

```
$ export KDESYCOCA=/your/root/directory/var/tmp/kde-$USER/ksycoca5
$ export QT_PLUGIN_PATH=/your/root/directory/lib64/plugins:/your/root/directory/lib/plugins:$QT_PLUGIN_PATH
$ export XDG_DATA_DIRS=/your/root/directory/share:$XDG_DATA_DIRS
$ /your/root/directory/bin/digikam
```

The same applies for all binaries in /your/root/directory/bin/

If your shell is bash, you can edit your .bashrc file (in \$HOME) and add the following alias:

```
DIGIKAMROOT="/your/root/directory"
alias digikam-dev="KDESYCOCA=\$DIGIKAMROOT/var/tmp/kde-$USER/ksycoca5
XDG_DATA_DIRS=\$DIGIKAMROOT/share:\$XDG_DATA_DIRS
QT_PLUGIN_PATH=\$DIGIKAMROOT/lib64/plugins:\$DIGIKAMROOT/lib/plugins:\$QT_PLUGIN_PATH
\$DIGIKAMROOT/bin/digikam"
```

then you can start your newly installed digikam with

```
$ digikam-dev
```

## 1.7 Debug Traces At Run-Time

digiKam uses categorized logging at run-time. By default, all debug messages are printed on the console. To disable output, you can either fine-grained control by using one or more logging categories listed below.

Note: under Windows, to catch all debug messages you need to install an extra Microsoft application named DebugView available at this url: <http://technet.microsoft.com/en-us/sysinternals/bb896647>. ↩  
aspix

### 1.7.1 Logging Using an Environment Variable

You can set the environment variable **QT\_LOGGING\_RULES**. Rules are divided by semicolons.

E.g. you can start digiKam like this on the command line with thumbnails and core database messages disabled:

```
export QT_LOGGING_RULES='digiKam.thumbsdb=false;digiKam.coredb=false'
digiKam
```

## 1.7.2 Logging Categories in digiKam

All logging categories are listed in

```

/* =====
 *
 * This file is a part of digiKam project
 * https://www.digikam.org
 *
 * Date      : 2014-09-08
 * Description : digiKam debug spaces
 *
 * SPDX-FileCopyrightText: 2014      by Laurent Montel <montel at kde dot org>
 * SPDX-FileCopyrightText: 2015      by Mohamed_Anwer <m_dot_anwer at gmx dot com>
 * SPDX-FileCopyrightText: 2014-2025 by Gilles Caulier <caulier dot gilles at gmail dot com>
 *
 * SPDX-License-Identifier: GPL-2.0-or-later
 *
 * ===== */

#include "digikam_debug.h"

// Local includes

#include "digikam_config.h"

Q_LOGGING_CATEGORY(DIGIKAM_GENERAL_LOG,          "digikam.general",          QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_WIDGETS_LOG,         "digikam.widgets",         QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_IOJOB_LOG,           "digikam.iojob",           QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_SHOWFOTO_LOG,        "digikam.showfoto",        QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_WEBSERVICES_LOG,     "digikam.webservices",     QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DATABASESERVER_LOG,  "digikam.databaseserver",  QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_IMPORTUI_LOG,        "digikam.import",          QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_METAENGINE_LOG,      "digikam.metaengine",      QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_RAWENGINE_LOG,       "digikam.rawengine",       QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_FACEENGINE_LOG,      "digikam.facesengine",     QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_AUTOTAGSENGINE_LOG,  "digikam.autotagsengine",  QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_GEOIFACE_LOG,        "digikam.geoface",         QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_GEOENGINE_LOG,       "digikam.geocore",         QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DNNMODELNGR_LOG,     "digikam.dnnmodelmanager", QtInfoMsg)

Q_LOGGING_CATEGORY(DIGIKAM_TESTS_LOG,           "digikam.tests",           QtInfoMsg)

Q_LOGGING_CATEGORY(DIGIKAM_DPLUGIN_RAWIMPORT_LOG, "digikam.dplugin.rawimport", QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DPLUGIN_GENERIC_LOG,  "digikam.dplugin.generic",  QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DPLUGIN_EDITOR_LOG,   "digikam.dplugin.editor",   QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DPLUGIN_BQM_LOG,      "digikam.dplugin.bqm",      QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DPLUGIN_LOG,          "digikam.dplugin",         QtInfoMsg)

Q_LOGGING_CATEGORY(DIGIKAM_DATABASE_LOG,         "digikam.database",        QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DBENGINE_LOG,         "digikam.dbengine",        QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DBJOB_LOG,            "digikam.dbjob",           QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_COREDB_LOG,          "digikam.coredb",          QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_FACEDB_LOG,          "digikam.facedb",          QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_THUMBSDB_LOG,        "digikam.thumbsdb",        QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_SIMILARITYDB_LOG,     "digikam.similaritydb",    QtInfoMsg)

// NOTE: per default only warnings and more severe messages are logged for other than general category

Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG,            "digikam.dimg",            QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_JPEG,       "digikam.dimg.jpeg",       QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_JP2K,       "digikam.dimg.jp2k",       QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_PGF,        "digikam.dimg.pgf",        QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_PNG,        "digikam.dimg.png",        QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_PPM,        "digikam.dimg.ppm",        QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_TIFF,       "digikam.dimg.tiff",       QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_RAW,        "digikam.dimg.raw",        QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_QIMAGE,     "digikam.dimg.qimage",     QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_HEIF,       "digikam.dimg.heif",       QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_DIMG_LOG_MAGICK,     "digikam.dimg.magick",     QtWarningMsg)

Q_LOGGING_CATEGORY(DIGIKAM_MEDIASRV_LOG,         "digikam.mediaserver",     QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_MEDIASRV_LOG_INFO,   "digikam.mediaserver.info", QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_MEDIASRV_LOG_DEBUG,  "digikam.mediaserver.debug", QtInfoMsg)
Q_LOGGING_CATEGORY(DIGIKAM_MEDIASRV_LOG_WARN,   "digikam.mediaserver.warn", QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_MEDIASRV_LOG_CRITICAL, "digikam.mediaserver.critical", QtWarningMsg)
Q_LOGGING_CATEGORY(DIGIKAM_MEDIASRV_LOG_FATAL,  "digikam.mediaserver.fatal", QtCriticalMsg)

void digikamSetDebugFilterRules(bool on)
{
    if (on)
    {
        QLoggingCategory::setFilterRules(QLatin1String("digikam.*=true\n"
            "digikam.dimg.jpeg=false\n"
            "digikam.dimg.jp2k=false\n"
            "digikam.dimg.pgf=false\n"
            "digikam.dimg.png=false\n"
            "digikam.dimg.ppm=false\n"
            "digikam.dimg.tiff=false\n"
            "digikam.dimg.raw=false\n"
            "digikam.dimg.qimage=false\n"
            "digikam.dimg.heif=false\n"
            "digikam.dimg.magick=false\n"
            "digikam.geocore=false"));
        // to much verbose
    }
    at the console
}

```

```
}
```

source code.

### 1.7.3 Further Reading

For more details see the Qt framework documentation about logging categories available at this url: <https://doc.qt.io/qt-5/qloggingcategory.html#details>

## 1.8 Cmake compilation rules

### 1.8.1 Introduction

The whole project is written mostly in C++/Qt and the Cmake framework is used to compile under Linux, macOS, and Windows. The Cmake rules have been configured to reduce the linking overhead and improve CPU utilization with modular design.

Independent Cmake configuration is presents in following folders:

- root source dir
- core
- doc

The Cmake rules will build the following targets:

- digikamcore shared lib
- digikamdatabase shared lib
- digikamgui shared lib
- digikam executable
- showfoto executable
- plugin shared libraries (dplugins)
- various test executables - if testing is enabled
- various unit-tests - if testing is enabled

Each of them depend on various sources which must be compiled before. A complete description of source code direction is given to the sourcedirs section.

## 1.8.2 CMake Implementation Details

### 1.8.2.1 Include Directories

Local include directories are all managed by this snippet of code:

```
set(DK_INCLUDES_ALL "")
HEADER_DIRECTORIES(DK_LOCAL_INCLUDES_RAW)
```

The libjpeg- folders are all included, so we need to delete them all and include the correct one only:

```
# This macro will set all paths which do not contain libjpeg-
# We will add later the directory we need
```

```
FOREACH(var ${DK_LOCAL_INCLUDES_RAW})
    STRING(REGEX MATCH "libjpeg-" item ${var})
    IF(item STREQUAL "")
        LIST(APPEND DK_LOCAL_INCLUDES ${var})
    ENDF (item)
ENDFOREACH(var)

set(DK_LOCAL_INCLUDES ${DK_LOCAL_INCLUDES}
    libs/jpegutils/${DIGIKAM_LIBJPEG_DIR})

include_directories(${DK_LOCAL_INCLUDES})
```

There is no need for manual intervention to add new includes, even if you add a new folder, just keep in mind to use:

```
#include "tagmngnrlstitem.h"
```

instead of :

```
#include "models/tagmngnrlstitem.h"
```

### 1.8.2.2 Shared Libraries

To avoid linking overhead and make a better use of sources there are some dynamic libs as these one:

- digikamcore : core components used by almost all executables as digiKam and Showfoto.
- digikamdatabase : database components, also used together with digikamcore but only for digiKam

Please add sources to digikam core or digikam database only if they don't depend on any big component from digikam main executable. These two shared libs must be kept small because they link in a lot of places

### 1.8.2.3 Static Libraries

Currently cmake configuration features a lots of shared libraries as:

- metadataedit
- geolocationedit
- digikamfaceengine

This libraries are linked in digikam main executable and some tests tools.

Avoid making static libraries if possible, and use OBJECT libraries instead. Only make STATIC libraries which does not depend on other digikam code. Also make sure you put the PRIVATE parameter when setting the target\_link\_libraries.

```
target_link_libraries(digikamcore
    PRIVATE
    Qt{QT_VERSION_MAJOR}::Core
    Qt{QT_VERSION_MAJOR}::Gui
    Qt{QT_VERSION_MAJOR}::Widgets
)
```



### 1.8.2.4 Object Libraries

While static libraries are still collection of objects, CMake offer a better approach by allowing to specify an OBJECT library:

```
set(libslideshow_SRCS
  slidetoolbar.cpp
  slideosd.cpp
  slideproperties.cpp
  slideimage.cpp
  slideerror.cpp
  slideend.cpp
  slideshow.cpp
  slidehelp.cpp
  slideshowsettings.cpp
)

add_library(slideshow_src OBJECT ${libslideshow_SRCS})
```

OBJECT library is a cmake internal implementation feature and allow to easily manage sources. Here is an example of how to make a shared lib using OBJECT libraries:

```
add_library(digikamcore
  SHARED
  ${TARGET_OBJECTS:slideshow_src} # the lib we made few lines above
  ${TARGET_OBJECTS:digikamdatabasecore_src}
  ${TARGET_OBJECTS:dimg_src}
  ....
)
```

## 1.9 Contribute To The Code

This section's purpose is to guide contributors and developers to help on the digiKam project.

### 1.9.1 Starting With Open-Source

Before to contribute to digiKam project, please take a look to this link which provide 10 golden rules for starting with open source project:

[http://schlitt.info/opensource/blog/0541\\_10\\_golden\\_rules\\_for\\_starting\\_with\\_open\\_source.html](http://schlitt.info/opensource/blog/0541_10_golden_rules_for_starting_with_open_source.html)

### 1.9.2 Source Code Formatting

Adhere to this style guide strictly while adding new code to digiKam or working on existing code.

#### 1.9.2.1 Indentation length

Indent with 4 spaces exactly.

For example:

```
void function()
{
  ....int a; // 4 spaces from beginning
  ....for (int i = 0 ; i < 10 ; ++i) // 4 spaces from beginning
  ....{ // 4 spaces from beginning
  .....a = i; // 4 spaces from previous indent block
```

Emacs by default will indent to 4 spaces vim users add this to you .vimrc set tabstop=4

### 1.9.2.2 Tabs vs Spaces

Absolutely no tabs. Use a sensible editor which will convert tabs to spaces. This will reduce unnecessary changes in your git commits.

Emacs by default will convert tab to spaces. For vim users, add this to your .vimrc set expandtab

### 1.9.2.3 Line length

Line length should never exceed 80 chars (unless really necessary - these cases are rare). Having long lines greatly reduces readability of code

### 1.9.2.4 Bracketing

In all cases, {} brackets should start on a newline and should be aligned with previous line (follow the indentation spaces). For example.

```
class A
{ //new line
...

for (int i = 0 ; i < 10 ; ++i)
{ //new line

if (a == foobar)
{ //new line
...
}
else
{ // new line
..
}
```

### 1.9.2.5 Positioning of Access modifiers

public, private, protected, public slots, ... should be aligned to the beginning of the line with no margin

```
class A
{
public: // aligned to left
...
private Q_SLOTS: // aligned to left
```

Follow a consistent order in defining these attributes. The recommended order is public, protected (functions), private (functions), signals, public slots, protected slots, private slots, private (variables)

## 1.9.3 Class, file and Variable names

### 1.9.3.1 Class and filenames

- filenames should always be in lower-case
- class names should match the filenames. Capitalize the first letter and other letters logically to improve readability

### 1.9.3.2 Protected Member variables

- protected member variable names should always be of the form `m_varName`.
- Capitalize logically so that it becomes easy to read it. Do not capitalize the first letter after `_` (Use `m_varName` not `m_VarName`)
- variable names should be indicative of their functionality and also of the type they belong too if they are instances of qt widgets. For example, `QCheckBox* m_autoRotateCheckBox`;

### 1.9.3.3 Non-Member variables

- non-member variables should follow the same naming convention as the member variables, except for the leading `m_`

### 1.9.3.4 Private Member variables

- private member variables must be stored in a `d` private container to reduce compilation time and improve binary compatibility between digiKam components. See more information how to use a 'd' private class at this url:

[https://community.kde.org/Policies/Library\\_Code\\_Policy](https://community.kde.org/Policies/Library_Code_Policy)

## 1.9.4 Comments and Whitespace

Use whitespaces liberally to improve readability. Add blank lines between logical sections of the code.

Comment as much as possible. Position comments at the beginning of the section/line you want to comment, NEVER at the end of the line, excepted for special cases for ex to describe enum values.

```
// put your comments here
a = (b == foobar) ? 1 : -1;

a = (b == foobar) ? 1 : -1; // you are asking for trouble by putting comments here
```

## 1.9.5 Header Files

- Add copyright to top of every file. Use the same header than others digiKam source code.
- Add double inclusion protection

```
#pragma once

class AnotherNiceClass
{
...
}
```

- Use forward declarations as much as possible.

```
class QFileInfo;

class A
{
...QFileInfo* m_fileInfo = nullptr;
```

## 1.9.6 Automatic source code formatting

The above coding style guidelines can be automatically applied with `astyle` (<http://astyle.sourceforge.net/>).

Run it in the directory where the files are located that should be formatted.

To apply the coding guidelines with `astyle` is to use the `fileformatter.py` script in `project/scripts` directory. This script will also clean up the source tree and remove backup files that had been created by `astyle`, if the appropriate command line argument is given.

To handle the command easier, create a bash function in `~/.bashrc`, e.g.

```
dkfrmcode()
{
    astyle --style=allman \
          --indent=spaces=4 \
          --convert-tabs \
          --indent-switches \
          --break-blocks \
          --break-closing-brackets \
          --pad-header \
          --align-pointer=type \
          --indent-coll-comments \
          --add-brackets \
          --min-conditional-indent=0 \
          `find $1 -type f -name '*.cpp'` `find $1 -type f -name '*.c'` `find $1 -type f -name '*.h'`
}

```

You can pass a parameter to the function, in this case the first parameter is the directory, where files should be formatted.

Examples:

1. Run `astyle` in the current directory
 

```
$> dkfrmcode
```
2. Run `astyle` in a different directory
 

```
$> dkfrmcode /home/user/code/git/digikam/
```

## 1.9.7 General recommendations

Please take a look into this contrib page tips before to write code/patches for digiKam project : <http://techbase.kde.org/Contribute>

Use the same `.cpp/.h` header than the rest of digiKam project.

Use a decent editor which does auto-indentation/syntax-highlighting for you, as Kate or QtCreator

There are excellent initializer scripts in the `kdesdk` package for `xemacs` and `vim` which can substantially increase your productivity.

Just to give a taste of what i can do with `emacs` (and `kdesdk`):

automatically insert copyright (and `ifdefs`) in new files. insertion of class function definitions for declared class functions in header with one keystroke switch between header and declaration files with one keystroke go to corresponding definition/declaration with one keystroke tab completion of variable/function names already declared.

## 1.9.8 GDB Backtrace

If you found a context to crash digiKam, you can provide a backtrace using GDB debugger. digiKam need to be compiled with all debug info else the backtrace will not suitable. There is a configure option for that:

```
$> cmake . -DCMAKE_BUILD_TYPE=debugfull
$> make
$> su
$> make install/fast
```

To make a backtrace with GDB use following command:

```
$ gdb digikam
> catch throw
> run
> ...
> _crash here_
> ...
> bt
> _the backtrace is here_
> quit
```

Post this backtrace at the right place (Bugzilla or development mailing list) for investigation by developers.

For Windows users, take a look on this tutorial :

[http://techbase.kde.org/Development/Tutorials/Debugging/Debugging\\_on\\_MS\\_Windows](http://techbase.kde.org/Development/Tutorials/Debugging/Debugging_on_MS_Windows)

## 1.9.9 Memory Leak

To check any memory leak problem in digiKam, valgrind is your friend ( <http://valgrind.org>) Try this command line to use with valgrind :

```
valgrind --tool=memcheck --leak-check=full --error-limit=no --suppressions=project/reports/digikam.supp digikam
```

NOTE: digikam.supp file is available in digikam/project sub-folder.

## 1.9.10 Profiling With Cachegrind

Valgrind also includes a tool to find out in which parts of your code time is spent.

```
valgrind --tool=callgrind digikam
```

Profiling can be disabled at startup to limit the output to the code you are interested in. Start with

```
valgrind --tool=callgrind --instr-atstart=no digikam
```

and prepare the situation you want to profile. Then, in another console, start profiling with "callgrind\_control -i on" and, after the situation has passed, request a profile dump with "callgrind\_control -d". The resulting callgrind.out files need to be viewed with the kcachegrind program, e.g.:

```
kcachegrind callgrind.out.16693.1
```

## 1.9.11 Unit Testing / Automated Testing

Unit Testing is great way to ensure that software units (in OOP this normally means classes) work as expected. Wikipedia gives a good introduction to Unit Testing:

[http://en.wikipedia.org/wiki/Unit\\_testing](http://en.wikipedia.org/wiki/Unit_testing)

It is also worth to follow most of QTest API rules with digiKam:

<http://doc.qt.io/qt-5/qtest-tutorial.html>

The digiKam test suite is located under tests and will be compiled if BUILD\_TESTING is turned ON at cmake configuration time. After compiling the source code the tests can be executed via

```
make test
```

The console output while running the tests is stored in Testing/Temporary/LastTest.log in the CMake binary dir.

All tests are simple binaries that can be executed separately if needed.

### 1.9.12 Checking For Corrupt Qt Signal Slot Connection

Use this alias for running digikam:

```
alias digikamdbg="digikam 2>&1 | tee - /tmp/digikam.out; echo -e \"\n\nPossible connection errors:\n\n\";
cat /tmp/digikam.out | grep -A2 'Object::connect'"
```

It will print a list of connection warnings after terminating the program. Moreover the complete console log of the last session is stored in /tmp/digikam.out.

### 1.9.13 Finding Duplicated Code

Code duplication should be avoided as bugs have to be fixed for every piece of duplicated code. The current duplication can be analyzed eg. with Simian: <http://www.redhillconsulting.com.au/products/simian/>

In the digikam checkout directory run:

```
java -jar simian.jar `find . -regex '.*\.(cpp|\.h)' | grep -v 3rdparty`
```

This prints out a list of duplicated code segments.

### 1.9.14 API Documentation Validation, User Documentation Validation, Source Code Checking

The following site check on a daily basis for the a.m. errors: [www.englishbreakfastnetwork.org/krazy/](http://www.englishbreakfastnetwork.org/krazy/)

It can be very useful, in particular before major releases. Don't trust it blindly! Sometimes they propose too advanced modifications that are no compatible with the prevailing include files.

### 1.9.15 Usability Issues

OpenUsability project has define default menu structure and keyboard shortcuts:

[http://wiki.openusability.org/guidelines/index.php/Appendices:Keyboard\\_Shortcuts](http://wiki.openusability.org/guidelines/index.php/Appendices:Keyboard_Shortcuts)

### 1.9.16 Generate API Documentation

To generate API documentation, you need to install:

- Doxygen program ( <http://www.doxygen.org>)
- Dot program ( <http://www.graphviz.org>)
- TeXLive package full version to generate the API doc in PDF ( <https://tug.org/texlive/>)

Under Ubuntu run:

```
sudo apt install doxygen graphviz texlive-full
```

After cmake generated a Makefile you can call 'make doc'. A new subfolder named 'html' will be created. Warning, this can take a while.

For finding documentation errors, doxygen generates a warning log file at the cmake binary dir called 'doxygen-warn.log'.

### 1.9.17 Speed Up The Code-Compile-Test Cycle

Assuming you have setup your environment in `~/.bashrc` as is suggested for development, you can add something like this to your `~/.bashrc`:

```
function digikam_start
{
LD_LIBRARY_PATH=${KDE_BUILDD}/extragear/graphics/lib:${LD_LIBRARY_PATH}
  ${KDE_BUILDD}/extragear/graphics/digikam/digikam/digikam
}

function digikam_start_gdb
{
LD_LIBRARY_PATH=${KDE_BUILDD}/extragear/graphics/lib:${LD_LIBRARY_PATH} gdb
  ${KDE_BUILDD}/extragear/graphics/digikam/digikam/digikam
}
```

This allows you to run digikam after compiling without the need of a "make install", even if you changed code in the libraries.

### 1.9.18 Working With Branches From Git Repository

Example to create a local 'dplugins' development branch based on master:

```
git checkout master
git checkout -b development/dplugins
```

Example to delete the local 'dplugins' development branch:

```
git checkout master
git branch -d development/dplugins
```

Example to create a remote 'dplugins' development branch from the local branch:

```
git push -u origin development/dplugins
```

Example to delete the remote 'dplugins' development branch:

```
git push origin :development/dplugins
```

### 1.9.19 Sync a Branch With Master From Git Repository

It typical to use a dedicated development branch in Git to keep the master code stable for production. To synchronize branches with master, use these commands in your local branch checkout:

```
$>git checkout master
$>git pull --rebase
$>git checkout -b MY_DEVEL_BRANCH GIT_REMOTE_PATH
  Branch 'MY_DEVEL_BRANCH' set up to track remote branch path 'GIT_REMOTE_PATH' from origin.
  To list GIT_REMOTE_PATH, use 'git branch -a' command
  Switched to a new branch 'MY_DEVEL_BRANCH'
$>git merge master
  Merge made by the 'recursive' strategy.
  ...
$>git push
  ...
```

The first 2 lines make sure that your local master repository is up to date. The 3rd line creates the local development branch "MY\_DEVEL\_BRANCH". If you have already created this branch, just run "git checkout MY\_DEVEL\_BRANCH". Merging between master and "MY\_DEVEL\_BRANCH" branch is done in the 4th line. Git might ask you to resolve conflicts here. When it's done, it will ask you to provide a commit message. Finally you push your merge into the remote repository.





# Chapter 2

## Namespace Index

### 2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">Digikam</a> . . . . .	<a href="#">89</a>
<a href="#">Digikam::Matrix</a> . . . . .	<a href="#">137</a>



# Chapter 3

## Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

A	
Digikam::ParallelAdapter< A > . . . . .	2819
Digikam::AlbumPointer< Digikam::Album > . . . . .	321
Digikam::AlbumPointer< Digikam::SAlbum > . . . . .	321
Digikam::AlbumPointer< Digikam::TAlbum > . . . . .	321
boost::default_bfs_visitor	
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::BreadthFirstSearchVisitor . . .	1847
boost::default_dfs_visitor	
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::DepthFirstSearchVisitor . . .	1849
Digikam::CoreDbBackendPrivate::ChangesetContainer< Digikam::AlbumChangeset > . . . . .	731
Digikam::CoreDbBackendPrivate::ChangesetContainer< Digikam::AlbumRootChangeset > . . . . .	731
Digikam::CoreDbBackendPrivate::ChangesetContainer< Digikam::CollectionImageChangeset > . . . . .	731
Digikam::CoreDbBackendPrivate::ChangesetContainer< Digikam::ImageChangeset > . . . . .	731
Digikam::CoreDbBackendPrivate::ChangesetContainer< Digikam::ImageTagChangeset > . . . . .	731
Digikam::CoreDbBackendPrivate::ChangesetContainer< Digikam::SearchChangeset > . . . . .	731
Digikam::CoreDbBackendPrivate::ChangesetContainer< Digikam::TagChangeset > . . . . .	731
cv::ParallelLoopBody	
Digikam::OpenCVDNNFaceRecognizer::Private::ParallelRecognizer . . . . .	2802
Digikam::OpenCVDNNFaceRecognizer::Private::ParallelTrainer . . . . .	2803
Digikam::AbstractAlbumTreeView::ContextMenuElement . . . . .	157
Digikam::AbstractAlbumTreeView::Private . . . . .	158
Digikam::AbstractMarkerTiler::ClickInfo . . . . .	193
Digikam::AbstractMarkerTiler::NonEmptyIterator . . . . .	193
Digikam::AbstractMarkerTiler::Tile . . . . .	194
Digikam::ActionData . . . . .	207
Digikam::Album . . . . .	252
Digikam::DAlbum . . . . .	770
Digikam::PAlbum . . . . .	2811
Digikam::SAlbum . . . . .	3002
Digikam::TAlbum . . . . .	3436
Digikam::AlbumChangeset . . . . .	262
Digikam::AlbumCopyMoveHint . . . . .	262
Digikam::AlbumInfo . . . . .	280
Digikam::AlbumIterator . . . . .	280
Digikam::AlbumManager::Private . . . . .	305

Digikam::AlbumManagerCreator	306
Digikam::AlbumPointer< T >	321
Digikam::AlbumRootChangeset	324
Digikam::AlbumRootInfo	324
Digikam::AlbumShortInfo	351
Digikam::AlbumSimplified	352
Digikam::AltLangStrEdit::Private	373
Digikam::AntiVignettingContainer	377
Digikam::ApplicationSettings::Private	391
Digikam::AssignedBatchTools	396
Digikam::AssignNameWidget::Private	409
Digikam::AutoTagsAssign	426
Digikam::BaloolInfo	458
Digikam::BatchToolSet	472
Digikam::BCGContainer	474
Digikam::BdEngineBackend::QueryState	489
Digikam::BdEngineBackendPrivate::AbstractUnlocker	493
Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker	494
Digikam::BdEngineBackendPrivate::BusyWaiter	495
Digikam::BdEngineBackendPrivate::ErrorLocker	497
Digikam::BorderContainer	520
Digikam::BWSepiaContainer	531
Digikam::CachedPixmapKey	538
Digikam::CachedPixmaps	538
Digikam::CameraMessageBox	550
Digikam::CameraNameHelper	550
Digikam::CameraType	555
Digikam::CamItemInfo	555
Digikam::CamItemSortSettings	558
Digikam::CaptionValues	571
Digikam::CBCContainer	577
Digikam::ChangingDB	583
Digikam::ChoiceSearchModel::Entry	599
Digikam::CMat	604
Digikam::CollectionImageChangeset	605
Digikam::CollectionLocation	607
Digikam::AlbumRootLocation	325
Digikam::CollectionManager::Private	618
Digikam::CollectionScanner::Private	628
Digikam::CollectionScannerHintContainer	630
Digikam::CollectionScannerHintContainerImplementation	632
Digikam::CollectionScannerObserver	635
Digikam::InitializationObserver	2147
Digikam::ScanController	3009
Digikam::SimpleCollectionScannerObserver	3239
Digikam::ColorFXContainer	636
Digikam::CommentInfo	650
Digikam::ContentAwareContainer	654
Digikam::ContextMenuHelper::Private	670
Digikam::CopyrightInfo	672
Digikam::CoreDB	673
Digikam::CoreDbAccess	721
Digikam::CoreDbAccessUnlock	724
Digikam::CoreDbBackendPrivate::ChangesetContainer< T >	731
Digikam::CoreDbDownloadHistory	732
Digikam::CoreDbNameFilter	733
Digikam::CoreDbOperationGroup	734

Digikam::CoreDbPrivilegesChecker	735
Digikam::CoreDbSchemaUpdater	735
Digikam::CoreDbTransaction	735
Digikam::CurvesContainer	749
Digikam::DAlbumInfo	773
Digikam::DatabaseBlob	775
Digikam::DatabaseFields::DatabaseFieldsEnumIterator< FieldName >	776
Digikam::DatabaseFields::DatabaseFieldsEnumIteratorSetOnly< FieldName >	777
Digikam::DatabaseFields::FieldMetaInfo< FieldName >	778
Digikam::DatabaseFields::Set	779
Digikam::DatabaseServerError	788
Digikam::DatabaseSettingsWidget::Private	791
Digikam::DateFormat	815
Digikam::DbEngineAccess	834
Digikam::DbEngineAction	834
Digikam::DbEngineActionElement	834
Digikam::DbEngineActionType	834
Digikam::DbEngineConfig	835
Digikam::DbEngineConfigSettings	836
Digikam::DbEngineConfigSettingsLoader	836
Digikam::DbEngineErrorAnswer	838
Digikam::BdEngineBackendPrivate	490
Digikam::CoreDbBackendPrivate	729
Digikam::DbEngineLocking	842
Digikam::DbEngineParameters	842
Digikam::DbEngineThreadData	847
Digikam::DBJobInfo	858
Digikam::AlbumsDBJobInfo	327
Digikam::DatesDBJobInfo	821
Digikam::GPSDBJobInfo	1796
Digikam::SearchesDBJobInfo	3030
Digikam::TagsDBJobInfo	3406
Digikam::DbKeysCollection	864
Digikam::CommonKeys	651
Digikam::MetadataKeys	2640
Digikam::PositionKeys	2847
Digikam::DCategorizedSortFilterProxyModel::Private	882
Digikam::DCategorizedView::Private	886
Digikam::DCategorizedView::Private::ElementInfo	890
Digikam::DColor	897
Digikam::DColorComposer	900
Digikam::DConfigDlgModelPrivate	927
Digikam::DConfigDlgWdgModelPrivate	963
Digikam::DConfigDlgTitle::Private	935
Digikam::DConfigDlgViewPrivate	942
Digikam::DConfigDlgWdgPrivate	964
Digikam::DDatePicker::Private	975
Digikam::DDateTable::Private::DatePaintingMode	986
Digikam::DeltaTime	1007
Digikam::DFileOperations	1020
Digikam::DigikamApp::Private	1046
Digikam::DImageHistory	1067
Digikam::DImageHistory::Entry	1070
Digikam::DImgBuiltinFilter	1089
Digikam::DImgFilterGenerator	1097
Digikam::BasicDImgFilterGenerator< T >	461
Digikam::DImgFilterManager	1099

Digikam::DImgLoader	1102
Digikam::DImgLoaderObserver	1105
Digikam::lccTransformFilter	1945
Digikam::LoadingTask	2559
Digikam::SharedLoadingTask	3192
Digikam::PreviewLoadingTask	2851
Digikam::ThumbnailLoadingTask	3473
Digikam::SavingTask	3006
Digikam::DImgStaticPriv	1110
Digikam::DisjointMetadataDataFields	1145
Digikam::DisjointMetadata::Private	1143
Digikam::DItemInfo	1155
Digikam::DMessageBox	1169
Digikam::DMetadataSettingsContainer	1191
Digikam::DMultiTabBar::Private	1202
Digikam::DMultiTabBarFrame::Private	1206
Digikam::DMultiTabBarTab::Private	1210
Digikam::DNGWriter	1212
Digikam::DNGWriter::Private	1213
Digikam::DNNBaseDetectorModel	1216
Digikam::DNNResnetDetector	1241
Digikam::DNNYoloDetector	1245
Digikam::DNNFaceDetectorBase	1219
Digikam::DNNFaceDetectorSSD	1221
Digikam::DNNFaceDetectorYOLO	1223
Digikam::DNNFaceDetectorYuNet	1225
Digikam::DNNFaceExtractorBase	1227
Digikam::DNNOpenFaceExtractor	1239
Digikam::DNNSFaceExtractor	1243
Digikam::DNNModelBase	1229
Digikam::DNNModelConfig	1230
Digikam::DNNModelNet	1234
Digikam::DNNModelSFace	1236
Digikam::DNNModelYuNet	1237
Digikam::DNNModelInfoContainer	1231
Digikam::DOnlineTranslator::Private	1280
Digikam::DOnlineTranslatorOption	1283
Digikam::DOnlineTts::Private	1290
Digikam::DownloadInfo	1290
Digikam::DownloadSettings	1291
Digikam::DPixelsAliasFilter	1292
Digikam::DPlainTextEdit::Private	1297
Digikam::DPluginAuthor	1305
Digikam::DPluginLoader::Private	1338
Digikam::DragDropModelImplementation	1361
Digikam::ImportItemModel	2065
Digikam::ImportThumbnailModel	2129
Digikam::ItemHistoryGraphModel	2283
Digikam::ItemModel	2345
Digikam::ItemThumbnailModel	2435
Digikam::ItemAlbumModel	2174
Digikam::ItemListModel	2334
ShowFoto::ShowfotoItemModel	3695
ShowFoto::ShowfotoThumbnailModel	3759
Digikam::DragDropViewImplementation	1364
Digikam::ItemViewCategorized	2442

Digikam::ImportCategorizedView . . . . .	2011
Digikam::ImportIconView . . . . .	2057
Digikam::ImportThumbnailBar . . . . .	2114
Digikam::ItemCategorizedView . . . . .	2182
Digikam::DigikamItemView . . . . .	1057
Digikam::ItemThumbnailBar . . . . .	2419
Digikam::LightTableThumbBar . . . . .	2525
ShowFoto::ShowfotoCategorizedView . . . . .	3649
ShowFoto::ShowfotoThumbnailBar . . . . .	3746
Digikam::TableViewTreeView . . . . .	3328
Digikam::VersionsTreeView . . . . .	3585
Digikam::DRawDecoder::Private . . . . .	1374
Digikam::DRawDecoderSettings . . . . .	1374
Digikam::DRawDecoding . . . . .	1385
Digikam::DRawInfo . . . . .	1387
Digikam::DServiceInfo . . . . .	1402
Digikam::DServiceMenu . . . . .	1402
Digikam::DTextEdit::Private . . . . .	1415
Digikam::DToolTipStyleSheet . . . . .	1419
Digikam::DTrash . . . . .	1420
Digikam::DTrashItemInfo . . . . .	1421
Digikam::DWItemDelegatePool . . . . .	1438
Digikam::DWItemDelegatePoolPrivate . . . . .	1439
Digikam::DXmlGuiWindow::Private . . . . .	1449
Digikam::EditorCore::Private . . . . .	1468
Digikam::EditorCore::Private::FileToSave . . . . .	1469
Digikam::EditorWindow::Private . . . . .	1488
Digikam::EffectMngr . . . . .	1489
Digikam::EffectMngr::Private . . . . .	1490
Digikam::Ellipsoid . . . . .	1491
Digikam::ExifToolParser::Private . . . . .	1524
Digikam::ExifToolProcess::Private::Command . . . . .	1531
Digikam::ExifToolProcess::Result . . . . .	1532
Digikam::ExposureSettingsContainer . . . . .	1539
Digikam::FaceDb . . . . .	1543
Digikam::FaceDb::Private . . . . .	1546
Digikam::FaceDbAccess . . . . .	1547
Digikam::FaceDbAccessUnlock . . . . .	1547
Digikam::FaceDbOperationGroup . . . . .	1551
Digikam::FaceDbSchemaUpdater . . . . .	1552
Digikam::FaceDetector . . . . .	1553
Digikam::FaceGroup::Private . . . . .	1559
Digikam::FacelItemRetriever . . . . .	1563
Digikam::FacePipelinePackage . . . . .	1595
Digikam::FacePipelineExtendedPackage . . . . .	1589
Digikam::FacePreprocessor . . . . .	1610
Digikam::RecognitionPreprocessor . . . . .	2944
Digikam::FaceScanSettings . . . . .	1622
Digikam::FaceScanWidget::Private . . . . .	1628
Digikam::FaceTags . . . . .	1635
Digikam::FaceTagsEditor . . . . .	1638
Digikam::FaceUtils . . . . .	1648
Digikam::FaceTagsIface . . . . .	1644
Digikam::FacePipelineFaceTagsIface . . . . .	1591
Digikam::FacialRecognitionWrapper . . . . .	1652
Digikam::FacialRecognitionWrapper::Private . . . . .	1655
Digikam::FFMpegConfigHelper . . . . .	1659

Digikam::FieldQueryBuilder	1663
Digikam::FileActionProgressItemCreator	1681
Digikam::PrivateProgressItemCreator	2864
Digikam::FileReadLocker	1684
Digikam::FileReadWriteLockKey	1684
Digikam::FileWriteLocker	1691
Digikam::FilmContainer	1691
Digikam::FilmContainer::Private	1693
Digikam::FilmFilter::Private	1697
Digikam::FilmGrainContainer	1697
Digikam::FilmProfile	1702
Digikam::Filter	1703
Digikam::FilterAction	1704
Digikam::DImgThreadedFilter::DefaultFilterAction< Filter >	1123
Digikam::FocusPoint	1729
Digikam::FocusPointGroup::Private	1733
Digikam::FrameOsd	1738
Digikam::FrameOsdSettings	1740
Digikam::FrameUtils	1741
Digikam::FreeRotationContainer	1741
Digikam::FullObjectDetection	1751
Digikam::FuzzySearchView::Private	1759
Digikam::GeoCoordinates	1760
Digikam::GeodeticCalculator	1761
Digikam::GeofaceCluster	1770
Digikam::GeofaceInternalWidgetInfo	1773
Digikam::GeolocationSettingsContainer	1779
Digikam::GPSDataContainer	1795
Digikam::GPSItemContainer	1803
Digikam::ItemGPS	2269
Digikam::GPSItemInfo	1807
Digikam::GPSUndoCommand::UndoInfo	1836
Digikam::Graph< VertexProperties, EdgeProperties >	1837
Digikam::ItemHistoryGraphData	2279
Digikam::Graph< VertexProperties, EdgeProperties >::DominatorTree	1845
Digikam::Graph< VertexProperties, EdgeProperties >::Edge	1845
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch	1846
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor	1848
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::BreadthFirstSearchVisitor	1847
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::DepthFirstSearchVisitor	1849
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::lessThanMapEdgeToTarget< GraphType, VertexLessThan >	1850
Digikam::Graph< VertexProperties, EdgeProperties >::Path	1850
Digikam::Graph< VertexProperties, EdgeProperties >::Vertex	1851
Digikam::GreycstorageContainer	1856
Digikam::GroupedImagesFinder	1863
Digikam::GroupingViewImplementation	1870
Digikam::DigikamItemView	1057
Digikam::ItemThumbnailBar	2419
Digikam::TableViewTreeView	3328
Digikam::GroupItemFilterSettings	1871
Digikam::GroupStateComputer	1872
Digikam::Haar::Calculator	1872
Digikam::Haar::ImageData	1873
Digikam::Haar::SignatureData	1873
Digikam::Haar::SignatureMap	1874
Digikam::Haar::WeightBin	1874



Digikam::Haar::Weights	1875
Digikam::Haarface	1875
Digikam::Haarface::Private	1879
Digikam::HaarProgressObserver	1880
Digikam::DuplicatesProgressObserver	1431
Digikam::HistoryEdgeProperties	1893
Digikam::HistoryImageld	1894
Digikam::HistoryVertexProperties	1896
Digikam::HotPixelContainer	1896
Digikam::HotPixelProps	1901
Digikam::HotPixelsWeights	1903
Digikam::HSLContainer	1907
Digikam::lccManager	1915
Digikam::lccPostLoadingManager	1919
Digikam::lccProfile	1922
Digikam::lccSettings::Private	1940
Digikam::ICCSettingsContainer	1940
Digikam::lccTransform	1942
Digikam::Identity	1949
Digikam::IdentityProvider	1950
Digikam::ImageChangeset	1953
Digikam::ImageCommonContainer	1954
Digikam::ImageCurves	1954
Digikam::ImageDialog::Private	1959
Digikam::ImageHistoryEntry	1968
Digikam::ImageIface	1968
Digikam::ImageLevels	1972
Digikam::ImageListProvider	1974
Digikam::EmptyImageListProvider	1504
Digikam::QListImageListProvider	2889
Digikam::ImageMetadataContainer	1975
Digikam::ImageQualityCalculator	1977
Digikam::ImageQualityCalculator::ResultDetection	1978
Digikam::ImageQualityContainer	1979
Digikam::ImageQualityParser::Private	1981
Digikam::ImageRelation	1994
Digikam::ImageTagChangeset	1997
Digikam::ImageTagProperty	1998
Digikam::ImageTagPropertyName	1999
Digikam::ImageWindow::Private	2006
Digikam::ImageZoomSettings	2006
Digikam::ImportUI::Private	2137
Digikam::InfraredContainer	2142
Digikam::InternalTagName	2149
Digikam::IOFileSettings	2153
Digikam::IOJobData	2155
Digikam::lptcCoreContactInfo	2163
Digikam::lptcCoreLocationInfo	2163
Digikam::ItemChangeHint	2192
Digikam::ItemComments	2193
Digikam::ItemCopyMoveHint	2202
Digikam::ItemCopyright	2203
Digikam::ItemDelegateOverlayContainer	2220
Digikam::ItemViewDelegate	2449
Digikam::ItemDelegate	2208
Digikam::DigikamItemDelegate	1049
Digikam::ItemFaceDelegate	2231

Digikam::ItemThumbnailDelegate . . . . .	2426
Digikam::ItemViewImportDelegate . . . . .	2459
Digikam::ImportDelegate . . . . .	2032
Digikam::ImportNormalDelegate . . . . .	2083
Digikam::ImportThumbnailDelegate . . . . .	2120
Digikam::VersionsDelegate . . . . .	3582
ShowFoto::ShowfotoItemViewDelegate . . . . .	3705
ShowFoto::ShowfotoDelegate . . . . .	3662
ShowFoto::ShowfotoNormalDelegate . . . . .	3713
ShowFoto::ShowfotoThumbnailDelegate . . . . .	3752
Digikam::ItemDescEditTab::Private . . . . .	2224
Digikam::ItemExtendedProperties . . . . .	2228
Digikam::ItemFilterModelPrepareHook . . . . .	2251
Digikam::ItemFilterModelTodoPackage . . . . .	2254
Digikam::ItemFilterSettings . . . . .	2257
Digikam::ItemFiltersHistoryTreeItem . . . . .	2261
Digikam::ItemHistoryGraph . . . . .	2274
Digikam::ItemIconView::Private . . . . .	2291
Digikam::ItemInfo . . . . .	2292
Digikam::ItemInfoSet . . . . .	2315
Digikam::ItemInfoStatic . . . . .	2315
Digikam::ItemLister . . . . .	2320
Digikam::ItemLister::Private . . . . .	2323
Digikam::ItemListerReceiver . . . . .	2330
Digikam::ItemListerValueListReceiver . . . . .	2332
Digikam::ItemListerJobReceiver . . . . .	2328
Digikam::ItemListerJobPartsSendingReceiver . . . . .	2326
Digikam::ItemListerJobGrowingPartsSendingReceiver . . . . .	2324
Digikam::ItemListerRecord . . . . .	2331
Digikam::ItemMetadataAdjustmentHint . . . . .	2343
Digikam::ItemPosition . . . . .	2354
Digikam::ItemPropertiesTab::Private . . . . .	2381
Digikam::ItemQueryBuilder . . . . .	2383
Digikam::ItemQueryPostHook . . . . .	2383
Digikam::ItemQueryPostHooks . . . . .	2384
Digikam::ItemScanInfo . . . . .	2395
Digikam::ItemScanner . . . . .	2395
Digikam::ItemScanner::Private . . . . .	2402
Digikam::ItemScannerCommit . . . . .	2402
Digikam::ItemShortInfo . . . . .	2412
Digikam::ItemSortSettings . . . . .	2413
Digikam::ItemTagPair . . . . .	2417
Digikam::ItemViewDelegatePrivate . . . . .	2455
Digikam::ItemDelegate::ItemDelegatePrivate . . . . .	2215
Digikam::DigikamItemDelegatePrivate . . . . .	1054
Digikam::ItemFaceDelegatePrivate . . . . .	2236
Digikam::ItemThumbnailDelegatePrivate . . . . .	2432
Digikam::ItemViewImportDelegatePrivate . . . . .	2465
Digikam::ImportDelegate::ImportDelegatePrivate . . . . .	2039
Digikam::ImportNormalDelegatePrivate . . . . .	2088
Digikam::ImportThumbnailDelegatePrivate . . . . .	2126
Digikam::JPEGUtils::digikam_source_mgr . . . . .	2477
Digikam::JPEGUtils::JpegRotator . . . . .	2477
Digikam::KDNNodeBase . . . . .	2480
Digikam::KDNNodeOpenFace . . . . .	2483
Digikam::KDNNodeSFace . . . . .	2485

Digikam::KNodeBase::NodeCompareResult	2482
Digikam::KDTreeBase	2487
Digikam::KDTreeOpenFace	2489
Digikam::KDTreeSFace	2490
Digikam::LcmsLock	2502
Digikam::LensDistortionPixelAccess	2506
Digikam::LensFunContainer	2508
Digikam::LensFunIface	2512
Digikam::LessThanByProximityToSubject	2515
Digikam::LevelsContainer	2515
Digikam::LightTableWindow::Private	2537
Digikam::LoadingCache::CacheLock	2548
Digikam::LoadingCacheInterface	2550
Digikam::LoadingDescription	2551
Digikam::LoadingDescription::PostProcessingParameters	2555
Digikam::LoadingDescription::PreviewParameters	2556
Digikam::LoadingProcess	2557
Digikam::SharedLoadingTask	3192
Digikam::LoadingProcessListener	2558
Digikam::SharedLoadingTask	3192
Digikam::LoadSaveFileInfoProvider	2561
Digikam::DatabaseLoadSaveFileInfoProvider	780
Digikam::LoadSaveNotifier	2563
Digikam::LoadSaveThread	2567
Digikam::ManagedLoadSaveThread	2604
Digikam::PreviewLoadThread	2854
Digikam::FacePreviewLoader	1611
Digikam::SharedLoadSaveThread	3196
Digikam::ThumbnailLoadThread	3476
Digikam::LoadSaveTask	2565
Digikam::LoadingTask	2559
Digikam::SavingTask	3006
Digikam::LocalContrastContainer	2573
Digikam::LocalizeContainer	2579
Digikam::LookupAltitude::Request	2585
Digikam::LookupFactory	2588
Digikam::MaintenanceData	2589
Digikam::MaintenanceDlg::Private	2590
Digikam::MaintenanceSettings	2592
Digikam::MapWidget::Private	2625
Digikam::Mat	2630
Digikam::MetadataHub	2633
Digikam::MetaEngine	2667
Digikam::DMetadata	1173
Digikam::MetaEngine::Private	2704
Digikam::MetaEngineData	2706
Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >	2707
Digikam::MetaEnginePreviews	2708
Digikam::MetaEngineRotation	2709
Digikam::MetaEngineSettingsContainer	2713
Digikam::MixerContainer	2719
Digikam::MLClassifierFoundation	2725
Digikam::FaceClassifierBase	1542
Digikam::FaceClassifier	1540
Digikam::MLClassifierFoundation::VotingGroups	2726
Digikam::MLClassifierFoundation::VotingGroups::VoteTally	2726

Digikam::MLPipelineFoundation::_MLPipelinePerformanceProfile	2730
Digikam::MLPipelinePackageFoundation	2731
Digikam::FacePipelinePackageBase	1596
Digikam::NamespaceEntry	2754
Digikam::NewlyAppearedFile	2762
Digikam::NRContainer	2781
Digikam::OpenCVDNNFaceDetector	2797
Digikam::OpenCVDNNFaceRecognizer	2799
Digikam::OpenCVDNNFaceRecognizer::Private	2801
Digikam::OpenfacePreprocessor	2803
Digikam::PageItem	2810
Digikam::PAlbumPath	2813
Digikam::ParallelWorkers	2824
Digikam::ParallelAdapter< Digikam::FileWorkerInterface >	2819
Digikam::ParallelAdapter< A >	2819
Digikam::Parser	2827
Digikam::DefaultRenameParser	994
Digikam::ImportRenameParser	2098
Digikam::ParseResults	2828
Digikam::ParseSettings	2829
Digikam::PhotoInfoContainer	2838
Digikam::PointTransformAffine	2847
Digikam::PreviewSettings	2861
Digikam::ProgressEntry	2868
Digikam::QueueMgrWindow::Private	2898
Digikam::QueueSettings	2901
Digikam::RandomNumberGenerator	2908
Digikam::RatingStarDrawer	2929
Digikam::RatingComboBoxDelegate	2919
Digikam::RatingComboBoxWidget	2921
Digikam::RecognitionBenchmark::Statistics	2943
Digikam::RecognitionTrainingUpdateQueue	2946
Digikam::RedEye::RegressionTree	2949
Digikam::RedEye::ShapePredictor	2950
Digikam::RedEye::SplitFeature	2951
Digikam::RedEyeCorrectionContainer	2951
Digikam::RefocusMatrix	2960
Digikam::RGInfo	2978
Digikam::RuleType	3000
Digikam::RuleTypeForConversion	3000
Digikam::SaveProperties	3005
Digikam::SavingContext	3005
Digikam::ScanController::FileMetadataWrite	3015
Digikam::ScanController::Private	3015
Digikam::ScanControllerCreator	3016
Digikam::SchemeManager	3020
Digikam::SearchChangeset	3029
Digikam::SearchInfo	3106
Digikam::SearchTextSettings	3131
Digikam::SearchTextFilterSettings	3130
Digikam::SearchViewThemedPartsCache	3141
Digikam::SearchView	3137
Digikam::SetupCollectionModel::Item	3172
Digikam::SetupMetadata::Private	3183
Digikam::SetupMisc::Private	3186
Digikam::SharedQueue< T >	3199
Digikam::SharpContainer	3199

Digikam::Sidebar::Private	3220
Digikam::SidebarSplitter::Private	3223
Digikam::SidebarState	3223
Digikam::SidecarFinder	3227
Digikam::SimilarityDb	3227
Digikam::SimilarityDbAccess	3233
Digikam::SimilarityDbSchemaUpdater	3239
Digikam::SimpleTreeModel::Item	3241
Digikam::SolidVolumeInfo	3250
Digikam::State	3259
Digikam::StateSavingObject	3259
Digikam::AbstractAlbumTreeView	147
Digikam::AbstractCountingAlbumTreeView	183
Digikam::AbstractCheckableAlbumTreeView	171
Digikam::AlbumTreeView	360
Digikam::AlbumSelectTreeView	345
Digikam::AlbumSelectionTreeView	335
Digikam::SearchTreeView	3132
Digikam::EditableSearchTreeView	1459
Digikam::NormalSearchTreeView	2775
Digikam::TagTreeView	3424
Digikam::TagFolderView	3351
Digikam::TagCheckView	3331
Digikam::TagFilterView	3344
Digikam::TagMngrTreeView	3367
Digikam::DateTreeView	827
Digikam::DateFolderView	809
Digikam::FaceScanWidget	1626
Digikam::FilterSideBarWidget	1714
Digikam::FuzzySearchView	1757
Digikam::GPSSearchView	1832
Digikam::LabelsTreeView	2498
Digikam::MapWidgetView	2626
Digikam::SearchTextBar	3123
Digikam::SearchTextBarDb	3127
Digikam::Sidebar	3214
Digikam::ImportItemPropertiesSideBarImport	2073
Digikam::ItemPropertiesSideBar	2368
Digikam::ItemPropertiesSideBarDB	2372
Digikam::SidebarWidget	3224
Digikam::AlbumFolderViewSideBarWidget	274
Digikam::DateFolderViewSideBarWidget	812
Digikam::FuzzySearchSideBarWidget	1753
Digikam::GPSSearchSideBarWidget	1828
Digikam::LabelsSideBarWidget	2495
Digikam::PeopleSideBarWidget	2830
Digikam::SearchSideBarWidget	3119
Digikam::TagViewSideBarWidget	3432
Digikam::TimelineSideBarWidget	3504
Digikam::TableView	3282
Digikam::TagsManager	3416
ShowFoto::ShowfotoFolderViewSideBar	3686
ShowFoto::ShowfotoStackViewSideBar	3742
Digikam::SubjectData	3274
Digikam::SubQueryBuilder	3278
Digikam::SystemSettings	3279
Digikam::TableViewColumnConfiguration	3288

Digikam::TableViewColumnDescription	3289
Digikam::TableViewColumnProfile	3290
Digikam::TableViewModel::Item	3326
Digikam::TableViewShared	3327
Digikam::TagChangeset	3330
Digikam::TagData	3339
Digikam::TaggingAction	3358
Digikam::TaggingActionFactory	3359
Digikam::TaggingActionFactory::ConstraintInterface	3361
Digikam::TagInfo	3361
Digikam::TagProperties	3385
Digikam::TagProperty	3391
Digikam::TagPropertyName	3392
Digikam::TagRegion	3393
Digikam::TagShortInfo	3410
Digikam::Template	3439
Digikam::TextureContainer	3451
Digikam::ThemeManager::Private	3457
Digikam::ThumbnailCreator	3461
Digikam::ThumbnailCreator::Private	3465
Digikam::ThumbnailIdentifier	3466
Digikam::ThumbnailInfo	3470
Digikam::ThumbnailImage	3467
Digikam::ThumbnailImageCatcher::Private	3469
Digikam::ThumbnailImageCatcher::Private::CatcherResult	3469
Digikam::ThumbnailInfoProvider	3472
Digikam::ThumbsDbInfoProvider	3493
Digikam::ThumbnailLoadThread::Private	3485
Digikam::ThumbnailLoadThreadStaticPriv	3485
Digikam::ThumbnailResult	3486
Digikam::ThumbnailSize	3486
Digikam::ThumbsDb	3487
Digikam::ThumbsDbAccess	3488
Digikam::ThumbsDbInfo	3493
Digikam::ThumbsDbSchemaUpdater	3494
Digikam::TileIndex	3500
Digikam::TimeAdjustContainer	3501
Digikam::TonalityContainer	3512
Digikam::TooltipCreator	3521
Digikam::TrackCorrelator::Correlation	3524
Digikam::TrackCorrelator::CorrelationOptions	3524
Digikam::TrackManager::Track	3528
Digikam::TrackManager::TrackPoint	3529
Digikam::TrackReader::TrackReadResult	3530
Digikam::TrainingDataProvider	3534
Digikam::RecognitionTrainingProvider	2945
Digikam::TransitionMngr	3539
Digikam::TransitionMngr::Private	3539
Digikam::TreeBranch	3544
Digikam::UndoAction	3561
Digikam::UndoActionIrreversible	3562
Digikam::UndoActionReversible	3563
Digikam::UndoCache	3564
Digikam::UndoManager	3565
Digikam::UndoMetadataContainer	3565
Digikam::UndoState	3566
Digikam::VersionFileInfo	3573

Digikam::VersionFileOperation	3573
Digikam::VersionItemFilterSettings	3575
Digikam::VersionManager	3577
Digikam::DatabaseVersionManager	794
Digikam::VersionManagerSettings	3578
Digikam::VersionNamingScheme	3579
Digikam::DefaultVersionNamingScheme	999
Digikam::VideoFrame	3588
Digikam::VideoInfoContainer	3588
Digikam::VideoMetadataContainer	3589
Digikam::VideoStripFilter	3589
Digikam::VideoThumbDecoder	3589
Digikam::VideoThumbDecoder::Private	3590
Digikam::VideoThumbnailer	3590
Digikam::VideoThumbWriter	3591
Digikam::VidSlideSettings	3591
Digikam::VisibilityObject	3604
Digikam::SearchField	3036
Digikam::SearchFieldAlbum	3039
Digikam::SearchFieldCheckBox	3043
Digikam::SearchFieldChoice	3047
Digikam::SearchFieldComboBox	3054
Digikam::SearchFieldColorDepth	3051
Digikam::SearchFieldPageOrientation	3071
Digikam::SearchFieldLabels	3063
Digikam::SearchFieldMonthDay	3067
Digikam::SearchFieldRangeDate	3074
Digikam::SearchFieldRangeDouble	3078
Digikam::SearchFieldRangeInt	3082
Digikam::SearchFieldRangeTime	3086
Digikam::SearchFieldRating	3090
Digikam::SearchFieldText	3094
Digikam::SearchFieldKeyword	3060
Digikam::WBContainer	3604
Digikam::Workflow	3619
Digikam::WSAlbum	3625
Digikam::WSToolUtils	3635
DImgPreviewItemPrivate public GraphicsDImgItem::GraphicsDImgItemPrivate	
Digikam::DImgPreviewItem	1108
Digikam::ItemPreviewCanvas	2358
dng_host	
Digikam::DNGWriterHost	1215
GraphicsDImgItemPrivate	
Digikam::GraphicsDImgItem	1852
Digikam::DImgPreviewItem	1108
Digikam::ImagePreviewItem	1976
Digikam::ImageRegionItem	1990
ItemFilterModelPrivate public QObject	
Digikam::ItemFilterModel	2239
Digikam::ItemAlbumFilterModel	2168
KXmlGuiWindow	
Digikam::DXmlGuiWindow	1444
Digikam::DigikamApp	1043
Digikam::EditorWindow	1482
Digikam::ImageWindow	2000
ShowFoto::Showfoto	3643
Digikam::ImportUI	2134

Digikam::LightTableWindow . . . . .	2534
Digikam::QueueMgrWindow . . . . .	2895
Marble::LayerInterface	
Digikam::BackendMarbleLayer . . . . .	455
Digikam::MetaEngineMergeHelper< Exiv2::ExifData, Exiv2::ExifKey, QLatin1String > . . . . .	2707
Digikam::ExifMetaEngineMergeHelper . . . . .	1509
Digikam::MetaEngineMergeHelper< Exiv2::IptcData, Exiv2::IptcKey, QLatin1String > . . . . .	2707
Digikam::IptcMetaEngineMergeHelper . . . . .	2164
Private public QSharedData	
Digikam::DImg . . . . .	1071
QAbstractButton	
Digikam::CoordinatesOverlayWidget . . . . .	670
Digikam::GroupIndicatorOverlayWidget . . . . .	1869
Digikam::ImportOverlayWidget . . . . .	2090
Digikam::ItemViewHoverButton . . . . .	2456
Digikam::FaceRejectionOverlayButton . . . . .	1620
Digikam::ImportRotateOverlayButton . . . . .	2103
Digikam::ItemFullScreenOverlayButton . . . . .	2266
Digikam::ItemRotateOverlayButton . . . . .	2393
Digikam::ItemSelectionOverlayButton . . . . .	2407
ShowFoto::ShowfotoCoordinatesOverlayWidget . . . . .	3660
QAbstractItemDelegate	
Digikam::ComboBoxDelegate . . . . .	649
Digikam::DItemDelegate . . . . .	1151
Digikam::ItemViewDelegate . . . . .	2449
Digikam::ItemViewImportDelegate . . . . .	2459
ShowFoto::ShowfotoItemViewDelegate . . . . .	3705
Digikam::DWItemDelegate . . . . .	1434
Digikam::SetupCollectionDelegate . . . . .	3165
QAbstractItemModel	
Digikam::AbstractAlbumModel . . . . .	140
Digikam::AbstractSpecificAlbumModel . . . . .	197
Digikam::AbstractCountingAlbumModel . . . . .	177
Digikam::AbstractCheckableAlbumModel . . . . .	163
Digikam::AlbumModel . . . . .	307
Digikam::SearchModel . . . . .	3107
Digikam::TagModel . . . . .	3373
Digikam::DateAlbumModel . . . . .	802
Digikam::BookmarksModel . . . . .	519
Digikam::DConfigDigModel . . . . .	925
Digikam::DConfigDigWdgModel . . . . .	956
Digikam::GPSItemModel . . . . .	1813
Digikam::ItemFiltersHistoryModel . . . . .	2260
Digikam::ItemHistoryGraphModel . . . . .	2283
Digikam::RGTagModel . . . . .	2979
Digikam::SetupCollectionModel . . . . .	3169
Digikam::SimpleTreeModel . . . . .	3240
Digikam::TableViewModel . . . . .	3323
Digikam::TagMgrListModel . . . . .	3363
Digikam::TrackListModel . . . . .	3526
QAbstractListModel	
Digikam::ChoiceSearchModel . . . . .	596
Digikam::ImportItemModel . . . . .	2065
Digikam::ItemModel . . . . .	2345
Digikam::ItemVersionsModel . . . . .	2441
Digikam::RatingComboBoxModel . . . . .	2920
ShowFoto::ShowfotoItemModel . . . . .	3695



QAbstractSlider	
Digikam::DSelector	1399
Digikam::DColorValueSelector	903
QAbstractTableModel	
Digikam::DTrashItemModel	1422
QAction	
Digikam::DPluginAction	1303
Digikam::RemoveFilterAction	2968
Digikam::RemoveFilterAction	2968
QByteArray	
Digikam::NonDeterministicRandomData	2769
QComboBox	
Digikam::AdvancedRenameInput	238
Digikam::CountrySelector	746
Digikam::DDateEdit	967
Digikam::GeolocationFilter	1776
Digikam::lccRenderingIntentComboBox	1936
Digikam::ImportFilterComboBox	2048
Digikam::MimeFilter	2718
Digikam::ModelIndexBasedComboBox	2735
Digikam::RatingComboBox	2917
Digikam::StayPoppedUpComboBox	3267
Digikam::ListViewComboBox	2541
Digikam::ChoiceSearchComboBox	593
Digikam::TreeViewComboBox	3545
Digikam::TreeViewLineEditComboBox	3548
Digikam::AlbumSelectComboBox	330
Digikam::AbstractAlbumTreeViewSelectComboBox	159
Digikam::AlbumTreeViewSelectComboBox	365
Digikam::TagTreeViewSelectComboBox	3429
Digikam::AddTagsComboBox	228
Digikam::SqueezedComboBox	3252
Digikam::lccProfilesComboBox	1926
Digikam::TimeZoneComboBox	3509
Digikam::WSComboboxIntermediate	3626
QCompleter	
Digikam::ModelCompleter	2733
Digikam::TagCompleter	3338
QDBusAbstractAdaptor	
CoreDbWatchAdaptor	139
QDialog	
Digikam::AddBookmarkDialog	226
Digikam::AdvancedRenameDialog	237
Digikam::AlbumPropsEdit	323
Digikam::AlbumSelectDialog	334
Digikam::BookmarksDialog	514
Digikam::CameraFolderDialog	542
Digikam::CameraInfoDialog	546
Digikam::CameraSelection	553
Digikam::CaptureDlg	571
Digikam::ChangeFaceRecognitionModelDlg	583
Digikam::ClockPhotoDialog	603
Digikam::ColorCorrectionDlg	636
Digikam::DConfigDlg	909
Digikam::Setup	3159
ShowFoto::ShowfotoSetup	3722
Digikam::DPluginAboutDlg	1302
Digikam::DPluginDialog	1321

Digikam::WSToolDialog . . . . .	3634
Digikam::DProgressDlg . . . . .	1358
Digikam::AdvancedRenameProcessDialog . . . . .	243
Digikam::DatabaseMigrationDialog . . . . .	781
Digikam::DbShrinkDialog . . . . .	871
Digikam::DeleteDialog . . . . .	1002
Digikam::FileSaveOptionsDlg . . . . .	1688
Digikam::FilesDownloader . . . . .	1688
Digikam::GeoPluginAboutDlg . . . . .	1783
Digikam::ICCPProfileInfoDlg . . . . .	1925
Digikam::ImportFilterDlg . . . . .	2049
Digikam::InfoDlg . . . . .	2141
Digikam::DBStatDlg . . . . .	872
Digikam::LibsInfoDlg . . . . .	2520
Digikam::RawCameraDlg . . . . .	2932
Digikam::SolidHardwareDlg . . . . .	3249
Digikam::MaintenanceDlg . . . . .	2589
Digikam::NamespaceEditDlg . . . . .	2753
Digikam::OnlineVersionDlg . . . . .	2796
Digikam::RuleDialog . . . . .	3000
Digikam::DatabaseOptionDialog . . . . .	784
Digikam::DateOptionDialog . . . . .	818
Digikam::DefaultValueDialog . . . . .	995
Digikam::MetadataOptionDialog . . . . .	2646
Digikam::RangeDialog . . . . .	2911
Digikam::ReplaceDialog . . . . .	2972
Digikam::SequenceNumberDialog . . . . .	3155
Digikam::SoftProofDialog . . . . .	3248
Digikam::TableViewConfigurationDialog . . . . .	3321
Digikam::TagEditDlg . . . . .	3342
Digikam::TooltipDialog . . . . .	3521
Digikam::VersioningPromptUserSaveDialog . . . . .	3575
Digikam::VidPlayerDlg . . . . .	3591
Digikam::WSLoginDialog . . . . .	3627
Digikam::WSNewAlbumDialog . . . . .	3628
Digikam::WSSelectUserDlg . . . . .	3629
Digikam::WebBrowserDlg . . . . .	3611
Digikam::WorkflowDlg . . . . .	3620
ShowFoto::ShowfotoFolderViewBookmarkDlg . . . . .	3680
ShowFoto::ShowfotoStackViewFavoriteItemDlg . . . . .	3735
QDockWidget	
Digikam::ThumbBarDock . . . . .	3459
QDoubleSpinBox	
Digikam::CustomStepsDoubleSpinBox . . . . .	760
QFileDialog	
Digikam::DFileDialog . . . . .	1019
QFileIconProvider	
Digikam::ImageDialogIconProvider . . . . .	1959
QFileSystemModel	
ShowFoto::ShowfotoFolderViewModel . . . . .	3685
QFrame	
Digikam::AssignNameWidget . . . . .	405
Digikam::DDatePicker . . . . .	970
Digikam::DHBox . . . . .	1036
Digikam::DDateTimeEdit . . . . .	987
Digikam::DFileSelector . . . . .	1024
Digikam::DFontSelect . . . . .	1033
Digikam::DVBox . . . . .	1432

Digikam::CaptionEdit	568
Digikam::ColorLabelWidget	646
Digikam::ColorLabelFilter	642
Digikam::DateFolderView	809
Digikam::FilterSideBarWidget	1714
Digikam::IccProfilesSettings	1931
Digikam::ItemDescEditTab	2222
Digikam::PickLabelWidget	2844
Digikam::PickLabelFilter	2840
Digikam::RatingBox	2915
Digikam::TransactionItem	3536
ShowFoto::ShowfotoFolderViewBar	3678
Digikam::DZoomBar	1456
Digikam::ImportView	2139
Digikam::ItemIconView	2286
Digikam::LocalizeSelector	2580
Digikam::OverlayWidget	2808
Digikam::ProgressView	2882
Digikam::RatingFilter	2924
Digikam::TemplateSelector	3446
Digikam::TextFilter	3450
Digikam::DLineWidget	1168
Digikam::DMultiTabBarFrame	1205
Digikam::DNotificationPopup	1247
Digikam::DNotificationWidget	1258
Digikam::DPopupFrame	1349
Digikam::LightTableView	2532
Digikam::PanIconFrame	2814
Digikam::PlaceholderWidget	2846
Digikam::StatusBarProgressWidget	3264
QGraphicsItem	
Digikam::DSelectionItem	1398
QGraphicsObject	
Digikam::ClickDragReleaseItem	601
Digikam::DImgChildItem	1093
Digikam::RegionFrameItem	2961
Digikam::FacelItem	1560
Digikam::FocusPointItem	1734
Digikam::RubberItem	2992
Digikam::GraphicsDImgItem	1852
QGraphicsView	
Digikam::DPreviewImage	1352
Digikam::GraphicsDImgView	1854
Digikam::Canvas	562
Digikam::ImageRegionWidget	1992
Digikam::ImportPreviewView	2091
Digikam::ItemPreviewView	2361
Digikam::LightTablePreview	2522
QGroupBox	
Digikam::FullScreenSettings	1752
QHash	
Digikam::DatabaseFields::Hash< QVariant >	778
Digikam::DatabaseFields::Hash< T >	778
QHBoxLayout	
Digikam::RadioButtonHBox	2904
QItemDelegate	
Digikam::GPSItemDelegate	1806
Digikam::RatingComboBoxDelegate	2919

Digikam::TableViewItemDelegate . . . . .	3322
QItemSelectionMode	
Digikam::GPSLinkItemSelectionMode . . . . .	1817
QLabel	
Digikam::DActiveLabel . . . . .	768
Digikam::DAdjustableLabel . . . . .	769
Digikam::DSqueezedClickLabel . . . . .	1408
Digikam::DTextLabelName . . . . .	1416
Digikam::DTextLabelValue . . . . .	1418
Digikam::DClickLabel . . . . .	896
Digikam::DCursorTracker . . . . .	965
Digikam::DItemToolTip . . . . .	1162
Digikam::BlackFrameToolTip . . . . .	501
Digikam::FreeSpaceToolTip . . . . .	1748
Digikam::ImageDialogToolTip . . . . .	1961
Digikam::ItemViewToolTip . . . . .	2467
Digikam::QueueToolTip . . . . .	2903
ShowFoto::ShowfotoFolderViewToolTip . . . . .	3689
ShowFoto::ShowfotoStackViewToolTip . . . . .	3744
Digikam::DTextEditClearButton . . . . .	1415
Digikam::EffectPreview . . . . .	1491
Digikam::TransitionPreview . . . . .	3541
Digikam::WorkingWidget . . . . .	3624
QLayout	
Digikam::DynamicLayout . . . . .	1451
QLineEdit	
Digikam::AddTagsLineEdit . . . . .	233
Digikam::DatePickerYearSelector . . . . .	820
Digikam::ProxyLineEdit . . . . .	2887
Digikam::ProxyClickLineEdit . . . . .	2885
Digikam::SearchTextBar . . . . .	3123
QList	
Digikam::AlbumPointerList< T > . . . . .	322
Digikam::FacePipelineFaceTagsIfaceList . . . . .	1594
Digikam::FileActionItemInfoList . . . . .	1664
Digikam::ItemInfoTaskSplitter . . . . .	2317
Digikam::ItemInfoList . . . . .	2314
Digikam::PackageLoadingDescriptionList . . . . .	2809
QListView	
Digikam::DCategorizedView . . . . .	882
Digikam::ActionCategorizedView . . . . .	205
Digikam::ItemViewCategorized . . . . .	2442
Digikam::NamespaceListView . . . . .	2756
QListWidget	
Digikam::DTextList . . . . .	1419
Digikam::PreviewList . . . . .	2849
QListWidgetItem	
Digikam::FilmContainer::ListItem . . . . .	1692
Digikam::PreviewListItem . . . . .	2850
QMainWindow	
Digikam::TagsManager . . . . .	3416
QMap	
Digikam::QMapForAdaptors< Vertex, Vertex > . . . . .	2891
Digikam::QMapForAdaptors< Vertex, int > . . . . .	2891
Digikam::CaptionsMap . . . . .	570
Digikam::QMapForAdaptors< Key, Value > . . . . .	2891
QMenu	
Digikam::ColorLabelMenuAction . . . . .	644

Digikam::DDatePickerPopup	976
Digikam::lccProfilesMenuAction	1929
Digikam::ModelMenu	2737
Digikam::BookmarksMenu	517
Digikam::PickLabelMenuAction	2842
Digikam::RatingMenuAction	2928
Digikam::TagsPopupMenu	3422
QMimeData	
Digikam::DAlbumDrag	772
Digikam::DCameraDragObject	875
Digikam::DCameraItemListDrag	876
Digikam::DItemDrag	1154
Digikam::DTagListDrag	1409
Digikam::MapDragData	2612
QObject	
Digikam::AbstractDetector	186
Digikam::AestheticDetector	250
Digikam::BlurDetector	503
Digikam::CompressionDetector	653
Digikam::ExposureDetector	1538
Digikam::NoiseDetector	2768
Digikam::AbstractItemDragDropHandler	187
Digikam::ImportDragDropHandler	2046
Digikam::ItemDragDropHandler	2226
ShowFoto::ShowfotoDragDropHandler	3670
Digikam::AbstractMarkerTiler	190
Digikam::GPSTiler	1818
Digikam::ItemMarkerTiler	2339
Digikam::ActionJob	211
Digikam::ActionTask	215
Digikam::AutotagsAssignmentTask	432
Digikam::DBJob	857
Digikam::AlbumsJob	353
Digikam::DatesJob	825
Digikam::GPSJob	1815
Digikam::SearchesJob	3035
Digikam::TagsJob	3410
Digikam::DatabaseTask	792
Digikam::FingerprintsTask	1727
Digikam::IOJob	2154
Digikam::BuildTrashCountersJob	530
Digikam::CopyOrMoveJob	671
Digikam::DTrashItemsListingJob	1426
Digikam::DeleteJob	1005
Digikam::EmptyDTrashItemsJob	1502
Digikam::RenameFileJob	2970
Digikam::RestoreDTrashItemsJob	2975
Digikam::ImageQualityTask	1987
Digikam::MetadataRemoveTask	2654
Digikam::MetadataSyncTask	2663
Digikam::ThumbsTask	3498
Digikam::VidSlideTask	3598
Digikam::AdvancedRenameManager	241
Digikam::Akonadiflface	251
Digikam::AlbumHistory	278
Digikam::AlbumLabelsSearchHandler	281
Digikam::AlbumManager	283
Digikam::AlbumModelDragDropHandler	313

Digikam::AlbumDragDropHandler . . . . .	264
Digikam::TagDragDropHandler . . . . .	3340
Digikam::AlbumModificationHelper . . . . .	315
Digikam::AlbumThumbnailLoader . . . . .	355
Digikam::AlbumWatch . . . . .	368
Digikam::ApplicationSettings . . . . .	382
Digikam::BalooWrap . . . . .	459
Digikam::BatchTool . . . . .	463
Digikam::BatchToolsFactory . . . . .	473
Digikam::BdEngineBackend . . . . .	479
Digikam::CoreDbBackend . . . . .	725
Digikam::FaceDbBackend . . . . .	1548
Digikam::SimilarityDbBackend . . . . .	3236
Digikam::ThumbsDbBackend . . . . .	3490
Digikam::BlackFrameListViewItem . . . . .	499
Digikam::BlackFrameParser . . . . .	500
Digikam::BookmarkNode . . . . .	513
Digikam::BookmarksManager . . . . .	515
Digikam::CameraList . . . . .	549
Digikam::CameraThumbsCtrl . . . . .	554
Digikam::CollectionManager . . . . .	611
Digikam::CollectionScanner . . . . .	622
Digikam::ContextMenuHelper . . . . .	658
Digikam::CoreDbCopyManager . . . . .	732
Digikam::CoreDbWatch . . . . .	744
Digikam::DAboutData . . . . .	764
Digikam::DBJobsManager . . . . .	859
Digikam::DBinaryIface . . . . .	849
Digikam::ExifToolBinary . . . . .	1511
Digikam::FFmpegBinary . . . . .	1657
Digikam::MysqlAdminBinary . . . . .	2742
Digikam::MysqlInitBinary . . . . .	2745
Digikam::MysqlServerBinary . . . . .	2748
Digikam::MysqlUpgradeBinary . . . . .	2751
Digikam::DCategoryDrawer . . . . .	891
Digikam::ImportCategoryDrawer . . . . .	2020
Digikam::ItemCategoryDrawer . . . . .	2190
Digikam::DConfigDlgMgr . . . . .	919
Digikam::DConfigDlgWdgItem . . . . .	951
Digikam::DDateTable::Private . . . . .	984
Digikam::DIO . . . . .	1133
Digikam::DInfoInterface . . . . .	1126
Digikam::DBInfoIface . . . . .	852
Digikam::BqmInfoIface . . . . .	527
Digikam::DMetalInfoIface . . . . .	1192
ShowFoto::ShowfotoInfoIface . . . . .	3691
Digikam::DKCamera . . . . .	1164
Digikam::GPCamera . . . . .	1784
Digikam::UMSCamera . . . . .	3555
Digikam::DMetadataSettings . . . . .	1190
Digikam::DModelFactory . . . . .	1196
Digikam::DNNModelManager . . . . .	1233
Digikam::DNotificationWidget::Private . . . . .	1267
Digikam::DOnlineTranslator . . . . .	1268
Digikam::DOnlineTts . . . . .	1284
Digikam::DPlugin . . . . .	1298
Digikam::DPluginBqm . . . . .	1306
Digikam::DPluginDImg . . . . .	1322

Digikam::DPluginEditor . . . . .	1327
Digikam::DPluginGeneric . . . . .	1331
Digikam::DPluginRawImport . . . . .	1340
Digikam::DPluginLoader . . . . .	1334
Digikam::DRawDecoder . . . . .	1367
Digikam::DWItemDelegatePrivate . . . . .	1440
Digikam::DWorkingPixmap . . . . .	1443
Digikam::DatabaseServerStarter . . . . .	789
Digikam::DbEngineErrorHandler . . . . .	839
Digikam::DbEngineGuiErrorHandler . . . . .	841
Digikam::DbHeaderListItem . . . . .	848
Digikam::DigikamItemView::Private . . . . .	1066
Digikam::DisjointMetadata . . . . .	1137
Digikam::DynamicThread . . . . .	1452
Digikam::DImgThreadedFilter . . . . .	1114
Digikam::AntiVignettingFilter . . . . .	378
Digikam::AutoLevelsFilter . . . . .	423
Digikam::BCGFilter . . . . .	475
Digikam::BWSepiaFilter . . . . .	533
Digikam::BlurFXFilter . . . . .	509
Digikam::BlurFilter . . . . .	505
Digikam::BorderFilter . . . . .	522
Digikam::CBFilter . . . . .	578
Digikam::CharcoalFilter . . . . .	584
Digikam::ColorFXFilter . . . . .	637
Digikam::ContentAwareFilter . . . . .	655
Digikam::CurvesFilter . . . . .	751
Digikam::DImgThreadedAnalyser . . . . .	1111
Digikam::AutoCrop . . . . .	415
Digikam::NREstimate . . . . .	2782
Digikam::DistortionFXFilter . . . . .	1147
Digikam::EmbossFilter . . . . .	1498
Digikam::EqualizeFilter . . . . .	1506
Digikam::FilmFilter . . . . .	1694
Digikam::FilmGrainFilter . . . . .	1698
Digikam::FilterActionFilter . . . . .	1708
Digikam::FreeRotationFilter . . . . .	1743
Digikam::GreycstorationFilter . . . . .	1858
Digikam::HSLFilter . . . . .	1908
Digikam::HotPixelFixer . . . . .	1898
Digikam::IccTransformFilter . . . . .	1945
Digikam::InfraredFilter . . . . .	2143
Digikam::InvertFilter . . . . .	2150
Digikam::LensDistortionFilter . . . . .	2503
Digikam::LensFunFilter . . . . .	2509
Digikam::LevelsFilter . . . . .	2516
Digikam::LocalContrastFilter . . . . .	2574
Digikam::MixerFilter . . . . .	2720
Digikam::NRFilter . . . . .	2786
Digikam::NormalizeFilter . . . . .	2771
Digikam::OilPaintFilter . . . . .	2791
Digikam::RainDropFilter . . . . .	2905
Digikam::RawProcessingFilter . . . . .	2935
Digikam::RedEyeCorrectionFilter . . . . .	2952
Digikam::RefocusFilter . . . . .	2957
Digikam::SharpenFilter . . . . .	3201
Digikam::ShearFilter . . . . .	3206
Digikam::StretchFilter . . . . .	3270

Digikam::TextureFilter . . . . .	3452
Digikam::TonalityFilter . . . . .	3513
Digikam::UnsharpMaskFilter . . . . .	3570
Digikam::WBFilter . . . . .	3606
Digikam::AutoExpoFilter . . . . .	419
Digikam::ImageHistogram . . . . .	1965
Digikam::LoadSaveThread . . . . .	2567
Digikam::ScanStateFilter . . . . .	3018
Digikam::EditorCore . . . . .	1465
Digikam::EditorTool . . . . .	1472
Digikam::EditorToolThreaded . . . . .	1478
Digikam::EditorToolface . . . . .	1475
Digikam::ExifToolListViewGroup . . . . .	1516
Digikam::ExifToolParser . . . . .	1519
Digikam::ExifToolProcess::Private . . . . .	1530
Digikam::FaceGroup . . . . .	1555
Digikam::FacePipeline . . . . .	1564
Digikam::FacePipeline::Private . . . . .	1570
Digikam::FaceUtils . . . . .	1648
Digikam::FileActionMngr . . . . .	1666
Digikam::FileActionMngr::Private . . . . .	1668
Digikam::FileActionProgressItemContainer . . . . .	1680
Digikam::FocusPointGroup . . . . .	1731
Digikam::FocusPointsExtractor . . . . .	1737
Digikam::FocusPointsWriter . . . . .	1738
Digikam::GPSBookmarkOwner . . . . .	1793
Digikam::GPSItemInfoSorter . . . . .	1808
Digikam::GPSItemListContextMenu . . . . .	1810
Digikam::GPSModelIndexProxyMapper . . . . .	1825
Digikam::GeoDragDropHandler . . . . .	1770
Digikam::MapDragDropHandler . . . . .	2613
Digikam::GeofaceGlobalObject . . . . .	1771
Digikam::GeoModelHelper . . . . .	1780
Digikam::GPSBookmarkModelHelper . . . . .	1790
Digikam::GPSGeofaceModelHelper . . . . .	1800
Digikam::ItemGPSModelHelper . . . . .	2272
Digikam::MapViewModelHelper . . . . .	2615
Digikam::GeolocationSettings . . . . .	1777
Digikam::GreycstorationSettings . . . . .	1863
Digikam::HistogramPainter . . . . .	1886
Digikam::IOJobsManager . . . . .	2157
Digikam::IccSettings . . . . .	1937
Digikam::ImageDialog . . . . .	1958
Digikam::ImageDialogIconProvider . . . . .	1959
Digikam::ImageQualityParser . . . . .	1980
Digikam::ImageQualityThreadPool . . . . .	1989
Digikam::ImportContextMenuHelper . . . . .	2022
Digikam::ImportIconView::Private . . . . .	2064
Digikam::ImportSettings . . . . .	2106
Digikam::ItemAttributesWatch . . . . .	2180
Digikam::ItemDelegateOverlay . . . . .	2217
Digikam::AbstractWidgetDelegateOverlay . . . . .	200
Digikam::GroupIndicatorOverlay . . . . .	1865
Digikam::HoverButtonDelegateOverlay . . . . .	1904
Digikam::ActionVersionsOverlay . . . . .	222
Digikam::FaceRejectionOverlay . . . . .	1616
Digikam::ImportRotateOverlay . . . . .	2099
Digikam::ItemFullScreenOverlay . . . . .	2262



Digikam::ItemRotateOverlay . . . . .	2389
Digikam::ItemSelectionOverlay . . . . .	2403
Digikam::ShowHideVersionsOverlay . . . . .	3210
Digikam::ImportCoordinatesOverlay . . . . .	2028
Digikam::ImportDownloadOverlay . . . . .	2042
Digikam::ImportLockOverlay . . . . .	2079
Digikam::ImportRatingOverlay . . . . .	2094
Digikam::ItemCoordinatesOverlay . . . . .	2199
Digikam::ItemRatingOverlay . . . . .	2385
Digikam::PersistentWidgetDelegateOverlay . . . . .	2834
Digikam::AssignNameOverlay . . . . .	400
Digikam::TagsLineEditOverlay . . . . .	3412
ShowFoto::ShowfotoCoordinatesOverlay . . . . .	3657
Digikam::ItemInfoAlbumsJob . . . . .	2308
Digikam::ItemInfoCache . . . . .	2309
Digikam::ItemInfoJob . . . . .	2313
Digikam::ItemListDragDropHandler . . . . .	2320
Digikam::GPSItemListDragDropHandler . . . . .	1812
Digikam::ItemSortCollator . . . . .	2412
Digikam::ItemViewUtilities . . . . .	2469
Digikam::ItemVisibilityController . . . . .	2471
Digikam::HidingStateChanger . . . . .	1881
Digikam::AssignNameWidgetStates . . . . .	411
Digikam::ItemVisibilityControllerPropertyObject . . . . .	2476
Digikam::AnimatedVisibility . . . . .	376
Digikam::ListItem . . . . .	2539
Digikam::LoadingCache . . . . .	2544
Digikam::LoadingCacheFileWatch . . . . .	2549
Digikam::ScanControllerLoadingCacheFileWatch . . . . .	3017
Digikam::LocalizeSettings . . . . .	2582
Digikam::LookupAltitude . . . . .	2584
Digikam::LookupAltitudeGeonames . . . . .	2586
Digikam::MLPipelineFoundation . . . . .	2727
Digikam::FacePipelineBase . . . . .	1572
Digikam::FacePipelineDetect . . . . .	1576
Digikam::FacePipelineDetectRecognize . . . . .	1580
Digikam::FacePipelineEdit . . . . .	1584
Digikam::FacePipelineRecognize . . . . .	1598
Digikam::FacePipelineReset . . . . .	1602
Digikam::FacePipelineRetrain . . . . .	1606
Digikam::MaintenanceMngr . . . . .	2591
Digikam::MapBackend . . . . .	2610
Digikam::BackendGoogleMaps . . . . .	440
Digikam::BackendMarble . . . . .	447
Digikam::MdKeyListViewItem . . . . .	2631
Digikam::MetaEngineSettings . . . . .	2712
Digikam::MetadataHubMngr . . . . .	2639
Digikam::MetadataPanel . . . . .	2648
Digikam::NetworkManager . . . . .	2757
Digikam::OnlineVersionChecker . . . . .	2794
Digikam::OnlineVersionDwnl . . . . .	2797
Digikam::ParallelPipes . . . . .	2822
Digikam::PreviewThreadWrapper . . . . .	2862
Digikam::PrivateProgressItemCreator . . . . .	2864
Digikam::ProgressItem . . . . .	2868
Digikam::AlbumParser . . . . .	319
Digikam::FileActionProgress . . . . .	1677

Digikam::MaintenanceTool	2597
Digikam::AutotagsAssignment	428
Digikam::DbCleaner	831
Digikam::DuplicatesFinder	1428
Digikam::FacesDetector	1629
Digikam::FacesEngine	1632
Digikam::FingerPrintsGenerator	1723
Digikam::ImageQualitySorter	1983
Digikam::MetadataRemover	2650
Digikam::MetadataSynchronizer	2659
Digikam::NewItemFinder	2759
Digikam::ThumbsGenerator	3495
Digikam::ProgressManager	2875
Digikam::RGBBackend	2977
Digikam::BackendGeonamesRG	433
Digikam::BackendGeonamesUSRG	436
Digikam::BackendOsmRG	455
Digikam::Rule	2995
Digikam::Modifier	2739
Digikam::CaseModifier	573
Digikam::DefaultValueModifier	996
Digikam::RangeModifier	2912
Digikam::RemoveDoublesModifier	2966
Digikam::ReplaceModifier	2973
Digikam::TrimmedModifier	3551
Digikam::UniqueModifier	3567
Digikam::Option	2805
Digikam::CameraNameOption	551
Digikam::DatabaseOption	782
Digikam::DateOption	816
Digikam::DirectoryNameOption	1135
Digikam::FilePropertiesOption	1682
Digikam::MetadataOption	2644
Digikam::SequenceNumberOption	3156
Digikam::SearchField	3036
Digikam::SearchModificationHelper	3114
Digikam::SetupRaw	3188
Digikam::SinglePhotoPreviewLayout	3242
Digikam::SyncJob	3279
Digikam::TableViewColumn	3285
Digikam::TableViewColumns::ColumnAudioVideoProperties	3291
Digikam::TableViewColumns::ColumnDigikamProperties	3295
Digikam::TableViewColumns::ColumnFileProperties	3300
Digikam::TableViewColumns::ColumnGeoProperties	3305
Digikam::TableViewColumns::ColumnItemProperties	3309
Digikam::TableViewColumns::ColumnPhotoProperties	3314
Digikam::TableViewColumns::ColumnThumbnail	3318
Digikam::TableViewColumnFactory	3289
Digikam::TableViewSelectionModelSyncer	3326
Digikam::TagModificationHelper	3379
Digikam::TagsActionMgr	3396
Digikam::TagsCache	3398
Digikam::TemplateManager	3444
Digikam::ThemeManager	3456
Digikam::ThreadManager	3458
Digikam::ThumbnailImageCatcher	3467
Digikam::TileGroupier	3500
Digikam::Token	3510

Digikam::TrackCorrelator	3523
Digikam::TrackManager	3527
Digikam::TrackReader	3529
Digikam::VisibilityController	3602
Digikam::WSSettings	3630
Digikam::WorkerObject	3616
Digikam::DatabaseWorkerInterface	796
Digikam::FileActionMngrDatabaseWorker	1670
Digikam::DatabaseWriter	799
Digikam::DetectionBenchmark	1009
Digikam::DetectionWorker	1012
Digikam::FileWorkerInterface	1689
Digikam::ParallelAdapter< Digikam::FileWorkerInterface >	2819
Digikam::FileActionMngrFileWorker	1674
Digikam::ItemFilterModelWorker	2255
Digikam::ItemFilterModelFilterer	2249
Digikam::ItemFilterModelPreparer	2252
Digikam::RecognitionBenchmark	2941
Digikam::RecognitionWorker	2947
Digikam::TrainerWorker	3531
Digikam::WorkflowManager	3623
ShowFoto::ShowfotoKineticScroller	3712
ShowFoto::ShowfotoSettings	3720
QPlainTextEdit	
Digikam::AdvancedRenameLineEdit	239
Digikam::DPlainTextEdit	1293
QProcess	
Digikam::ExifToolProcess	1525
QProgressBar	
Digikam::DProgressWdg	1359
QProgressDialog	
Digikam::DBusyDlg	873
QPushButton	
Digikam::ColorLabelSelector	645
Digikam::CtrlButton	747
Digikam::DColorSelector	902
Digikam::DMultiTabBarButton	1203
Digikam::DMultiTabBarTab	1207
Digikam::DetByClockPhotoButton	1008
Digikam::PickLabelSelector	2843
QReadLocker	
Digikam::ItemInfoReadLocker	2315
QRunnable	
Digikam::ActionJob	211
QScrollArea	
Digikam::DExpanderBox	1014
Digikam::DExpanderBoxExclusive	1017
Digikam::DRawDecoderWidget	1382
Digikam::ImportItemPropertiesTab	2077
Digikam::ItemPropertiesTab	2377
Digikam::ItemSelectionPropertiesTab	2410
Digikam::TemplateViewer	3448
Digikam::EditorToolSettings	1476
Digikam::FuzzySearchView	1757
Digikam::ICCPreviewWidget	1921
Digikam::ImageDialogPreview	1960
Digikam::SetupAlbumView	3162
Digikam::SetupCamera	3163

Digikam::SetupCategory	3164
Digikam::SetupCollections	3173
Digikam::SetupDatabase	3175
Digikam::SetupEditor	3176
Digikam::SetupEditorIface	3177
Digikam::SetupGeolocation	3178
Digikam::SetupICC	3179
Digikam::SetupIOFiles	3181
Digikam::SetupImageQualitySorter	3180
Digikam::SetupLightTable	3181
Digikam::SetupMetadata	3182
Digikam::SetupMime	3184
Digikam::SetupMisc	3185
Digikam::SetupPlugins	3187
Digikam::SetupTemplate	3189
Digikam::SetupToolTip	3190
Digikam::SetupVersioning	3191
Digikam::SubjectWidget	3277
Digikam::SubjectEdit	3275
Digikam::TagsEdit	3409
Digikam::TimeAdjustSettings	3503
Digikam::TransactionItemView	3538
ShowFoto::ShowfotoSetupMetadata	3725
ShowFoto::ShowfotoSetupMisc	3726
ShowFoto::ShowfotoSetupPlugins	3727
ShowFoto::ShowfotoSetupRaw	3728
ShowFoto::ShowfotoSetupToolTip	3729
QSharedData	
Digikam::FacePipelineExtendedPackage	1589
Digikam::GeofaceSharedData	1774
Digikam::ItemHistoryGraphData	2279
Digikam::ItemInfoData	2311
Digikam::MLPipelinePackageNotify	2732
Digikam::MetaEngineData::Private	2706
Digikam::TwoProgressItemsContainer	3553
Digikam::FileActionProgressItemContainer	1680
QSortFilterProxyModel	
Digikam::AddBookmarkProxyModel	226
Digikam::AlbumFilterModel	267
Digikam::CheckableAlbumFilterModel	588
Digikam::SearchFilterModel	3098
Digikam::TagPropertiesFilterModel	3388
Digikam::TagsManagerFilterModel	3419
Digikam::DCategorizedSortFilterProxyModel	877
Digikam::ActionSortFilterProxyModel	213
Digikam::ImageSortFilterModel	1995
Digikam::ItemFilterModel	2239
Digikam::NoDuplicatesItemFilterModel	2766
Digikam::ImportSortFilterModel	2109
Digikam::ImportFilterModel	2051
Digikam::NoDuplicatesImportFilterModel	2763
ShowFoto::ShowfotoSortFilterModel	3730
ShowFoto::NoDuplicatesShowfotoFilterModel	3640
ShowFoto::ShowfotoFilterModel	3673
Digikam::GPSItemSortProxyModel	1814
Digikam::TreeProxyModel	3544
QSpinBox	
Digikam::CustomStepsIntSpinBox	762

QSplashScreen	
Digikam::DSplashScreen	1407
QSplitter	
Digikam::SidebarSplitter	3221
QSqlQuery	
Digikam::DbEngineSqlQuery	847
QStackedWidget	
Digikam::DConfigDlgStackedWidget	928
Digikam::DPreviewManager	1356
Digikam::EditorStackView	1470
Digikam::ExifToolWidget	1533
Digikam::FileSaveOptionsBox	1686
Digikam::ImportStackedView	2112
Digikam::MediaPlayerView	2632
Digikam::StackedView	3256
Digikam::StatusProgressBar	3265
Digikam::ToolSettingsView	3518
QStandardItemModel	
Digikam::CategorizedItemModel	576
Digikam::ActionItemModel	208
QStyledItemDelegate	
Digikam::AlbumTreeViewDelegate	364
Digikam::ItemFiltersHistoryItemDelegate	2259
Digikam::ThumbnailAligningDelegate	3461
Digikam::VersionsDelegate	3582
QSyntaxHighlighter	
Digikam::Highlighter	1884
QTabBar	
Digikam::QueuePoolBar	2901
QTabWidget	
Digikam::AlbumSelectTabs	344
Digikam::FaceScanWidget	1626
Digikam::ItemPropertiesColorsTab	2364
Digikam::ItemPropertiesMetadataTab	2367
Digikam::ItemPropertiesVersionsTab	2382
Digikam::QueuePool	2899
Digikam::QueueSettingsView	2902
Digikam::TemplatePanel	3445
Digikam::ToolsView	3520
QTemporaryFile	
Digikam::SafeTemporaryFile	3001
QTextBrowser	
Digikam::DTextBrowser	1410
QTextEdit	
Digikam::DTextEdit	1411
QThread	
Digikam::ActionThreadBase	219
Digikam::ActionThread	217
Digikam::DBJobsThread	862
Digikam::AlbumsDBJobsThread	328
Digikam::DatesDBJobsThread	823
Digikam::GPSDBJobsThread	1798
Digikam::SearchesDBJobsThread	3032
Digikam::TagsDBJobsThread	3407
Digikam::IOJobsThread	2159
Digikam::MaintenanceThread	2594
Digikam::VidSlideThread	3600
Digikam::CameraController	540

Digikam::CameraHistoryUpdater . . . . .	545
Digikam::DBusThread . . . . .	874
Digikam::CameraAutoDetectThread . . . . .	539
Digikam::DatabaseCopyThread . . . . .	776
Digikam::DatabaseServer . . . . .	787
Digikam::DbEngineConnectionChecker . . . . .	837
Digikam::ExifToolThread . . . . .	1532
Digikam::ImageQualityThread . . . . .	1988
Digikam::ProcessLauncher . . . . .	2866
Digikam::FFmpegLauncher . . . . .	1661
Digikam::ScanController . . . . .	3009
Digikam::TrackCorrelatorThread . . . . .	3525
QTreeView	
Digikam::AbstractAlbumTreeView . . . . .	147
Digikam::GPSItemList . . . . .	1809
Digikam::SetupCollectionTreeView . . . . .	3174
Digikam::TableViewTreeView . . . . .	3328
Digikam::TagMngrListView . . . . .	3365
Digikam::VersionsTreeView . . . . .	3585
ShowFoto::ShowfotoFolderViewList . . . . .	3684
QTreeWidget	
Digikam::AssignedListView . . . . .	397
Digikam::BlackFrameListView . . . . .	498
Digikam::CameraFolderView . . . . .	544
Digikam::CameraItem . . . . .	548
Digikam::DBinarySearch . . . . .	851
Digikam::DHistoryView . . . . .	1037
Digikam::DItemsListView . . . . .	1160
Digikam::DPluginConfView . . . . .	1310
Digikam::DPluginConfViewBqm . . . . .	1313
Digikam::DPluginConfViewDImg . . . . .	1315
Digikam::DPluginConfViewEditor . . . . .	1317
Digikam::DPluginConfViewGeneric . . . . .	1319
Digikam::DbKeySelector . . . . .	868
Digikam::DeleteItemList . . . . .	1004
Digikam::ExifToolListView . . . . .	1515
Digikam::FindDuplicatesAlbum . . . . .	1719
Digikam::LabelsTreeView . . . . .	2498
Digikam::LanguagesList . . . . .	2501
Digikam::MetadataListView . . . . .	2642
Digikam::MetadataSelector . . . . .	2655
Digikam::QueueListView . . . . .	2892
Digikam::TemplateList . . . . .	3442
Digikam::ToolsListView . . . . .	3519
Digikam::WorkflowList . . . . .	3622
ShowFoto::ShowfotoFolderViewBookmarkList . . . . .	3682
ShowFoto::ShowfotoStackViewFavoriteList . . . . .	3736
ShowFoto::ShowfotoStackViewList . . . . .	3740
QTreeWidgetItem	
Digikam::AdvancedRenameListItem . . . . .	240
Digikam::AssignedListViewItem . . . . .	398
Digikam::BlackFrameListViewItem . . . . .	499
Digikam::CameraFolderItem . . . . .	543
Digikam::CameraItem . . . . .	547
Digikam::DItemsListViewItem . . . . .	1161
Digikam::DbHeaderListItem . . . . .	848
Digikam::DbKeySelectorItem . . . . .	869
Digikam::DeleteItem . . . . .	1003

Digikam::ExifToolListViewGroup . . . . .	1516
Digikam::ExifToolListViewItem . . . . .	1517
Digikam::FindDuplicatesAlbumItem . . . . .	1720
Digikam::MdKeyListViewItem . . . . .	2631
Digikam::MetadataListViewItem . . . . .	2643
Digikam::MetadataSelectorItem . . . . .	2656
Digikam::QueueListViewItem . . . . .	2894
Digikam::TemplateListItem . . . . .	3443
Digikam::ToolListViewGroup . . . . .	3516
Digikam::ToolListViewItem . . . . .	3517
Digikam::WorkflowItem . . . . .	3621
ShowFoto::ShowfotoFolderViewBookmarkItem . . . . .	3681
ShowFoto::ShowfotoStackViewFavoriteItem . . . . .	3733
ShowFoto::ShowfotoStackViewItem . . . . .	3739
QUndoCommand	
Digikam::ChangeBookmarkCommand . . . . .	582
Digikam::GPSUndoCommand . . . . .	1835
Digikam::RemoveBookmarksCommand . . . . .	2964
Digikam::InsertBookmarksCommand . . . . .	2148
ShowFoto::ShowfotoFolderViewUndo . . . . .	3690
QUrl	
Digikam::CoreDbUrl . . . . .	737
QValidator	
Digikam::DatePickerValidator . . . . .	819
QVector	
Digikam::SparseModelIndexVector . . . . .	3250
QWebEnginePage	
Digikam::HTMLWidgetPage . . . . .	1914
Digikam::WelcomePageViewPage . . . . .	3615
QWebEngineView	
Digikam::HTMLWidget . . . . .	1913
Digikam::WebWidget . . . . .	3612
Digikam::WelcomePageView . . . . .	3614
QWebView	
Digikam::WebWidget . . . . .	3612
QWidget	
Digikam::AbstractSearchGroupContainer . . . . .	195
Digikam::SearchGroup . . . . .	3102
Digikam::SearchView . . . . .	3137
Digikam::AdvancedMetadataTab . . . . .	236
Digikam::AdvancedRenameWidget . . . . .	245
Digikam::AdvancedSettings . . . . .	249
Digikam::AlbumCustomizer . . . . .	263
Digikam::AlbumSelectWidget . . . . .	351
Digikam::AlbumSelectors . . . . .	340
Digikam::AltLangStrEdit . . . . .	369
Digikam::AnimatedClearButton . . . . .	374
Digikam::AntiVignettingSettings . . . . .	381
Digikam::AudPlayerWdg . . . . .	414
Digikam::BCGSettings . . . . .	478
Digikam::BWSepiaSettings . . . . .	537
Digikam::BorderSettings . . . . .	526
Digikam::CBSettings . . . . .	581
Digikam::CIE TongueWidget . . . . .	600
Digikam::CaptureWidget . . . . .	572
Digikam::ColorFXSettings . . . . .	640
Digikam::ColorGradientWidget . . . . .	641
Digikam::CurvesBox . . . . .	748

Digikam::CurvesSettings . . . . .	755
Digikam::CurvesWidget . . . . .	757
Digikam::DAbstractSliderSpinBox . . . . .	766
Digikam::DDoubleSliderSpinBox . . . . .	991
Digikam::DSliderSpinBox . . . . .	1404
Digikam::DArrowClickLabel . . . . .	774
Digikam::DComboBox . . . . .	908
Digikam::DConfigDlgTitle . . . . .	929
Digikam::DConfigDlgView . . . . .	936
Digikam::DConfigDlgWdg . . . . .	943
Digikam::DDateTable . . . . .	979
Digikam::DDoubleNumInput . . . . .	989
Digikam::DFontProperties . . . . .	1026
Digikam::DGradientSlider . . . . .	1035
Digikam::DImgLoaderSettings . . . . .	1106
Digikam::DIntNumInput . . . . .	1130
Digikam::DIntRangeBox . . . . .	1131
Digikam::DItemsList . . . . .	1157
Digikam::DLabelExpander . . . . .	1167
Digikam::DMultiTabBar . . . . .	1198
Digikam::Sidebar . . . . .	3214
Digikam::DNGConvertSettings . . . . .	1210
Digikam::DNGSettings . . . . .	1211
Digikam::DPluginSetup . . . . .	1343
Digikam::DPointSelect . . . . .	1344
Digikam::DHueSaturationSelector . . . . .	1038
Digikam::DSaveSettingsWidget . . . . .	1396
Digikam::DatabaseSettingsWidget . . . . .	790
Digikam::DbKeySelectorView . . . . .	870
Digikam::DeleteWidget . . . . .	1006
Digikam::DragHandle . . . . .	1366
Digikam::ExifToolConfPanel . . . . .	1513
Digikam::ExifToolErrorView . . . . .	1514
Digikam::ExifToolLoadingView . . . . .	1518
Digikam::FileSaveConflictBox . . . . .	1685
Digikam::FilmGrainSettings . . . . .	1702
Digikam::FilterStatusBar . . . . .	1718
Digikam::FiltersHistoryWidget . . . . .	1713
Digikam::FindDuplicatesView . . . . .	1721
Digikam::FrameOsdWidget . . . . .	1741
Digikam::FreeRotationSettings . . . . .	1746
Digikam::FreeSpaceWidget . . . . .	1750
Digikam::GPSCorrelatorWidget . . . . .	1794
Digikam::GPSSearchView . . . . .	1832
Digikam::HSLSettings . . . . .	1911
Digikam::HSPreviewWidget . . . . .	1912
Digikam::HistogramBox . . . . .	1885
Digikam::HistogramWidget . . . . .	1891
Digikam::HotPixelSettings . . . . .	1902
Digikam::ImageGuideWidget . . . . .	1963
Digikam::ImageQualityConfSelector . . . . .	1978
Digikam::ImageQualitySettings . . . . .	1982
Digikam::ItemPropertiesGPSTab . . . . .	2365
Digikam::ItemPropertiesHistoryTab . . . . .	2366
Digikam::LensFunCameraSelector . . . . .	2507
Digikam::LensFunSettings . . . . .	2514
Digikam::LocalContrastSettings . . . . .	2577
Digikam::LocalizeConfig . . . . .	2578



Digikam::LocalizeSelectorList	2581
Digikam::MapWidget	2618
Digikam::MapWidgetView	2626
Digikam::MetadataSelectorView	2657
Digikam::MetadataStatusBar	2658
Digikam::MetadataWidget	2665
Digikam::ExifWidget	1535
Digikam::ICCProfileWidget	1933
Digikam::IptcWidget	2165
Digikam::MakerNoteWidget	2601
Digikam::XmpWidget	3637
Digikam::MixerSettings	2724
Digikam::MonthWidget	2741
Digikam::NRSettings	2790
Digikam::PanIconWidget	2817
Digikam::PreviewToolBar	2863
Digikam::RGWidget	2988
Digikam::RatingWidget	2930
Digikam::RatingComboBoxWidget	2921
Digikam::RatingFilterWidget	2926
Digikam::RedEyeCorrectionSettings	2956
Digikam::RenameCustomizer	2969
Digikam::ScriptingSettings	3028
Digikam::SearchFieldGroup	3057
Digikam::SearchFieldGroupLabel	3058
Digikam::SearchGroupLabel	3105
Digikam::SearchTabHeader	3122
Digikam::SearchViewBottomBar	3140
Digikam::SearchWindow	3142
Digikam::SharpSettings	3205
Digikam::SidebarWidget	3224
Digikam::SketchWidget	3245
Digikam::SlideVideo	3247
Digikam::SpellCheckConfig	3251
Digikam::StyleSheetDebugger	3273
Digikam::SystemSettingsWidget	3281
Digikam::TableView	3282
Digikam::TableViewColumnConfigurationWidget	3288
Digikam::TableViewColumns::ColumnFileConfigurationWidget	3298
Digikam::TableViewColumns::ColumnGeoConfigurationWidget	3303
Digikam::TableViewColumns::ColumnPhotoConfigurationWidget	3312
Digikam::TagList	3362
Digikam::TagPropWidget	3392
Digikam::TextureSettings	3455
Digikam::TimeLineWidget	3507
Digikam::TrashView	3542
Digikam::VersionsWidget	3587
Digikam::WBSettings	3610
Digikam::WSSettingsWidget	3632
ShowFoto::ShowfotoFolderViewBookmarks	3683
ShowFoto::ShowfotoFolderViewSideBar	3686
ShowFoto::ShowfotoStackViewFavorites	3738
ShowFoto::ShowfotoStackViewSideBar	3742
QWidgetAction	
Digikam::DLogoAction	1169
QWizard	
Digikam::DWizardDlg	1441
Digikam::FirstRunDlg	1728

QWizardPage	
Digikam::DWizardPage	1442
Digikam::CollectionPage	620
Digikam::DatabasePage	786
Digikam::MetadataPage	2647
Digikam::MigrateFromDigikam4Page	2716
Digikam::OpenFilePage	2804
Digikam::PreviewPage	2860
Digikam::RawPage	2933
Digikam::StartScanPage	3258
Digikam::TooltipsPage	3522
Digikam::WelcomePage	3613
QWriteLocker	
Digikam::ItemInfoWriteLocker	2319
QXmlStreamReader	
Digikam::SearchXmlReader	3147
Digikam::KeywordSearchReader	2491
Digikam::SearchXmlCachingReader	3144
Digikam::XbelReader	3636
QXmlStreamWriter	
Digikam::SearchXmlWriter	3151
Digikam::KeywordSearchWriter	2493
Digikam::XbelWriter	3636
Digikam::SharedQueue< QString >	3199
ShowFoto::Showfoto::Private	3648
ShowFoto::ShowfotoItemInfo	3693
ShowFoto::ShowfotoItemSortSettings	3701
ShowFoto::ShowfotoItemViewDelegatePrivate	3710
ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate	3668
ShowFoto::ShowfotoNormalDelegatePrivate	3718
ShowFoto::ShowfotoThumbnailDelegatePrivate	3757

# Chapter 4

## Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CoreDbWatchAdaptor	139
Digikam::AbstractAlbumModel	140
Digikam::AbstractAlbumTreeView	147
Digikam::AbstractAlbumTreeView::ContextMenuElement	157
Digikam::AbstractAlbumTreeView::Private	158
Digikam::AbstractAlbumTreeViewSelectComboBox	159
Digikam::AbstractCheckableAlbumModel	163
Digikam::AbstractCheckableAlbumTreeView	171
Digikam::AbstractCountingAlbumModel	177
Digikam::AbstractCountingAlbumTreeView	183
Digikam::AbstractDetector	186
Digikam::AbstractItemDragDropHandler	187
Digikam::AbstractMarkerTiler	190
Digikam::AbstractMarkerTiler::ClickInfo	193
Digikam::AbstractMarkerTiler::NonEmptyIterator	193
Digikam::AbstractMarkerTiler::Tile	194
Digikam::AbstractSearchGroupContainer	195
Digikam::AbstractSpecificAlbumModel	197
Digikam::AbstractWidgetDelegateOverlay	200
Digikam::ActionCategorizedView	205
Digikam::ActionData	207
Digikam::ActionItemModel	208
Digikam::ActionJob	211
Digikam::ActionSortFilterProxyModel	213
Digikam::ActionTask	215
Digikam::ActionThread	217
Digikam::ActionThreadBase	219
Digikam::ActionVersionsOverlay	222
Digikam::AddBookmarkDialog	226
Digikam::AddBookmarkProxyModel	226
Digikam::AddTagsComboBox	228
Digikam::AddTagsLineEdit	233
Digikam::AdvancedMetadataTab	236
Digikam::AdvancedRenameDialog	237
Digikam::AdvancedRenameInput	238

Digikam::AdvancedRenameLineEdit	239
Digikam::AdvancedRenameListItem	240
Digikam::AdvancedRenameManager	241
Digikam::AdvancedRenameProcessDialog	243
Digikam::AdvancedRenameWidget	245
Digikam::AdvancedSettings	249
Digikam::AestheticDetector	250
Digikam::AkonadiIface	251
Digikam::Album	
Abstract base class for all album types	252
Digikam::AlbumChangeset	262
Digikam::AlbumCopyMoveHint	262
Digikam::AlbumCustomizer	263
Digikam::AlbumDragDropHandler	264
Digikam::AlbumFilterModel	267
Digikam::AlbumFolderViewSideBarWidget	274
Digikam::AlbumHistory	278
Digikam::AlbumInfo	280
Digikam::AlbumIterator	280
Digikam::AlbumLabelsSearchHandler	281
Digikam::AlbumManager	283
Digikam::AlbumManager::Private	305
Digikam::AlbumManagerCreator	306
Digikam::AlbumModel	307
Digikam::AlbumModelDragDropHandler	313
Digikam::AlbumModificationHelper	315
Digikam::AlbumParser	319
Digikam::AlbumPointer< T >	321
Digikam::AlbumPointerList< T >	322
Digikam::AlbumPropsEdit	323
Digikam::AlbumRootChangeset	324
Digikam::AlbumRootInfo	324
Digikam::AlbumRootLocation	325
Digikam::AlbumsDBJobInfo	327
Digikam::AlbumsDBJobsThread	328
Digikam::AlbumSelectComboBox	330
Digikam::AlbumSelectDialog	334
Digikam::AlbumSelectionTreeView	335
Digikam::AlbumSelectors	340
Digikam::AlbumSelectTabs	344
Digikam::AlbumSelectTreeView	345
Digikam::AlbumSelectWidget	351
Digikam::AlbumShortInfo	351
Digikam::AlbumSimplified	352
Digikam::AlbumsJob	353
Digikam::AlbumThumbnailLoader	355
Digikam::AlbumTreeView	360
Digikam::AlbumTreeViewDelegate	364
Digikam::AlbumTreeViewSelectComboBox	365
Digikam::AlbumWatch	368
Digikam::AltLangStrEdit	369
Digikam::AltLangStrEdit::Private	373
Digikam::AnimatedClearButton	374
Digikam::AnimatedVisibility	376
Digikam::AntiVignettingContainer	377
Digikam::AntiVignettingFilter	378
Digikam::AntiVignettingSettings	381
Digikam::ApplicationSettings	382

Digikam::ApplicationSettings::Private	391
Digikam::AssignedBatchTools	396
Digikam::AssignedListView	397
Digikam::AssignedListViewItem	398
Digikam::AssignNameOverlay	400
Digikam::AssignNameWidget	405
Digikam::AssignNameWidget::Private	409
Digikam::AssignNameWidgetStates	411
Digikam::AudPlayerWdg	414
Digikam::AutoCrop	415
Digikam::AutoExpoFilter	419
Digikam::AutoLevelsFilter	423
Digikam::AutoTagsAssign	426
Digikam::AutotagsAssignment	428
Digikam::AutotagsAssignmentTask	432
Digikam::BackendGeonamesRG	
This class calls Geonames' reverse geocoding service	433
Digikam::BackendGeonamesUSRG	
This class calls Geonames' get address service available only for USA locations	436
Digikam::BackendGoogleMaps	440
Digikam::BackendMarble	447
Digikam::BackendMarbleLayer	455
Digikam::BackendOsmRG	
This class calls Open Street Map's reverse geocoding service	455
Digikam::BalooInfo	458
Digikam::BalooWrap	
Singleton class which offer functionality for reading and writing image comment, tags and rating from Baloo to digiKam and from digiKam to Baloo	459
Digikam::BasicDImgFilterGenerator< T >	461
Digikam::BatchTool	463
Digikam::BatchToolSet	472
Digikam::BatchToolsFactory	473
Digikam::BCGContainer	474
Digikam::BCGFilter	475
Digikam::BCGSettings	478
Digikam::BdEngineBackend	479
Digikam::BdEngineBackend::QueryState	489
Digikam::BdEngineBackendPrivate	490
Digikam::BdEngineBackendPrivate::AbstractUnlocker	493
Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker	494
Digikam::BdEngineBackendPrivate::BusyWaiter	495
Digikam::BdEngineBackendPrivate::ErrorLocker	497
Digikam::BlackFrameListView	498
Digikam::BlackFrameListViewItem	499
Digikam::BlackFrameParser	500
Digikam::BlackFrameToolTip	501
Digikam::BlurDetector	503
Digikam::BlurFilter	505
Digikam::BlurFXFilter	509
Digikam::BookmarkNode	513
Digikam::BookmarksDialog	514
Digikam::BookmarksManager	515
Digikam::BookmarksMenu	517
Digikam::BookmarksModel	519
Digikam::BorderContainer	520
Digikam::BorderFilter	522
Digikam::BorderSettings	526
Digikam::BqmInfolface	527

Digikam::BuildTrashCountersJob	530
Digikam::BWSepiaContainer	531
Digikam::BWSepiaFilter	533
Digikam::BWSepiaSettings	537
Digikam::CachedPixmapKey	538
Digikam::CachedPixmaps	538
Digikam::CameraAutoDetectThread	539
Digikam::CameraController	540
Digikam::CameraFolderDialog	542
Digikam::CameraFolderItem	543
Digikam::CameraFolderView	544
Digikam::CameraHistoryUpdater	545
Digikam::CameraInfoDialog	546
Digikam::CameraItem	547
Digikam::CameraItemList	548
Digikam::CameraList	549
Digikam::CameraMessageBox	550
Digikam::CameraNameHelper	550
Digikam::CameraNameOption	551
Digikam::CameraSelection	553
Digikam::CameraThumbsCtrl	554
Digikam::CameraType	555
Digikam::CamItemInfo	555
Digikam::CamItemSortSettings	558
Digikam::Canvas	562
Digikam::CaptionEdit	568
Digikam::CaptionsMap	570
Digikam::CaptionValues	571
Digikam::CaptureDlg	571
Digikam::CaptureWidget	572
Digikam::CaseModifier	573
Digikam::CategorizedItemModel	576
Digikam::CBContainer	577
Digikam::CBFilter	578
Digikam::CBSettings	581
Digikam::ChangeBookmarkCommand	582
Digikam::ChangeFaceRecognitionModelDlg	583
Digikam::ChangingDB	583
Digikam::CharcoalFilter	584
Digikam::CheckableAlbumFilterModel	588
Digikam::ChoiceSearchComboBox	593
Digikam::ChoiceSearchModel	596
Digikam::ChoiceSearchModel::Entry	599
Digikam::CIETongueWidget	600
Digikam::ClickDragReleaseItem	601
Digikam::ClockPhotoDialog	603
Digikam::CMat	604
Digikam::CollectionImageChangeset	605
Digikam::CollectionLocation	607
Digikam::CollectionManager	611
Digikam::CollectionManager::Private	618
Digikam::CollectionPage	620
Digikam::CollectionScanner	622
Digikam::CollectionScanner::Private	628
Digikam::CollectionScannerHintContainer	630
Digikam::CollectionScannerHintContainerImplementation	632
Digikam::CollectionScannerObserver	635
Digikam::ColorCorrectionDlg	636

Digikam::ColorFXContainer	636
Digikam::ColorFXFilter	637
Digikam::ColorFXSettings	640
Digikam::ColorGradientWidget	641
Digikam::ColorLabelFilter	642
Digikam::ColorLabelMenuAction	644
Digikam::ColorLabelSelector	645
Digikam::ColorLabelWidget	646
Digikam::ComboBoxDelegate	649
Digikam::CommentInfo	650
Digikam::CommonKeys	651
Digikam::CompressionDetector	653
Digikam::ContentAwareContainer	654
Digikam::ContentAwareFilter	655
Digikam::ContextMenuHelper	
A helper class to add actions and special menus to the context menu	658
Digikam::ContextMenuHelper::Private	670
Digikam::CoordinatesOverlayWidget	670
Digikam::CopyOrMoveJob	671
Digikam::CopyrightInfo	672
Digikam::CoreDB	673
Digikam::CoreDbAccess	721
Digikam::CoreDbAccessUnlock	724
Digikam::CoreDbBackend	725
Digikam::CoreDbBackendPrivate	729
Digikam::CoreDbBackendPrivate::ChangesetContainer< T >	731
Digikam::CoreDbCopyManager	732
Digikam::CoreDbDownloadHistory	732
Digikam::CoreDbNameFilter	733
Digikam::CoreDbOperationGroup	734
Digikam::CoreDbPrivilegesChecker	735
Digikam::CoreDbSchemaUpdater	735
Digikam::CoreDbTransaction	735
Digikam::CoreDbUrl	737
Digikam::CoreDbWatch	744
Digikam::CountrySelector	746
Digikam::CtrlButton	747
Digikam::CurvesBox	748
Digikam::CurvesContainer	749
Digikam::CurvesFilter	751
Digikam::CurvesSettings	755
Digikam::CurvesWidget	757
Digikam::CustomStepsDoubleSpinBox	760
Digikam::CustomStepsIntSpinBox	762
Digikam::DAboutData	764
Digikam::DAbstractSliderSpinBox	766
Digikam::DActiveLabel	768
Digikam::DAdjustableLabel	769
Digikam::DAlbum	770
Digikam::DAlbumDrag	772
Digikam::DAlbumInfo	773
Digikam::DArrowClickLabel	774
Digikam::DatabaseBlob	775
Digikam::DatabaseCopyThread	776
Digikam::DatabaseFields::DatabaseFieldsEnumIterator< fieldName >	776
Digikam::DatabaseFields::DatabaseFieldsEnumIteratorSetOnly< fieldName >	777
Digikam::DatabaseFields::FieldMetaInfo< fieldName >	778
Digikam::DatabaseFields::Hash< T >	778

Digikam::DatabaseFields::Set	779
Digikam::DatabaseLoadSaveFileInfoProvider	780
Digikam::DatabaseMigrationDialog	781
Digikam::DatabaseOption	782
Digikam::DatabaseOptionDialog	784
Digikam::DatabasePage	786
Digikam::DatabaseServer	787
Digikam::DatabaseServerError	788
Digikam::DatabaseServerStarter	789
Digikam::DatabaseSettingsWidget	790
Digikam::DatabaseSettingsWidget::Private	791
Digikam::DatabaseTask	792
Digikam::DatabaseVersionManager	794
Digikam::DatabaseWorkerInterface	796
Digikam::DatabaseWriter	799
Digikam::DateAlbumModel	802
Digikam::DateFolderView	809
Digikam::DateFolderViewSideBarWidget	812
Digikam::DateFormat	815
Digikam::DateOption	816
Digikam::DateOptionDialog	818
Digikam::DatePickerValidator	819
Digikam::DatePickerYearSelector	820
Digikam::DatesDBJobInfo	821
Digikam::DatesDBJobsThread	823
Digikam::DatesJob	825
Digikam::DateTreeView	827
Digikam::DbCleaner	831
Digikam::DbEngineAccess	834
Digikam::DbEngineAction	834
Digikam::DbEngineActionElement	834
Digikam::DbEngineActionType	834
Digikam::DbEngineConfig	835
Digikam::DbEngineConfigSettings	836
Digikam::DbEngineConfigSettingsLoader	836
Digikam::DbEngineConnectionChecker	837
Digikam::DbEngineErrorAnswer	838
Digikam::DbEngineErrorHandler	839
Digikam::DbEngineGuiErrorHandler	841
Digikam::DbEngineLocking	842
Digikam::DbEngineParameters	842
Digikam::DbEngineSqlQuery	847
Digikam::DbEngineThreadData	847
Digikam::DbHeaderListItem	848
Digikam::DBinaryIface	849
Digikam::DBinarySearch	851
Digikam::DBInfoIface	852
Digikam::DBJob	857
Digikam::DBJobInfo	858
Digikam::DBJobsManager	859
Digikam::DBJobsThread	862
Digikam::DbKeysCollection	
A class for managing / grouping database keys	864
Digikam::DbKeySelector	868
Digikam::DbKeySelectorItem	869
Digikam::DbKeySelectorView	870
Digikam::DbShrinkDialog	871
Digikam::DBStatDlg	872



Digikam::DBusyDlg	873
Digikam::DBusyThread	874
Digikam::DCameraDragObject	875
Digikam::DCameraItemListDrag	876
Digikam::DCategorizedSortFilterProxyModel	877
Digikam::DCategorizedSortFilterProxyModel::Private	882
Digikam::DCategorizedView	
Item view for listing items	882
Digikam::DCategorizedView::Private	886
Digikam::DCategorizedView::Private::ElementInfo	
Attributes	890
Digikam::DCategoryDrawer	891
Digikam::DClickLabel	896
Digikam::DColor	897
Digikam::DColorComposer	900
Digikam::DColorSelector	902
Digikam::DColorValueSelector	903
Digikam::DComboBox	908
Digikam::DConfigDlg	
A dialog base class which can handle multiple pages	909
Digikam::DConfigDlgMgr	919
Digikam::DConfigDlgModel	
A base class for a model used by <a href="#">DConfigDlgView</a>	925
Digikam::DConfigDlgModelPrivate	927
Digikam::DConfigDlgStackedWidget	928
Digikam::DConfigDlgTitle	929
Digikam::DConfigDlgTitle::Private	935
Digikam::DConfigDlgView	
A base class which can handle multiple pages	936
Digikam::DConfigDlgViewPrivate	942
Digikam::DConfigDlgWdg	
Page widget with many layouts (faces)	943
Digikam::DConfigDlgWdgItem	951
Digikam::DConfigDlgWdgModel	956
Digikam::DConfigDlgWdgModelPrivate	963
Digikam::DConfigDlgWdgPrivate	964
Digikam::DCursorTracker	965
Digikam::DDateEdit	967
Digikam::DDatePicker	970
Digikam::DDatePicker::Private	975
Digikam::DDatePickerPopup	
This menu helps the user to select a date quickly	976
Digikam::DDateTable	979
Digikam::DDateTable::Private	984
Digikam::DDateTable::Private::DatePaintingMode	986
Digikam::DDateTimeEdit	987
Digikam::DDoubleNumInput	989
Digikam::DDoubleSliderSpinBox	991
Digikam::DefaultRenameParser	994
Digikam::DefaultValueDialog	995
Digikam::DefaultValueModifier	996
Digikam::DefaultVersionNamingScheme	999
Digikam::DeleteDialog	1002
Digikam::DeleteItem	1003
Digikam::DeleteItemList	1004
Digikam::DeleteJob	1005
Digikam::DeleteWidget	1006
Digikam::DeltaTime	1007

Digikam::DetByClockPhotoButton	1008
Digikam::DetectionBenchmarker	1009
Digikam::DetectionWorker	1012
Digikam::DExpanderBox	1014
Digikam::DExpanderBoxExclusive	1017
Digikam::DFileDialog	1019
Digikam::DFileOperations	1020
Digikam::DFileSelector	1024
Digikam::DFontProperties	1026
Digikam::DFontSelect	1033
Digikam::DGradientSlider	1035
Digikam::DHBox	1036
Digikam::DHistoryView	1037
Digikam::DHueSaturationSelector	1038
Digikam::DigikamApp	1043
Digikam::DigikamApp::Private	1046
Digikam::DigikamItemDelegate	1049
Digikam::DigikamItemDelegatePrivate	1054
Digikam::DigikamItemView	1057
Digikam::DigikamItemView::Private	1066
Digikam::DImageHistory	1067
Digikam::DImageHistory::Entry	1070
Digikam::DImg	1071
Digikam::DImgBuiltinFilter	1089
Digikam::DImgChildItem	1093
Digikam::DImgFilterGenerator	1097
Digikam::DImgFilterManager	1099
Digikam::DImgLoader	1102
Digikam::DImgLoaderObserver	1105
Digikam::DImgLoaderSettings	1106
Digikam::DImgPreviewItem	1108
Digikam::DImgStaticPriv	1110
Digikam::DImgThreadedAnalyser	1111
Digikam::DImgThreadedFilter	1114
Digikam::DImgThreadedFilter::DefaultFilterAction< Filter >	1123
Digikam::DInfoInterface	1126
Digikam::DIntNumInput	1130
Digikam::DIntRangeBox	1131
Digikam::DIO	1133
Digikam::DirectoryNameOption	1135
Digikam::DisjointMetadata	1137
Digikam::DisjointMetadata::Private	1143
Digikam::DisjointMetadataDataFields	1145
Digikam::DistortionFXFilter	1147
Digikam::DItemDelegate	1151
Digikam::DItemDrag	1154
Digikam::DItemInfo	1155
Digikam::DItemsList	1157
Digikam::DItemsListView	1160
Digikam::DItemsListViewItem	1161
Digikam::DItemToolTip	1162
Digikam::DKCamera	1164
Digikam::DLabelExpander	1167
Digikam::DLineWidget	1168
Digikam::DLogoAction	1169
Digikam::DMessageBox	1169
Digikam::DMetadata	1173
Digikam::DMetadataSettings	1190

Digikam::DMetadataSettingsContainer	1191
Digikam::DMetaInfoInterface	1192
Digikam::DModelFactory	1196
Digikam::DMultiTabBar	1198
Digikam::DMultiTabBar::Private	1202
Digikam::DMultiTabBarButton	1203
Digikam::DMultiTabBarFrame	1205
Digikam::DMultiTabBarFrame::Private	1206
Digikam::DMultiTabBarTab	1207
Digikam::DMultiTabBarTab::Private	1210
Digikam::DNGConvertSettings	1210
Digikam::DNGSettings	1211
Digikam::DNGWriter	1212
Digikam::DNGWriter::Private	1213
Digikam::DNGWriterHost	1215
Digikam::DNNBaseDetectorModel	1216
Digikam::DNNFaceDetectorBase	1219
Digikam::DNNFaceDetectorSSD	1221
Digikam::DNNFaceDetectorYOLO	1223
Digikam::DNNFaceDetectorYuNet	1225
Digikam::DNNFaceExtractorBase	1227
Digikam::DNNModelBase	1229
Digikam::DNNModelConfig	1230
Digikam::DNNModelInfoContainer	1231
Digikam::DNNModelManager	1233
Digikam::DNNModelNet	1234
Digikam::DNNModelSFace	1236
Digikam::DNNModelYuNet	1237
Digikam::DNNOpenFaceExtractor	1239
Digikam::DNNResnetDetector	1241
Digikam::DNNFaceExtractor	1243
Digikam::DNNYoloDetector	1245
Digikam::DNotificationPopup	
A dialog-like popup that displays messages without interrupting the user	1247
Digikam::DNotificationWidget	1258
Digikam::DNotificationWidget::Private	1267
Digikam::DOnlineTranslator	
Provides translation data	1268
Digikam::DOnlineTranslator::Private	1280
Digikam::DOnlineTranslatorOption	
Contains translation options for a single word	1283
Digikam::DOnlineTts	
Provides TTS URL generation	1284
Digikam::DOnlineTts::Private	1290
Digikam::DownloadInfo	1290
Digikam::DownloadSettings	1291
Digikam::DPixelsAliasFilter	1292
Digikam::DPlainTextEdit	1293
Digikam::DPlainTextEdit::Private	1297
Digikam::DPlugin	1298
Digikam::DPluginAboutDlg	1302
Digikam::DPluginAction	1303
Digikam::DPluginAuthor	1305
Digikam::DPluginBqm	1306
Digikam::DPluginConfView	1310
Digikam::DPluginConfViewBqm	1313
Digikam::DPluginConfViewDlg	1315
Digikam::DPluginConfViewEditor	1317

Digikam::DPluginConfViewGeneric	1319
Digikam::DPluginDialog	1321
Digikam::DPluginDImg	1322
Digikam::DPluginEditor	1327
Digikam::DPluginGeneric	1331
Digikam::DPluginLoader	
The class that handles digiKam's external plugins	1334
Digikam::DPluginLoader::Private	1338
Digikam::DPluginRawImport	1340
Digikam::DPluginSetup	1343
Digikam::DPointSelect	1344
Digikam::DPopupFrame	1349
Digikam::DPreviewImage	1352
Digikam::DPreviewManager	1356
Digikam::DProgressDlg	1358
Digikam::DProgressWdg	1359
Digikam::DragDropModelImplementation	1361
Digikam::DragDropViewImplementation	1364
Digikam::DragHandle	1366
Digikam::DRawDecoder	1367
Digikam::DRawDecoder::Private	1374
Digikam::DRawDecoderSettings	1374
Digikam::DRawDecoderWidget	1382
Digikam::DRawDecoding	1385
Digikam::DRawInfo	1387
Digikam::DSaveSettingsWidget	1396
Digikam::DSelectionItem	1398
Digikam::DSelector	1399
Digikam::DServiceInfo	1402
Digikam::DServiceMenu	1402
Digikam::DSliderSpinBox	1404
Digikam::DSplashScreen	1407
Digikam::DSqueezedClickLabel	1408
Digikam::DTagListDrag	1409
Digikam::DTextBrowser	1410
Digikam::DTextEdit	1411
Digikam::DTextEdit::Private	1415
Digikam::DTextEditClearButton	1415
Digikam::DTextLabelName	1416
Digikam::DTextLabelValue	1418
Digikam::DTextList	1419
Digikam::DToolTipStyleSheet	1419
Digikam::DTrash	1420
Digikam::DTrashItemInfo	1421
Digikam::DTrashItemModel	1422
Digikam::DTrashItemsListingJob	1426
Digikam::DuplicatesFinder	1428
Digikam::DuplicatesProgressObserver	1431
Digikam::DVBox	1432
Digikam::DWItemDelegate	1434
Digikam::DWItemDelegatePool	1438
Digikam::DWItemDelegatePoolPrivate	1439
Digikam::DWItemDelegatePrivate	1440
Digikam::DWizardDlg	1441
Digikam::DWizardPage	1442
Digikam::DWorkingPixmap	1443
Digikam::DXmlGuiWindow	1444
Digikam::DXmlGuiWindow::Private	1449

Digikam::DynamicLayout	1451
Digikam::DynamicThread	1452
Digikam::DZoomBar	1456
Digikam::EditableSearchTreeView	1459
Digikam::EditorCore	1465
Digikam::EditorCore::Private	1468
Digikam::EditorCore::Private::FileToSave	1469
Digikam::EditorStackView	1470
Digikam::EditorTool	1472
Digikam::EditorToolface	1475
Digikam::EditorToolSettings	1476
Digikam::EditorToolThreaded	1478
Digikam::EditorWindow	1482
Digikam::EditorWindow::Private	1488
Digikam::EffectMgr	1489
Digikam::EffectMgr::Private	1490
Digikam::EffectPreview	1491
Digikam::Ellipsoid	1491
Digikam::EmbossFilter	1498
Digikam::EmptyDTrashItemsJob	1502
Digikam::EmptyImageListProvider	1504
Digikam::EqualizeFilter	1506
Digikam::ExifMetaEngineMergeHelper	1509
Digikam::ExifToolBinary	1511
Digikam::ExifToolConfPanel	1513
Digikam::ExifToolErrorView	1514
Digikam::ExifToolListView	1515
Digikam::ExifToolListViewGroup	1516
Digikam::ExifToolListViewItem	1517
Digikam::ExifToolLoadingView	1518
Digikam::ExifToolParser	1519
Digikam::ExifToolParser::Private	1524
Digikam::ExifToolProcess	1525
Digikam::ExifToolProcess::Private	1530
Digikam::ExifToolProcess::Private::Command	1531
Digikam::ExifToolProcess::Result	1532
Digikam::ExifToolThread	1532
Digikam::ExifToolWidget	1533
Digikam::ExifWidget	1535
Digikam::ExposureDetector	1538
Digikam::ExposureSettingsContainer	1539
Digikam::FaceClassifier	1540
Digikam::FaceClassifierBase	1542
Digikam::FaceDb	1543
Digikam::FaceDb::Private	1546
Digikam::FaceDbAccess	1547
Digikam::FaceDbAccessUnlock	1547
Digikam::FaceDbBackend	1548
Digikam::FaceDbOperationGroup	1551
Digikam::FaceDbSchemaUpdater	1552
Digikam::FaceDetector	1553
Digikam::FaceGroup	1555
Digikam::FaceGroup::Private	1559
Digikam::FaceItem	1560
Digikam::FaceItemRetriever	1563
Digikam::FacePipeline	1564
Digikam::FacePipeline::Private	1570
Digikam::FacePipelineBase	1572

Digikam::FacePipelineDetect	1576
Digikam::FacePipelineDetectRecognize	1580
Digikam::FacePipelineEdit	1584
Digikam::FacePipelineExtendedPackage	1589
Digikam::FacePipelineFaceTagsIface	1591
Digikam::FacePipelineFaceTagsIfaceList	1594
Digikam::FacePipelinePackage	1595
Digikam::FacePipelinePackageBase	1596
Digikam::FacePipelineRecognize	1598
Digikam::FacePipelineReset	1602
Digikam::FacePipelineRetrain	1606
Digikam::FacePreprocessor	1610
Digikam::FacePreviewLoader	1611
Digikam::FaceRejectionOverlay	1616
Digikam::FaceRejectionOverlayButton	1620
Digikam::FaceScanSettings	1622
Digikam::FaceScanWidget	1626
Digikam::FaceScanWidget::Private	1628
Digikam::FacesDetector	1629
Digikam::FacesEngine	1632
Digikam::FaceTags	1635
Digikam::FaceTagsEditor	1638
Digikam::FaceTagsIface	1644
Digikam::FaceUtils	1648
Digikam::FacialRecognitionWrapper	1652
Digikam::FacialRecognitionWrapper::Private	1655
Digikam::FFmpegBinary	1657
Digikam::FFmpegConfigHelper	1659
Digikam::FFmpegLauncher	1661
Digikam::FieldQueryBuilder	1663
Digikam::FileActionItemInfoList	1664
Digikam::FileActionMngr	1666
Digikam::FileActionMngr::Private	1668
Digikam::FileActionMngrDatabaseWorker	1670
Digikam::FileActionMngrFileWorker	1674
Digikam::FileActionProgress	1677
Digikam::FileActionProgressItemContainer	1680
Digikam::FileActionProgressItemCreator	1681
Digikam::FilePropertiesOption	1682
Digikam::FileReadLocker	1684
Digikam::FileReadWriteLockKey	1684
Digikam::FileSaveConflictBox	1685
Digikam::FileSaveOptionsBox	1686
Digikam::FileSaveOptionsDlg	1688
Digikam::FilesDownloader	1688
Digikam::FileWorkerInterface	1689
Digikam::FileWriteLocker	1691
Digikam::FilmContainer	1691
Digikam::FilmContainer::ListItem	1692
Digikam::FilmContainer::Private	1693
Digikam::FilmFilter	1694
Digikam::FilmFilter::Private	1697
Digikam::FilmGrainContainer	1697
Digikam::FilmGrainFilter	1698
Digikam::FilmGrainSettings	1702
Digikam::FilmProfile	1702
Digikam::Filter	1703
Digikam::FilterAction	1704

Digikam::FilterActionFilter	1708
Digikam::FiltersHistoryWidget	1713
Digikam::FilterSideBarWidget	1714
Digikam::FilterStatusBar	1718
Digikam::FindDuplicatesAlbum	
The FindDuplicatesAlbum class Widgets used to show all reference images	1719
Digikam::FindDuplicatesAlbumItem	1720
Digikam::FindDuplicatesView	1721
Digikam::FingerPrintsGenerator	1723
Digikam::FingerprintsTask	1727
Digikam::FirstRunDlg	1728
Digikam::FocusPoint	1729
Digikam::FocusPointGroup	1731
Digikam::FocusPointGroup::Private	1733
Digikam::FocusPointItem	1734
Digikam::FocusPointsExtractor	1737
Digikam::FocusPointsWriter	1738
Digikam::FrameOsd	1738
Digikam::FrameOsdSettings	1740
Digikam::FrameOsdWidget	1741
Digikam::FrameUtils	1741
Digikam::FreeRotationContainer	1741
Digikam::FreeRotationFilter	1743
Digikam::FreeRotationSettings	1746
Digikam::FreeSpaceToolTip	1748
Digikam::FreeSpaceWidget	1750
Digikam::FullObjectDetection	1751
Digikam::FullScreenSettings	1752
Digikam::FuzzySearchSideBarWidget	1753
Digikam::FuzzySearchView	1757
Digikam::FuzzySearchView::Private	1759
Digikam::GeoCoordinates	1760
Digikam::GeodeticCalculator	1761
Digikam::GeoDragDropHandler	1770
Digikam::GeofaceCluster	1770
Digikam::GeofaceGlobalObject	
Global object for geolocation interface to hold items common to all geolocation interface Widget instances	1771
Digikam::GeofaceInternalWidgetInfo	
Class to hold information about map widgets stored in the <a href="#">GeofaceGlobalObject</a>	1773
Digikam::GeofaceSharedData	1774
Digikam::GeolocationFilter	1776
Digikam::GeolocationSettings	1777
Digikam::GeolocationSettingsContainer	1779
Digikam::GeoModelHelper	
Helper class to access data in models	1780
Digikam::GeoPluginAboutDlg	1783
Digikam::GPCamera	1784
Digikam::GPSBookmarkModelHelper	1790
Digikam::GPSBookmarkOwner	1793
Digikam::GPSCorrelatorWidget	1794
Digikam::GPSDataContainer	1795
Digikam::GPSDBJobInfo	1796
Digikam::GPSDBJobsThread	1798
Digikam::GPSGeofaceModelHelper	1800
Digikam::GPSItemContainer	1803
Digikam::GPSItemDelegate	1806
Digikam::GPSItemInfo	1807

Digikam::GPSItemInfoSorter	1808
Digikam::GPSItemList	1809
Digikam::GPSItemListContextMenu	1810
Digikam::GPSItemListDragDropHandler	1812
Digikam::GPSItemModel	1813
Digikam::GPSItemSortProxyModel	1814
Digikam::GPSJob	1815
Digikam::GPSLinkItemSelectionModel	1817
Digikam::GPSMarkerTiler	
Marker model for storing data needed to display markers on the map. The data is retrieved from Digikam's database	1818
Digikam::GPSModelIndexProxyMapper	1825
Digikam::GPSSearchSideBarWidget	1828
Digikam::GPSSearchView	1832
Digikam::GPSUndoCommand	1835
Digikam::GPSUndoCommand::UndoInfo	1836
Digikam::Graph< VertexProperties, EdgeProperties >	1837
Digikam::Graph< VertexProperties, EdgeProperties >::DominatorTree	1845
Digikam::Graph< VertexProperties, EdgeProperties >::Edge	1845
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch	1846
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::BreadthFirstSearchVisitor	1847
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor	1848
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::DepthFirstSearchVisitor	1849
Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::lessThanMapEdgeToTarget< GraphType, VertexLessThan	1850
Digikam::Graph< VertexProperties, EdgeProperties >::Path	1850
Digikam::Graph< VertexProperties, EdgeProperties >::Vertex	1851
Digikam::GraphicsDImgItem	1852
Digikam::GraphicsDImgView	1854
Digikam::GreycstorationContainer	1856
Digikam::GreycstorationFilter	1858
Digikam::GreycstorationSettings	1863
Digikam::GroupedImagesFinder	1863
Digikam::GroupIndicatorOverlay	1865
Digikam::GroupIndicatorOverlayWidget	1869
Digikam::GroupingViewImplementation	1870
Digikam::GroupItemFilterSettings	1871
Digikam::GroupStateComputer	1872
Digikam::Haar::Calculator	1872
Digikam::Haar::ImageData	1873
Digikam::Haar::SignatureData	1873
Digikam::Haar::SignatureMap	1874
Digikam::Haar::WeightBin	1874
Digikam::Haar::Weights	1875
Digikam::HaarIface	1875
Digikam::HaarIface::Private	1879
Digikam::HaarProgressObserver	1880
Digikam::HidingStateChanger	1881
Digikam::Highlighter	1884
Digikam::HistogramBox	1885
Digikam::HistogramPainter	1886
Digikam::HistogramWidget	1891
Digikam::HistoryEdgeProperties	1893
Digikam::HistoryImageld	1894
Digikam::HistoryVertexProperties	1896
Digikam::HotPixelContainer	1896
Digikam::HotPixelFixer	1898
Digikam::HotPixelProps	1901



Digikam::HotPixelSettings	1902
Digikam::HotPixelsWeights	1903
Digikam::HoverButtonDelegateOverlay	1904
Digikam::HSLContainer	1907
Digikam::HSLFilter	1908
Digikam::HSLSettings	1911
Digikam::HSPreviewWidget	1912
Digikam::HTMLWidget	1913
Digikam::HTMLWidgetPage	1914
Digikam::IccManager	1915
Digikam::IccPostLoadingManager	1919
Digikam::ICCPreviewWidget	1921
Digikam::IccProfile	1922
Digikam::ICCPProfileInfoDlg	1925
Digikam::IccProfilesComboBox	1926
Digikam::IccProfilesMenuAction	1929
Digikam::IccProfilesSettings	1931
Digikam::ICCPProfileWidget	1933
Digikam::IccRenderingIntentComboBox	1936
Digikam::IccSettings	1937
Digikam::IccSettings::Private	1940
Digikam::ICCSettingsContainer	1940
Digikam::IccTransform	1942
Digikam::IccTransformFilter	1945
Digikam::Identity	1949
Digikam::IdentityProvider	1950
Digikam::ImageChangeset	1953
Digikam::ImageCommonContainer	1954
Digikam::ImageCurves	1954
Digikam::ImageDialog	1958
Digikam::ImageDialog::Private	1959
Digikam::ImageDialogIconProvider	1959
Digikam::ImageDialogPreview	1960
Digikam::ImageDialogToolTip	1961
Digikam::ImageGuideWidget	1963
Digikam::ImageHistogram	1965
Digikam::ImageHistoryEntry	1968
Digikam::ImageIface	1968
Digikam::ImageLevels	1972
Digikam::ImageListProvider	1974
Digikam::ImageMetadataContainer	1975
Digikam::ImagePreviewItem	1976
Digikam::ImageQualityCalculator	1977
Digikam::ImageQualityCalculator::ResultDetection	1978
Digikam::ImageQualityConfSelector	1978
Digikam::ImageQualityContainer	1979
Digikam::ImageQualityParser	1980
Digikam::ImageQualityParser::Private	1981
Digikam::ImageQualitySettings	1982
Digikam::ImageQualitySorter	1983
Digikam::ImageQualityTask	1987
Digikam::ImageQualityThread	1988
Digikam::ImageQualityThreadPool	1989
Digikam::ImageRegionItem	1990
Digikam::ImageRegionWidget	1992
Digikam::ImageRelation	1994
Digikam::ImageSortFilterModel	1995
Digikam::ImageTagChangeset	1997

Digikam::ImageTagProperty	1998
Digikam::ImageTagPropertyName	1999
Digikam::ImageWindow	2000
Digikam::ImageWindow::Private	2006
Digikam::ImageZoomSettings	2006
Digikam::ImportCategorizedView	2011
Digikam::ImportCategoryDrawer	2020
Digikam::ImportContextMenuHelper	2022
Digikam::ImportCoordinatesOverlay	2028
Digikam::ImportDelegate	2032
Digikam::ImportDelegate::ImportDelegatePrivate	2039
Digikam::ImportDownloadOverlay	2042
Digikam::ImportDragDropHandler	2046
Digikam::ImportFilterComboBox	2048
Digikam::ImportFilterDlg	2049
Digikam::ImportFilterModel	2051
Digikam::ImportIconView	2057
Digikam::ImportIconView::Private	2064
Digikam::ImportItemModel	2065
Digikam::ImportItemPropertiesSideBarImport	2073
Digikam::ImportItemPropertiesTab	2077
Digikam::ImportLockOverlay	2079
Digikam::ImportNormalDelegate	2083
Digikam::ImportNormalDelegatePrivate	2088
Digikam::ImportOverlayWidget	2090
Digikam::ImportPreviewView	2091
Digikam::ImportRatingOverlay	2094
Digikam::ImportRenameParser	2098
Digikam::ImportRotateOverlay	2099
Digikam::ImportRotateOverlayButton	2103
Digikam::ImportSettings	2106
Digikam::ImportSortFilterModel	2109
Digikam::ImportStackedView	2112
Digikam::ImportThumbnailBar	2114
Digikam::ImportThumbnailDelegate	2120
Digikam::ImportThumbnailDelegatePrivate	2126
Digikam::ImportThumbnailModel	2129
Digikam::ImportUI	2134
Digikam::ImportUI::Private	2137
Digikam::ImportView	2139
Digikam::InfoDlg	2141
Digikam::InfraredContainer	2142
Digikam::InfraredFilter	2143
Digikam::InitializationObserver	2147
Digikam::InsertBookmarksCommand	2148
Digikam::InternalTagName	2149
Digikam::InvertFilter	2150
Digikam::IOFileSettings	2153
Digikam::IOJob	2154
Digikam::IOJobData	2155
Digikam::IOJobsManager	2157
Digikam::IOJobsThread	2159
Digikam::lptcCoreContactInfo	2163
Digikam::lptcCoreLocationInfo	2163
Digikam::lptcMetaEngineMergeHelper	2164
Digikam::lptcWidget	2165
Digikam::ItemAlbumFilterModel	2168
Digikam::ItemAlbumModel	2174

Digikam::ItemAttributesWatch	2180
Digikam::ItemCategorizedView	2182
Digikam::ItemCategoryDrawer	2190
Digikam::ItemChangeHint	2192
Digikam::ItemComments	2193
Digikam::ItemCoordinatesOverlay	2199
Digikam::ItemCopyMoveHint	2202
Digikam::ItemCopyright	2203
Digikam::ItemDelegate	2208
Digikam::ItemDelegate::ItemDelegatePrivate	2215
Digikam::ItemDelegateOverlay	2217
Digikam::ItemDelegateOverlayContainer	2220
Digikam::ItemDescEditTab	2222
Digikam::ItemDescEditTab::Private	2224
Digikam::ItemDragDropHandler	2226
Digikam::ItemExtendedProperties	2228
Digikam::ItemFaceDelegate	2231
Digikam::ItemFaceDelegatePrivate	2236
Digikam::ItemFilterModel	2239
Digikam::ItemFilterModelFilterer	2249
Digikam::ItemFilterModelPrepareHook	2251
Digikam::ItemFilterModelPreparer	2252
Digikam::ItemFilterModelTodoPackage	2254
Digikam::ItemFilterModelWorker	2255
Digikam::ItemFilterSettings	2257
Digikam::ItemFiltersHistoryItemDelegate	2259
Digikam::ItemFiltersHistoryModel	2260
Digikam::ItemFiltersHistoryTreeItem	2261
Digikam::ItemFullScreenOverlay	2262
Digikam::ItemFullScreenOverlayButton	2266
Digikam::ItemGPS	2269
Digikam::ItemGPSModelHelper	2272
Digikam::ItemHistoryGraph	2274
Digikam::ItemHistoryGraphData	2279
Digikam::ItemHistoryGraphModel	2283
Digikam::ItemIconView	2286
Digikam::ItemIconView::Private	2291
Digikam::ItemInfo	
Provides access to the database for a single image. The properties can be read and written. Information will be cached	2292
Digikam::ItemInfoAlbumsJob	2308
Digikam::ItemInfoCache	2309
Digikam::ItemInfoData	2311
Digikam::ItemInfoJob	2313
Digikam::ItemInfoList	2314
Digikam::ItemInfoReadLocker	2315
Digikam::ItemInfoSet	2315
Digikam::ItemInfoStatic	2315
Digikam::ItemInfoTaskSplitter	2317
Digikam::ItemInfoWriteLocker	2319
Digikam::ItemListDragDropHandler	2320
Digikam::ItemLISTER	2320
Digikam::ItemLISTER::Private	2323
Digikam::ItemLISTERJobGrowingPartsSendingReceiver	2324
Digikam::ItemLISTERJobPartsSendingReceiver	2326
Digikam::ItemLISTERJobReceiver	2328
Digikam::ItemLISTERReceiver	2330
Digikam::ItemLISTERRecord	2331

Digikam::ItemListerValueListReceiver	2332
Digikam::ItemListModel	2334
Digikam::ItemMarkerTiler	2339
Digikam::ItemMetadataAdjustmentHint	2343
Digikam::ItemModel	2345
Digikam::ItemPosition	2354
Digikam::ItemPreviewCanvas	2358
Digikam::ItemPreviewView	2361
Digikam::ItemPropertiesColorsTab	2364
Digikam::ItemPropertiesGPSTab	2365
Digikam::ItemPropertiesHistoryTab	2366
Digikam::ItemPropertiesMetadataTab	2367
Digikam::ItemPropertiesSideBar	2368
Digikam::ItemPropertiesSideBarDB	2372
Digikam::ItemPropertiesTab	2377
Digikam::ItemPropertiesTab::Private	2381
Digikam::ItemPropertiesVersionsTab	2382
Digikam::ItemQueryBuilder	2383
Digikam::ItemQueryPostHook	2383
Digikam::ItemQueryPostHooks	2384
Digikam::ItemRatingOverlay	2385
Digikam::ItemRotateOverlay	2389
Digikam::ItemRotateOverlayButton	2393
Digikam::ItemScanInfo	2395
Digikam::ItemScanner	2395
Digikam::ItemScanner::Private	2402
Digikam::ItemScannerCommit	2402
Digikam::ItemSelectionOverlay	2403
Digikam::ItemSelectionOverlayButton	2407
Digikam::ItemSelectionPropertiesTab	2410
Digikam::ItemShortInfo	2412
Digikam::ItemSortCollator	2412
Digikam::ItemSortSettings	2413
Digikam::ItemTagPair	2417
Digikam::ItemThumbnailBar	2419
Digikam::ItemThumbnailDelegate	2426
Digikam::ItemThumbnailDelegatePrivate	2432
Digikam::ItemThumbnailModel	2435
Digikam::ItemVersionsModel	2441
Digikam::ItemViewCategorized	2442
Digikam::ItemViewDelegate	2449
Digikam::ItemViewDelegatePrivate	2455
Digikam::ItemViewHoverButton	2456
Digikam::ItemViewImportDelegate	2459
Digikam::ItemViewImportDelegatePrivate	2465
Digikam::ItemViewToolTip	2467
Digikam::ItemViewUtilities	2469
Digikam::ItemVisibilityController	2471
Digikam::ItemVisibilityControllerPropertyObject	2476
Digikam::JPEGUtils::digikam_source_mgr	2477
Digikam::JPEGUtils::JpegRotator	2477
Digikam::KDNNodeBase	2480
Digikam::KDNNodeBase::NodeCompareResult	2482
Digikam::KDNNodeOpenFace	2483
Digikam::KDNNodeSFace	2485
Digikam::KDTreeBase	2487
Digikam::KDTreeOpenFace	2489
Digikam::KDTreeSFace	2490

Digikam::KeywordSearchReader	2491
Digikam::KeywordSearchWriter	2493
Digikam::LabelsSideBarWidget	2495
Digikam::LabelsTreeView	2498
Digikam::LanguagesList	2501
Digikam::LcmsLock	2502
Digikam::LensDistortionFilter	2503
Digikam::LensDistortionPixelAccess	2506
Digikam::LensFunCameraSelector	2507
Digikam::LensFunContainer	2508
Digikam::LensFunFilter	2509
Digikam::LensFunIface	2512
Digikam::LensFunSettings	2514
Digikam::LessThanByProximityToSubject	2515
Digikam::LevelsContainer	2515
Digikam::LevelsFilter	2516
Digikam::LibsInfoDlg	2520
Digikam::LightTablePreview	2522
Digikam::LightTableThumbBar	2525
Digikam::LightTableView	2532
Digikam::LightTableWindow	2534
Digikam::LightTableWindow::Private	2537
Digikam::ListItem	2539
Digikam::ListViewComboBox	2541
Digikam::LoadingCache	2544
Digikam::LoadingCache::CacheLock	2548
Digikam::LoadingCacheFileWatch	2549
Digikam::LoadingCacheInterface	2550
Digikam::LoadingDescription	2551
Digikam::LoadingDescription::PostProcessingParameters	2555
Digikam::LoadingDescription::PreviewParameters	2556
Digikam::LoadingProcess	2557
Digikam::LoadingProcessListener	2558
Digikam::LoadingTask	2559
Digikam::LoadSaveFileInfoProvider	2561
Digikam::LoadSaveNotifier	2563
Digikam::LoadSaveTask	2565
Digikam::LoadSaveThread	2567
Digikam::LocalContrastContainer	2573
Digikam::LocalContrastFilter	2574
Digikam::LocalContrastSettings	2577
Digikam::LocalizeConfig	2578
Digikam::LocalizeContainer	2579
Digikam::LocalizeSelector	2580
Digikam::LocalizeSelectorList	2581
Digikam::LocalizeSettings	2582
Digikam::LookupAltitude	2584
Digikam::LookupAltitude::Request	2585
Digikam::LookupAltitudeGeonames	2586
Digikam::LookupFactory	2588
Digikam::MaintenanceData	2589
Digikam::MaintenanceDlg	2589
Digikam::MaintenanceDlg::Private	2590
Digikam::MaintenanceMgr	2591
Digikam::MaintenanceSettings	2592
Digikam::MaintenanceThread	2594
Digikam::MaintenanceTool	2597
Digikam::MakerNoteWidget	2601

Digikam::ManagedLoadSaveThread	2604
Digikam::MapBackend	2610
Digikam::MapDragData	2612
Digikam::MapDragDropHandler	2613
Digikam::MapViewModelHelper	2615
Digikam::MapWidget	
The central map view class of geolocation interface	2618
Digikam::MapWidget::Private	2625
Digikam::MapWidgetView	
Class containing digiKam's central map view	2626
Digikam::Mat	2630
Digikam::MdKeyListItem	2631
Digikam::MediaPlayerView	2632
Digikam::MetadataHub	2633
Digikam::MetadataHubMngr	2639
Digikam::MetadataKeys	2640
Digikam::MetadataListView	2642
Digikam::MetadataListItem	2643
Digikam::MetadataOption	2644
Digikam::MetadataOptionDialog	2646
Digikam::MetadataPage	2647
Digikam::MetadataPanel	2648
Digikam::MetadataRemover	2650
Digikam::MetadataRemoveTask	2654
Digikam::MetadataSelector	2655
Digikam::MetadataSelectorItem	2656
Digikam::MetadataSelectorView	2657
Digikam::MetadataStatusBar	2658
Digikam::MetadataSynchronizer	2659
Digikam::MetadataSyncTask	2663
Digikam::MetadataWidget	2665
Digikam::MetaEngine	2667
Digikam::MetaEngine::Private	2704
Digikam::MetaEngineData	2706
Digikam::MetaEngineData::Private	2706
Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >	2707
Digikam::MetaEnginePreviews	2708
Digikam::MetaEngineRotation	2709
Digikam::MetaEngineSettings	2712
Digikam::MetaEngineSettingsContainer	
The class <code>MetaEngineSettingsContainer</code> encapsulates all metadata related settings	2713
Digikam::MigrateFromDigiKam4Page	2716
Digikam::MimeFilter	2718
Digikam::MixerContainer	2719
Digikam::MixerFilter	2720
Digikam::MixerSettings	2724
Digikam::MLClassifierFoundation	2725
Digikam::MLClassifierFoundation::VotingGroups	2726
Digikam::MLClassifierFoundation::VotingGroups::VoteTally	2726
Digikam::MLPipelineFoundation	2727
Digikam::MLPipelineFoundation::_MLPipelinePerformanceProfile	2730
Digikam::MLPipelinePackageFoundation	2731
Digikam::MLPipelinePackageNotify	2732
Digikam::ModelCompleter	2733
Digikam::ModelIndexedComboBox	2735
Digikam::ModelMenu	2737
Digikam::Modifier	2739
Digikam::MonthWidget	2741

Digikam::MysqlAdminBinary	2742
Digikam::MysqlInitBinary	2745
Digikam::MysqlServerBinary	2748
Digikam::MysqlUpgradeBinary	2751
Digikam::NamespaceEditDlg	2753
Digikam::NamespaceEntry	
Provide a simple container for dmetadata namespaces variables, such as names, what types of data expects and extra xml tags	2754
Digikam::NamespaceListView	2756
Digikam::NetworkManager	2757
Digikam::NewItemFinder	2759
Digikam::NewlyAppearedFile	2762
Digikam::NoDuplicatesImportFilterModel	2763
Digikam::NoDuplicatesItemFilterModel	2766
Digikam::NoiseDetector	2768
Digikam::NonDeterministicRandomData	2769
Digikam::NormalizeFilter	2771
Digikam::NormalSearchTreeView	2775
Digikam::NRContainer	2781
Digikam::NREstimate	2782
Digikam::NRFilter	2786
Digikam::NRSettings	2790
Digikam::OilPaintFilter	2791
Digikam::OnlineVersionChecker	2794
Digikam::OnlineVersionDlg	2796
Digikam::OnlineVersionDwnl	2797
Digikam::OpenCVDNNFaceDetector	2797
Digikam::OpenCVDNNFaceRecognizer	2799
Digikam::OpenCVDNNFaceRecognizer::Private	2801
Digikam::OpenCVDNNFaceRecognizer::Private::ParallelRecognizer	2802
Digikam::OpenCVDNNFaceRecognizer::Private::ParallelTrainer	2803
Digikam::OpenfacePreprocessor	2803
Digikam::OpenFilePage	2804
Digikam::Option	2805
Digikam::OverlayWidget	2808
Digikam::PackageLoadingDescriptionList	2809
Digikam::PageItem	2810
Digikam::PALbum	2811
Digikam::PALbumPath	2813
Digikam::PanIconFrame	2814
Digikam::PanIconWidget	2817
Digikam::ParallelAdapter< A >	2819
Digikam::ParallelPipes	2822
Digikam::ParallelWorkers	2824
Digikam::Parser	2827
Digikam::ParseResults	2828
Digikam::ParseSettings	2829
Digikam::PeopleSideBarWidget	2830
Digikam::PersistentWidgetDelegateOverlay	2834
Digikam::PhotoInfoContainer	2838
Digikam::PickLabelFilter	2840
Digikam::PickLabelMenuAction	2842
Digikam::PickLabelSelector	2843
Digikam::PickLabelWidget	2844
Digikam::PlaceholderWidget	2846
Digikam::PointTransformAffine	2847
Digikam::PositionKeys	2847
Digikam::PreviewList	2849

Digikam::PreviewListItem	2850
Digikam::PreviewLoadingTask	2851
Digikam::PreviewLoadThread	2854
Digikam::PreviewPage	2860
Digikam::PreviewSettings	2861
Digikam::PreviewThreadWrapper	2862
Digikam::PreviewToolBar	2863
Digikam::PrivateProgressItemCreator	2864
Digikam::ProcessLauncher	2866
Digikam::ProgressEntry	2868
Digikam::ProgressItem	2868
Digikam::ProgressManager	
The <a href="#">ProgressManager</a> singleton keeps track of all ongoing transactions and notifies observers (progress dialogs) when their progress percent value changes, when they are completed (by their owner), and when they are canceled. Each <a href="#">ProgressItem</a> emits those signals individually and the singleton broadcasts them. Use the <a href="#">createProgressItem()</a> statics to acquire an item and then call <code>-&gt;setProgress( int percent )</code> on it every time you want to update the item and <code>-&gt;setComplete()</code> when the operation is done. This will delete the item. Connect to the item's <a href="#">progressItemCanceled()</a> signal to be notified when the user cancels the transaction using one of the observing progress dialogs or by calling <code>item-&gt;cancel()</code> in some other way. The owner is responsible for calling <code>setComplete()</code> on the item, even if it is canceled. Use the standard <code>CancelHandler()</code> slot if that is all you want to do on cancel	
Digikam::ProgressView	2882
Digikam::ProxyClickLineEdit	2885
Digikam::ProxyLineEdit	2887
Digikam::QListImageListProvider	2889
Digikam::QMapForAdaptors< Key, Value >	2891
Digikam::QueueListView	2892
Digikam::QueueListViewItem	2894
Digikam::QueueMgrWindow	2895
Digikam::QueueMgrWindow::Private	2898
Digikam::QueuePool	2899
Digikam::QueuePoolBar	2901
Digikam::QueueSettings	2901
Digikam::QueueSettingsView	2902
Digikam::QueueToolTip	2903
Digikam::RadioButtonHBox	2904
Digikam::RainDropFilter	2905
Digikam::RandomNumberGenerator	2908
Digikam::RangeDialog	2911
Digikam::RangeModifier	2912
Digikam::RatingBox	2915
Digikam::RatingComboBox	2917
Digikam::RatingComboBoxDelegate	2919
Digikam::RatingComboBoxModel	2920
Digikam::RatingComboBoxWidget	2921
Digikam::RatingFilter	2924
Digikam::RatingFilterWidget	2926
Digikam::RatingMenuAction	2928
Digikam::RatingStarDrawer	2929
Digikam::RatingWidget	2930
Digikam::RawCameraDlg	2932
Digikam::RawPage	2933
Digikam::RawProcessingFilter	2935
Digikam::RecognitionBenchmark	2941
Digikam::RecognitionBenchmark::Statistics	2943
Digikam::RecognitionPreprocessor	2944
Digikam::RecognitionTrainingProvider	2945



Digikam::RecognitionTrainingUpdateQueue	2946
Digikam::RecognitionWorker	2947
Digikam::RedEye::RegressionTree	2949
Digikam::RedEye::ShapePredictor	2950
Digikam::RedEye::SplitFeature	2951
Digikam::RedEyeCorrectionContainer	2951
Digikam::RedEyeCorrectionFilter	2952
Digikam::RedEyeCorrectionSettings	2956
Digikam::RefocusFilter	2957
Digikam::RefocusMatrix	2960
Digikam::RegionFrameItem	2961
Digikam::RemoveBookmarksCommand	2964
Digikam::RemoveDoublesModifier	2966
Digikam::RemoveFilterAction	2968
Digikam::RenameCustomizer	2969
Digikam::RenameFileJob	2970
Digikam::ReplaceDialog	2972
Digikam::ReplaceModifier	2973
Digikam::RestoreDTrashItemsJob	2975
Digikam::RGBBackend	
This class is a base class for Open Street Map and Geonames backends	2977
Digikam::RGInfo	
This class contains data needed in reverse geocoding process	2978
Digikam::RGTagModel	
The model that holds data for the tag tree displayed in ReverseGeocodingWidget	2979
Digikam::RGWidget	
Main widget for reverse geocoding	2988
Digikam::RubberItem	2992
Digikam::Rule	2995
Digikam::RuleDialog	3000
Digikam::RuleType	3000
Digikam::RuleTypeForConversion	3000
Digikam::SafeTemporaryFile	3001
Digikam::SAlbum	3002
Digikam::SaveProperties	3005
Digikam::SavingContext	3005
Digikam::SavingTask	3006
Digikam::ScanController	3009
Digikam::ScanController::FileMetadataWrite	3015
Digikam::ScanController::Private	3015
Digikam::ScanControllerCreator	3016
Digikam::ScanControllerLoadingCacheFileWatch	3017
Digikam::ScanStateFilter	3018
Digikam::SchemeManager	3020
Digikam::ScriptingSettings	3028
Digikam::SearchChangeset	3029
Digikam::SearchesDBJobInfo	3030
Digikam::SearchesDBJobsThread	3032
Digikam::SearchesJob	3035
Digikam::SearchField	3036
Digikam::SearchFieldAlbum	3039
Digikam::SearchFieldCheckBox	3043
Digikam::SearchFieldChoice	3047
Digikam::SearchFieldColorDepth	3051
Digikam::SearchFieldComboBox	3054
Digikam::SearchFieldGroup	3057
Digikam::SearchFieldGroupLabel	3058
Digikam::SearchFieldKeyword	3060

Digikam::SearchFieldLabels	3063
Digikam::SearchFieldMonthDay	3067
Digikam::SearchFieldPageOrientation	3071
Digikam::SearchFieldRangeDate	3074
Digikam::SearchFieldRangeDouble	3078
Digikam::SearchFieldRangeInt	3082
Digikam::SearchFieldRangeTime	3086
Digikam::SearchFieldRating	3090
Digikam::SearchFieldText	3094
Digikam::SearchFilterModel	3098
Digikam::SearchGroup	3102
Digikam::SearchGroupLabel	3105
Digikam::SearchInfo	3106
Digikam::SearchModel	3107
Digikam::SearchModificationHelper	3114
Digikam::SearchSideBarWidget	3119
Digikam::SearchTabHeader	3122
Digikam::SearchTextBar	3123
Digikam::SearchTextBarDb	3127
Digikam::SearchTextFilterSettings	3130
Digikam::SearchTextSettings	3131
Digikam::SearchTreeView	3132
Digikam::SearchView	3137
Digikam::SearchViewBottomBar	3140
Digikam::SearchViewThemedPartsCache	3141
Digikam::SearchWindow	3142
Digikam::SearchXmlCachingReader	3144
Digikam::SearchXmlReader	3147
Digikam::SearchXmlWriter	3151
Digikam::SequenceNumberDialog	3155
Digikam::SequenceNumberOption	3156
Digikam::Setup	3159
Digikam::SetupAlbumView	3162
Digikam::SetupCamera	3163
Digikam::SetupCategory	3164
Digikam::SetupCollectionDelegate	3165
Digikam::SetupCollectionModel	3169
Digikam::SetupCollectionModel::Item	3172
Digikam::SetupCollections	3173
Digikam::SetupCollectionTreeView	3174
Digikam::SetupDatabase	3175
Digikam::SetupEditor	3176
Digikam::SetupEditorIface	3177
Digikam::SetupGeolocation	3178
Digikam::SetupICC	3179
Digikam::SetupImageQualitySorter	3180
Digikam::SetupIOFiles	3181
Digikam::SetupLightTable	3181
Digikam::SetupMetadata	3182
Digikam::SetupMetadata::Private	3183
Digikam::SetupMime	3184
Digikam::SetupMisc	3185
Digikam::SetupMisc::Private	3186
Digikam::SetupPlugins	3187
Digikam::SetupRaw	3188
Digikam::SetupTemplate	3189
Digikam::SetupToolTip	3190
Digikam::SetupVersioning	3191

Digikam::SharedLoadingTask	3192
Digikam::SharedLoadSaveThread	3196
Digikam::SharedQueue< T >	3199
Digikam::SharpContainer	3199
Digikam::SharpenFilter	3201
Digikam::SharpSettings	3205
Digikam::ShearFilter	3206
Digikam::ShowHideVersionsOverlay	3210
Digikam::Sidebar	3214
Digikam::Sidebar::Private	3220
Digikam::SidebarSplitter	3221
Digikam::SidebarSplitter::Private	3223
Digikam::SidebarState	3223
Digikam::SidebarWidget	3224
Digikam::SidecarFinder	3227
Digikam::SimilarityDb	3227
Digikam::SimilarityDbAccess	3233
Digikam::SimilarityDbBackend	3236
Digikam::SimilarityDbSchemaUpdater	3239
Digikam::SimpleCollectionScannerObserver	3239
Digikam::SimpleTreeModel	3240
Digikam::SimpleTreeModel::Item	3241
Digikam::SinglePhotoPreviewLayout	3242
Digikam::SketchWidget	3245
Digikam::SlideVideo	3247
Digikam::SoftProofDialog	3248
Digikam::SolidHardwareDlg	3249
Digikam::SolidVolumeInfo	3250
Digikam::SparseModelIndexVector	3250
Digikam::SpellCheckConfig	3251
Digikam::SqueezedComboBox	3252
Digikam::StackedView	3256
Digikam::StartScanPage	3258
Digikam::State	3259
Digikam::StateSavingObject	3259
Digikam::StatusBarProgressWidget	3264
Digikam::StatusProgressBar	3265
Digikam::StayPoppedUpComboBox	3267
Digikam::StretchFilter	3270
Digikam::StyleSheetDebugger	3273
Digikam::SubjectData	3274
Digikam::SubjectEdit	3275
Digikam::SubjectWidget	3277
Digikam::SubQueryBuilder	3278
Digikam::SyncJob	3279
Digikam::SystemSettings	3279
Digikam::SystemSettingsWidget	3281
Digikam::TableView	3282
Digikam::TableViewColumn	3285
Digikam::TableViewColumnConfiguration	3288
Digikam::TableViewColumnConfigurationWidget	3288
Digikam::TableViewColumnDescription	3289
Digikam::TableViewColumnFactory	3289
Digikam::TableViewColumnProfile	3290
Digikam::TableViewColumns::ColumnAudioVideoProperties	3291
Digikam::TableViewColumns::ColumnDigikamProperties	3295
Digikam::TableViewColumns::ColumnFileConfigurationWidget	3298
Digikam::TableViewColumns::ColumnFileProperties	3300

Digikam::TableViewColumns::ColumnGeoConfigurationWidget	3303
Digikam::TableViewColumns::ColumnGeoProperties	3305
Digikam::TableViewColumns::ColumnItemProperties	3309
Digikam::TableViewColumns::ColumnPhotoConfigurationWidget	3312
Digikam::TableViewColumns::ColumnPhotoProperties	3314
Digikam::TableViewColumns::ColumnThumbnail	3318
Digikam::TableViewConfigurationDialog	3321
Digikam::TableViewItemDelegate	3322
Digikam::TableViewModel	3323
Digikam::TableViewModel::Item	3326
Digikam::TableViewSelectionModelSyncer	3326
Digikam::TableViewShared	3327
Digikam::TableViewTreeView	3328
Digikam::TagChangeset	3330
Digikam::TagCheckView	3331
Digikam::TagCompleter	3338
Digikam::TagData	3339
Digikam::TagDragDropHandler	3340
Digikam::TagEditDlg	3342
Digikam::TagFilterView	3344
Digikam::TagFolderView	3351
Digikam::TaggingAction	3358
Digikam::TaggingActionFactory	3359
Digikam::TaggingActionFactory::ConstraintInterface	3361
Digikam::TagInfo	3361
Digikam::TagList	3362
Digikam::TagMngrListModel	3363
Digikam::TagMngrListView	3365
Digikam::TagMngrTreeView	3367
Digikam::TagModel	3373
Digikam::TagModificationHelper	3379
Digikam::TagProperties	3385
Digikam::TagPropertiesFilterModel	3388
Digikam::TagProperty	3391
Digikam::TagPropertyName	3392
Digikam::TagPropWidget	3392
Digikam::TagRegion	3393
Digikam::TagsActionMngr	3396
Digikam::TagsCache	3398
Digikam::TagsDBJobInfo	3406
Digikam::TagsDBJobsThread	3407
Digikam::TagsEdit	3409
Digikam::TagShortInfo	3410
Digikam::TagsJob	3410
Digikam::TagsLineEditOverlay	3412
Digikam::TagsManager	3416
Digikam::TagsManagerFilterModel	3419
Digikam::TagsPopupMenu	3422
Digikam::TagTreeView	3424
Digikam::TagTreeViewSelectComboBox	3429
Digikam::TagViewSideBarWidget	3432
Digikam::TAlbum	3436
Digikam::Template	3439
Digikam::TemplateList	3442
Digikam::TemplateListItem	3443
Digikam::TemplateManager	3444
Digikam::TemplatePanel	3445
Digikam::TemplateSelector	3446

Digikam::TemplateViewer	3448
Digikam::TextFilter	3450
Digikam::TextureContainer	3451
Digikam::TextureFilter	3452
Digikam::TextureSettings	3455
Digikam::ThemeManager	3456
Digikam::ThemeManager::Private	3457
Digikam::ThreadManager	3458
Digikam::ThumbBarDock	3459
Digikam::ThumbnailAligningDelegate	3461
Digikam::ThumbnailCreator	3461
Digikam::ThumbnailCreator::Private	3465
Digikam::ThumbnailIdentifier	3466
Digikam::ThumbnailImage	3467
Digikam::ThumbnailImageCatcher	3467
Digikam::ThumbnailImageCatcher::Private	3469
Digikam::ThumbnailImageCatcher::Private::CatcherResult	3469
Digikam::ThumbnailInfo	3470
Digikam::ThumbnailInfoProvider	3472
Digikam::ThumbnailLoadingTask	3473
Digikam::ThumbnailLoadThread	3476
Digikam::ThumbnailLoadThread::Private	3485
Digikam::ThumbnailLoadThreadStaticPriv	3485
Digikam::ThumbnailResult	3486
Digikam::ThumbnailSize	3486
Digikam::ThumbsDb	3487
Digikam::ThumbsDbAccess	3488
Digikam::ThumbsDbBackend	3490
Digikam::ThumbsDbInfo	3493
Digikam::ThumbsDbInfoProvider	3493
Digikam::ThumbsDbSchemaUpdater	3494
Digikam::ThumbsGenerator	3495
Digikam::ThumbsTask	3498
Digikam::TileGrouper	3500
Digikam::TileIndex	3500
Digikam::TimeAdjustContainer	3501
Digikam::TimeAdjustSettings	3503
Digikam::TimelineSideBarWidget	3504
Digikam::TimeLineWidget	3507
Digikam::TimeZoneComboBox	3509
Digikam::Token	
Token is the smallest parsing unit in AdvancedRename utility	3510
Digikam::TonalityContainer	3512
Digikam::TonalityFilter	3513
Digikam::ToolListViewGroup	3516
Digikam::ToolListViewItem	3517
Digikam::ToolSettingsView	3518
Digikam::ToolsListView	3519
Digikam::ToolsView	3520
Digikam::TooltipCreator	3521
Digikam::TooltipDialog	3521
Digikam::TooltipsPage	3522
Digikam::TrackCorrelator	3523
Digikam::TrackCorrelator::Correlation	3524
Digikam::TrackCorrelator::CorrelationOptions	3524
Digikam::TrackCorrelatorThread	3525
Digikam::TrackListModel	3526
Digikam::TrackManager	3527

Digikam::TrackManager::Track	3528
Digikam::TrackManager::TrackPoint	3529
Digikam::TrackReader	3529
Digikam::TrackReader::TrackReadResult	3530
Digikam::TrainerWorker	3531
Digikam::TrainingDataProvider	3534
Digikam::TransactionItem	3536
Digikam::TransactionItemView	3538
Digikam::TransitionMngr	3539
Digikam::TransitionMngr::Private	3539
Digikam::TransitionPreview	3541
Digikam::TrashView	3542
Digikam::TreeBranch	3544
Digikam::TreeProxyModel	3544
Digikam::TreeViewComboBox	3545
Digikam::TreeViewLineEditComboBox	3548
Digikam::TrimmedModifier	3551
Digikam::TwoProgressItemsContainer	3553
Digikam::UMSCamera	3555
Digikam::UndoAction	3561
Digikam::UndoActionIrreversible	3562
Digikam::UndoActionReversible	3563
Digikam::UndoCache	3564
Digikam::UndoManager	3565
Digikam::UndoMetadataContainer	3565
Digikam::UndoState	3566
Digikam::UniqueModifier	3567
Digikam::UnsharpMaskFilter	3570
Digikam::VersionFileInfo	3573
Digikam::VersionFileOperation	3573
Digikam::VersioningPromptUserSaveDialog	3575
Digikam::VersionItemFilterSettings	3575
Digikam::VersionManager	3577
Digikam::VersionManagerSettings	3578
Digikam::VersionNamingScheme	3579
Digikam::VersionsDelegate	3582
Digikam::VersionsTreeView	3585
Digikam::VersionsWidget	3587
Digikam::VideoFrame	3588
Digikam::VideoInfoContainer	3588
Digikam::VideoMetadataContainer	3589
Digikam::VideoStripFilter	3589
Digikam::VideoThumbDecoder	3589
Digikam::VideoThumbDecoder::Private	3590
Digikam::VideoThumbnailer	3590
Digikam::VideoThumbWriter	3591
Digikam::VidPlayerDlg	3591
Digikam::VidSlideSettings	3591
Digikam::VidSlideTask	3598
Digikam::VidSlideThread	3600
Digikam::VisibilityController	3602
Digikam::VisibilityObject	3604
Digikam::WBContainer	3604
Digikam::WBFilter	3606
Digikam::WBSettings	3610
Digikam::WebBrowserDlg	3611
Digikam::WebWidget	3612
Digikam::WelcomePage	3613

Digikam::WelcomePageView	3614
Digikam::WelcomePageViewPage	3615
Digikam::WorkerObject	3616
Digikam::Workflow	3619
Digikam::WorkflowDlg	3620
Digikam::WorkflowItem	3621
Digikam::WorkflowList	3622
Digikam::WorkflowManager	3623
Digikam::WorkingWidget	3624
Digikam::WSAlbum	3625
Digikam::WSComboBoxIntermediate	3626
Digikam::WSLoginDialog	3627
Digikam::WSNewAlbumDialog	3628
Digikam::WSSelectUserDlg	3629
Digikam::WSSettings	3630
Digikam::WSSettingsWidget	3632
Digikam::WSToolDialog	3634
Digikam::WSToolUtils	3635
Digikam::XbelReader	3636
Digikam::XbelWriter	3636
Digikam::XmpWidget	3637
ShowFoto::NoDuplicatesShowfotoFilterModel	3640
ShowFoto::Showfoto	3643
ShowFoto::Showfoto::Private	3648
ShowFoto::ShowfotoCategorizedView	3649
ShowFoto::ShowfotoCoordinatesOverlay	3657
ShowFoto::ShowfotoCoordinatesOverlayWidget	3660
ShowFoto::ShowfotoDelegate	3662
ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate	3668
ShowFoto::ShowfotoDragDropHandler	3670
ShowFoto::ShowfotoFilterModel	3673
ShowFoto::ShowfotoFolderViewBar	3678
ShowFoto::ShowfotoFolderViewBookmarkDlg	3680
ShowFoto::ShowfotoFolderViewBookmarkItem	3681
ShowFoto::ShowfotoFolderViewBookmarkList	3682
ShowFoto::ShowfotoFolderViewBookmarks	3683
ShowFoto::ShowfotoFolderViewList	3684
ShowFoto::ShowfotoFolderViewModel	3685
ShowFoto::ShowfotoFolderViewSideBar	3686
ShowFoto::ShowfotoFolderViewToolTip	3689
ShowFoto::ShowfotoFolderViewUndo	3690
ShowFoto::ShowfotoInfoface	3691
ShowFoto::ShowfotoItemInfo	3693
ShowFoto::ShowfotoItemModel	3695
ShowFoto::ShowfotoItemSortSettings	3701
ShowFoto::ShowfotoItemViewDelegate	3705
ShowFoto::ShowfotoItemViewDelegatePrivate	3710
ShowFoto::ShowfotoKineticScroller	3712
ShowFoto::ShowfotoNormalDelegate	3713
ShowFoto::ShowfotoNormalDelegatePrivate	3718
ShowFoto::ShowfotoSettings	3720
ShowFoto::ShowfotoSetup	3722
ShowFoto::ShowfotoSetupMetadata	3725
ShowFoto::ShowfotoSetupMisc	3726
ShowFoto::ShowfotoSetupPlugins	3727
ShowFoto::ShowfotoSetupRaw	3728
ShowFoto::ShowfotoSetupToolTip	3729
ShowFoto::ShowfotoSortFilterModel	3730

---

ShowFoto::ShowfotoStackViewFavoriteItem	3733
ShowFoto::ShowfotoStackViewFavoriteItemDlg	3735
ShowFoto::ShowfotoStackViewFavoriteList	3736
ShowFoto::ShowfotoStackViewFavorites	3738
ShowFoto::ShowfotoStackViewItem	3739
ShowFoto::ShowfotoStackViewList	3740
ShowFoto::ShowfotoStackViewSideBar	3742
ShowFoto::ShowfotoStackViewToolTip	3744
ShowFoto::ShowfotoThumbnailBar	3746
ShowFoto::ShowfotoThumbnailDelegate	3752
ShowFoto::ShowfotoThumbnailDelegatePrivate	3757
ShowFoto::ShowfotoThumbnailModel	3759



# Chapter 5

## Namespace Documentation

### 5.1 Digikam Namespace Reference

#### Namespaces

- namespace [Matrix](#)

#### Classes

- class [AbstractAlbumModel](#)
- class [AbstractAlbumTreeView](#)
- class [AbstractAlbumTreeViewSelectComboBox](#)
- class [AbstractCheckableAlbumModel](#)
- class [AbstractCheckableAlbumTreeView](#)
- class [AbstractCountingAlbumModel](#)
- class [AbstractCountingAlbumTreeView](#)
- class [AbstractDetector](#)
- class [AbstractItemDragDropHandler](#)
- class [AbstractMarkerTiler](#)
- class [AbstractSearchGroupContainer](#)
- class [AbstractSpecificAlbumModel](#)
- class [AbstractWidgetDelegateOverlay](#)
- class [ActionCategorizedView](#)
- class [ActionData](#)
- class [ActionItemModel](#)
- class [ActionJob](#)
- class [ActionSortFilterProxyModel](#)
- class [ActionTask](#)
- class [ActionThread](#)
- class [ActionThreadBase](#)
- class [ActionVersionsOverlay](#)
- class [AddBookmarkDialog](#)
- class [AddBookmarkProxyModel](#)
- class [AddTagsComboBox](#)
- class [AddTagsLineEdit](#)
- class [AdvancedMetadataTab](#)
- class [AdvancedRenameDialog](#)

- class [AdvancedRenameInput](#)
- class [AdvancedRenameLineEdit](#)
- class [AdvancedRenameListItem](#)
- class [AdvancedRenameManager](#)
- class [AdvancedRenameProcessDialog](#)
- class [AdvancedRenameWidget](#)
- class [AdvancedSettings](#)
- class [AestheticDetector](#)
- class [AkonadiIface](#)
- class [Album](#)
  - *Abstract base class for all album types.*
- class [AlbumChangeset](#)
- class [AlbumCopyMoveHint](#)
- class [AlbumCustomizer](#)
- class [AlbumDragDropHandler](#)
- class [AlbumFilterModel](#)
- class [AlbumFolderViewSideBarWidget](#)
- class [AlbumHistory](#)
- class [AlbumInfo](#)
- class [AlbumIterator](#)
- class [AlbumLabelsSearchHandler](#)
- class [AlbumManager](#)
- class [AlbumManagerCreator](#)
- class [AlbumModel](#)
- class [AlbumModelDragDropHandler](#)
- class [AlbumModificationHelper](#)
- class [AlbumParser](#)
- class [AlbumPointer](#)
- class [AlbumPointerList](#)
- class [AlbumPropsEdit](#)
- class [AlbumRootChangeset](#)
- class [AlbumRootInfo](#)
- class [AlbumRootLocation](#)
- class [AlbumsDBJobInfo](#)
- class [AlbumsDBJobsThread](#)
- class [AlbumSelectComboBox](#)
- class [AlbumSelectDialog](#)
- class [AlbumSelectionTreeView](#)
- class [AlbumSelectors](#)
- class [AlbumSelectTabs](#)
- class [AlbumSelectTreeView](#)
- class [AlbumSelectWidget](#)
- class [AlbumShortInfo](#)
- class [AlbumSimplified](#)
- class [AlbumsJob](#)
- class [AlbumThumbnailLoader](#)
- class [AlbumTreeView](#)
- class [AlbumTreeViewDelegate](#)
- class [AlbumTreeViewSelectComboBox](#)
- class [AlbumWatch](#)
- class [AltLangStrEdit](#)
- class [AnimatedClearButton](#)
- class [AnimatedVisibility](#)
- class [AntiVignettingContainer](#)

- class [AntiVignettingFilter](#)
- class [AntiVignettingSettings](#)
- class [ApplicationSettings](#)
- class [AssignedBatchTools](#)
- class [AssignedListView](#)
- class [AssignedListViewItem](#)
- class [AssignNameOverlay](#)
- class [AssignNameWidget](#)
- class [AssignNameWidgetStates](#)
- class [AudPlayerWdg](#)
- class [AutoCrop](#)
- class [AutoExpoFilter](#)
- class [AutoLevelsFilter](#)
- class [AutoTagsAssign](#)
- class [AutotagsAssignment](#)
- class [AutotagsAssignmentTask](#)
- class [BackendGeonamesRG](#)

*This class calls Geonames' reverse geocoding service.*

- class [BackendGeonamesUSRG](#)

*This class calls Geonames' get address service available only for USA locations.*

- class [BackendGoogleMaps](#)
- class [BackendMarble](#)
- class [BackendMarbleLayer](#)
- class [BackendOsmRG](#)

*This class calls Open Street Map's reverse geocoding service.*

- class [BalooInfo](#)
- class [BalooWrap](#)

*The [BalooWrap](#) class is a singleton class which offer functionality for reading and writing image comment, tags and rating from Baloo to digiKam and from digiKam to Baloo.*

- class [BasicDImgFilterGenerator](#)
- class [BatchTool](#)
- class [BatchToolSet](#)
- class [BatchToolsFactory](#)
- class [BCGContainer](#)
- class [BCGFilter](#)
- class [BCGSettings](#)
- class [BdEngineBackend](#)
- class [BdEngineBackendPrivate](#)
- class [BlackFrameListView](#)
- class [BlackFrameListViewItem](#)
- class [BlackFrameParser](#)
- class [BlackFrameToolTip](#)
- class [BlurDetector](#)
- class [BlurFilter](#)
- class [BlurFXFilter](#)
- class [BookmarkNode](#)
- class [BookmarksDialog](#)
- class [BookmarksManager](#)
- class [BookmarksMenu](#)
- class [BookmarksModel](#)
- class [BorderContainer](#)
- class [BorderFilter](#)
- class [BorderSettings](#)
- class [BqmInfoiface](#)

- class [BuildTrashCountersJob](#)
- class [BWSepiaContainer](#)
- class [BWSepiaFilter](#)
- class [BWSepiaSettings](#)
- class [CachedPixmapKey](#)
- class [CachedPixmaps](#)
- class [CameraAutoDetectThread](#)
- class [CameraController](#)
- class [CameraFolderDialog](#)
- class [CameraFolderItem](#)
- class [CameraFolderView](#)
- class [CameraHistoryUpdater](#)
- class [CameraInfoDialog](#)
- class [CameraItem](#)
- class [CameraItemList](#)
- class [CameraList](#)
- class [CameraMessageBox](#)
- class [CameraNameHelper](#)
- class [CameraNameOption](#)
- class [CameraSelection](#)
- class [CameraThumbsCtrl](#)
- class [CameraType](#)
- class [CamlItemInfo](#)
- class [CamlItemSortSettings](#)
- class [Canvas](#)
- class [CaptionEdit](#)
- class [CaptionsMap](#)
- class [CaptionValues](#)
- class [CaptureDlg](#)
- class [CaptureWidget](#)
- class [CaseModifier](#)
- class [CategorizedItemModel](#)
- class [CBContainer](#)
- class [CBFilter](#)
- class [CBSettings](#)
- class [ChangeBookmarkCommand](#)
- class [ChangeFaceRecognitionModelDlg](#)
- class [ChangingDB](#)
- class [CharcoalFilter](#)
- class [CheckableAlbumFilterModel](#)
- class [ChoiceSearchComboBox](#)
- class [ChoiceSearchModel](#)
- class [CIETongueWidget](#)
- class [ClickDragReleaseItem](#)
- class [ClockPhotoDialog](#)
- struct [CMat](#)
- class [CollectionImageChangeset](#)
- class [CollectionLocation](#)
- class [CollectionManager](#)
- class [CollectionPage](#)
- class [CollectionScanner](#)
- class [CollectionScannerHintContainer](#)
- class [CollectionScannerHintContainerImplementation](#)
- class [CollectionScannerObserver](#)
- class [ColorCorrectionDlg](#)

- class [ColorFXContainer](#)
- class [ColorFXFilter](#)
- class [ColorFXSettings](#)
- class [ColorGradientWidget](#)
- class [ColorLabelFilter](#)
- class [ColorLabelMenuAction](#)
- class [ColorLabelSelector](#)
- class [ColorLabelWidget](#)
- class [ComboBoxDelegate](#)
- class [CommentInfo](#)
- class [CommonKeys](#)
- class [CompressionDetector](#)
- class [ContentAwareContainer](#)
- class [ContentAwareFilter](#)
- class [ContextMenuHelper](#)
  - *A helper class to add actions and special menus to the context menu.*
- class [CoordinatesOverlayWidget](#)
- class [CopyOrMoveJob](#)
- class [CopyrightInfo](#)
- class [CoreDB](#)
- class [CoreDbAccess](#)
- class [CoreDbAccessUnlock](#)
- class [CoreDbBackend](#)
- class [CoreDbBackendPrivate](#)
- class [CoreDbCopyManager](#)
- class [CoreDbDownloadHistory](#)
- class [CoreDbNameFilter](#)
- class [CoreDbOperationGroup](#)
- class [CoreDbPrivilegesChecker](#)
- class [CoreDbSchemaUpdater](#)
- class [CoreDbTransaction](#)
- class [CoreDbUrl](#)
- class [CoreDbWatch](#)
- class [CountrySelector](#)
- class [CtrlButton](#)
- class [CurvesBox](#)
- class [CurvesContainer](#)
- class [CurvesFilter](#)
- class [CurvesSettings](#)
- class [CurvesWidget](#)
- class [CustomStepsDoubleSpinBox](#)
- class [CustomStepsIntSpinBox](#)
- class [DAboutData](#)
- class [DAbstractSliderSpinBox](#)
- class [DActiveLabel](#)
- class [DAdjustableLabel](#)
- class [DAlbum](#)
- class [DAlbumDrag](#)
- class [DAlbumInfo](#)
- class [DArrowClickLabel](#)
- class [DatabaseBlob](#)
- class [DatabaseCopyThread](#)
- class [DatabaseLoadSaveFileInfoProvider](#)
- class [DatabaseMigrationDialog](#)

- class [DatabaseOption](#)
- class [DatabaseOptionDialog](#)
- class [DatabasePage](#)
- class [DatabaseServer](#)
- class [DatabaseServerError](#)
- class [DatabaseServerStarter](#)
- class [DatabaseSettingsWidget](#)
- class [DatabaseTask](#)
- class [DatabaseVersionManager](#)
- class [DatabaseWorkerInterface](#)
- class [DatabaseWriter](#)
- class [DateAlbumModel](#)
- class [DateFolderView](#)
- class [DateFolderViewSideBarWidget](#)
- class [DateFormat](#)
- class [DateOption](#)
- class [DateOptionDialog](#)
- class [DatePickerValidator](#)
- class [DatePickerYearSelector](#)
- class [DatesDBJobInfo](#)
- class [DatesDBJobsThread](#)
- class [DatesJob](#)
- class [DateTreeView](#)
- class [DbCleaner](#)
- class [DbEngineAccess](#)
- class [DbEngineAction](#)
- class [DbEngineActionElement](#)
- class [DbEngineActionType](#)
- class [DbEngineConfig](#)
- class [DbEngineConfigSettings](#)
- class [DbEngineConfigSettingsLoader](#)
- class [DbEngineConnectionChecker](#)
- class [DbEngineErrorAnswer](#)
- class [DbEngineErrorHandler](#)
- class [DbEngineGuiErrorHandler](#)
- class [DbEngineLocking](#)
- class [DbEngineParameters](#)
- class [DbEngineSqlQuery](#)
- class [DbEngineThreadData](#)
- class [DbHeaderListItem](#)
- class [DBinaryIface](#)
- class [DBinarySearch](#)
- class [DBInfoIface](#)
- class [DBJob](#)
- class [DBJobInfo](#)
- class [DBJobsManager](#)
- class [DBJobsThread](#)
- class [DbKeysCollection](#)
  - *A class for managing / grouping database keys.*
- class [DbKeySelector](#)
- class [DbKeySelectorItem](#)
- class [DbKeySelectorView](#)
- class [DbShrinkDialog](#)
- class [DBStatDlg](#)

- class [DBusyDlg](#)
- class [DBusyThread](#)
- class [DCameraDragObject](#)
- class [DCameraItemDrag](#)
- class [DCategorizedSortFilterProxyModel](#)
- class [DCategorizedView](#)
  - *Item view for listing items.*
- class [DCategoryDrawer](#)
- class [DClickLabel](#)
- class [DColor](#)
- class [DColorComposer](#)
- class [DColorSelector](#)
- class [DColorValueSelector](#)
- class [DComboBox](#)
- class [DConfigDlg](#)
  - *A dialog base class which can handle multiple pages.*
- class [DConfigDlgMgr](#)
- class [DConfigDlgModel](#)
  - *A base class for a model used by [DConfigDlgView](#).*
- class [DConfigDlgModelPrivate](#)
- class [DConfigDlgStackedWidget](#)
- class [DConfigDlgTitle](#)
- class [DConfigDlgView](#)
  - *A base class which can handle multiple pages.*
- class [DConfigDlgViewPrivate](#)
- class [DConfigDlgWdg](#)
  - *Page widget with many layouts (faces).*
- class [DConfigDlgWdgItem](#)
- class [DConfigDlgWdgModel](#)
- class [DConfigDlgWdgModelPrivate](#)
- class [DConfigDlgWdgPrivate](#)
- class [DCursorTracker](#)
- class [DDateEdit](#)
- class [DDatePicker](#)
- class [DDatePickerPopup](#)
  - *This menu helps the user to select a date quickly.*
- class [DDateTable](#)
- class [DDateTimeEdit](#)
- class [DDoubleNumInput](#)
- class [DDoubleSliderSpinBox](#)
- class [DefaultRenameParser](#)
- class [DefaultValueDialog](#)
- class [DefaultValueModifier](#)
- class [DefaultVersionNamingScheme](#)
- class [DeleteDialog](#)
- class [DeleteItem](#)
- class [DeleteItemList](#)
- class [DeleteJob](#)
- class [DeleteWidget](#)
- class [DeltaTime](#)
- class [DetByClockPhotoButton](#)
- class [DetectionBenchmark](#)
- class [DetectionWorker](#)
- class [DExpanderBox](#)

- class [DExpanderBoxExclusive](#)
- class [DFileDialog](#)
- class [DFileOperations](#)
- class [DFileSelector](#)
- class [DFontProperties](#)
- class [DFontSelect](#)
- class [DGradientSlider](#)
- class [DHBox](#)
- class [DHistoryView](#)
- class [DHueSaturationSelector](#)
- class [DigikamApp](#)
- class [DigikamItemDelegate](#)
- class [DigikamItemDelegatePrivate](#)
- class [DigikamItemView](#)
- class [DImageHistory](#)
- class [DImg](#)
- class [DImgBuiltinFilter](#)
- class [DImgChildItem](#)
- class [DImgFilterGenerator](#)
- class [DImgFilterManager](#)
- class [DImgLoader](#)
- class [DImgLoaderObserver](#)
- class [DImgLoaderSettings](#)
- class [DImgPreviewItem](#)
- class [DImgStaticPriv](#)
- class [DImgThreadedAnalyser](#)
- class [DImgThreadedFilter](#)
- class [DInfoInterface](#)
- class [DIntNumInput](#)
- class [DIntRangeBox](#)
- class [DIO](#)
- class [DirectoryNameOption](#)
- class [DisjointMetadata](#)
- class [DisjointMetadataDataFields](#)
- class [DistortionFXFilter](#)
- class [DItemDelegate](#)
- class [DItemDrag](#)
- class [DItemInfo](#)
- class [DItemsList](#)
- class [DItemsListView](#)
- class [DItemsListViewItem](#)
- class [DItemToolTip](#)
- class [DKCamera](#)
- class [DLabelExpander](#)
- class [DLineWidget](#)
- class [DLogoAction](#)
- class [DMessageBox](#)
- class [DMetadata](#)
- class [DMetadataSettings](#)
- class [DMetadataSettingsContainer](#)
- class [DMetaInfoface](#)
- class [DModelFactory](#)
- class [DMultiTabBar](#)
- class [DMultiTabBarButton](#)
- class [DMultiTabBarFrame](#)



- class [DMultiTabBarTab](#)
- class [DNGConvertSettings](#)
- class [DNGSettings](#)
- class [DNGWriter](#)
- class [DNGWriterHost](#)
- class [DNNBaseDetectorModel](#)
- class [DNNFaceDetectorBase](#)
- class [DNNFaceDetectorSSD](#)
- class [DNNFaceDetectorYOLO](#)
- class [DNNFaceDetectorYuNet](#)
- class [DNNFaceExtractorBase](#)
- class [DNNModelBase](#)
- class [DNNModelConfig](#)
- class [DNNModelInfoContainer](#)
- class [DNNModelManager](#)
- class [DNNModelNet](#)
- class [DNNModelSFace](#)
- class [DNNModelYuNet](#)
- class [DNNOpenFaceExtractor](#)
- class [DNNResnetDetector](#)
- class [DNNSFaceExtractor](#)
- class [DNNYoloDetector](#)
- class [DNotificationPopup](#)

*A dialog-like popup that displays messages without interrupting the user.*

- class [DNotificationWidget](#)
- class [DOnlineTranslator](#)

*Provides translation data.*

- struct [DOnlineTranslatorOption](#)

*Contains translation options for a single word.*

- class [DOnlineTts](#)

*Provides TTS URL generation.*

- class [DownloadInfo](#)
- class [DownloadSettings](#)
- class [DPixelsAliasFilter](#)
- class [DPlainTextEdit](#)
- class [DPlugin](#)
- class [DPluginAboutDlg](#)
- class [DPluginAction](#)
- class [DPluginAuthor](#)
- class [DPluginBqm](#)
- class [DPluginConfView](#)
- class [DPluginConfViewBqm](#)
- class [DPluginConfViewDImg](#)
- class [DPluginConfViewEditor](#)
- class [DPluginConfViewGeneric](#)
- class [DPluginDialog](#)
- class [DPluginDImg](#)
- class [DPluginEditor](#)
- class [DPluginGeneric](#)
- class [DPluginLoader](#)

*The class that handles digiKam's external plugins.*

- class [DPluginRawImport](#)
- class [DPluginSetup](#)
- class [DPointSelect](#)

- class [DPopupFrame](#)
- class [DPreviewImage](#)
- class [DPreviewManager](#)
- class [DProgressDlg](#)
- class [DProgressWdg](#)
- class [DragDropModelImplementation](#)
- class [DragDropViewImplementation](#)
- class [DragHandle](#)
- class [DRawDecoder](#)
- class [DRawDecoderSettings](#)
- class [DRawDecoderWidget](#)
- class [DRawDecoding](#)
- class [DRawInfo](#)
- class [DSaveSettingsWidget](#)
- class [DSelectedItem](#)
- class [DSelector](#)
- class [DServiceInfo](#)
- class [DServiceMenu](#)
- class [DSliderSpinBox](#)
- class [DSplashScreen](#)
- class [DSqueezedClickLabel](#)
- class [DTagListDrag](#)
- class [DTextBrowser](#)
- class [DTextEdit](#)
- class [DTextEditClearButton](#)
- class [DTextLabelName](#)
- class [DTextLabelValue](#)
- class [DTextList](#)
- class [DToolTipStyleSheet](#)
- class [DTrash](#)
- class [DTrashItemInfo](#)
- class [DTrashItemModel](#)
- class [DTrashItemsListingJob](#)
- class [DuplicatesFinder](#)
- class [DuplicatesProgressObserver](#)
- class [DVBox](#)
- class [DWItemDelegate](#)
- class [DWItemDelegatePool](#)
- class [DWItemDelegatePoolPrivate](#)
- class [DWItemDelegatePrivate](#)
- class [DWizardDlg](#)
- class [DWizardPage](#)
- class [DWorkingPixmap](#)
- class [DXmlGuiWindow](#)
- class [DynamicLayout](#)
- class [DynamicThread](#)
- class [DZoomBar](#)
- class [EditableSearchTreeView](#)
- class [EditorCore](#)
- class [EditorStackView](#)
- class [EditorTool](#)
- class [EditorTooliface](#)
- class [EditorToolSettings](#)
- class [EditorToolThreaded](#)
- class [EditorWindow](#)

- class [EffectMngr](#)
- class [EffectPreview](#)
- class [Ellipsoid](#)
- class [EmbossFilter](#)
- class [EmptyDTrashItemsJob](#)
- class [EmptyImageListProvider](#)
- class [EqualizeFilter](#)
- class [ExifMetaEngineMergeHelper](#)
- class [ExifToolBinary](#)
- class [ExifToolConfPanel](#)
- class [ExifToolErrorView](#)
- class [ExifToolListView](#)
- class [ExifToolListViewGroup](#)
- class [ExifToolListViewItem](#)
- class [ExifToolLoadingView](#)
- class [ExifToolParser](#)
- class [ExifToolProcess](#)
- class [ExifToolThread](#)
- class [ExifToolWidget](#)
- class [ExifWidget](#)
- class [ExposureDetector](#)
- class [ExposureSettingsContainer](#)
- class [FaceClassifier](#)
- class [FaceClassifierBase](#)
- class [FaceDb](#)
- class [FaceDbAccess](#)
- class [FaceDbAccessUnlock](#)
- class [FaceDbBackend](#)
- class [FaceDbOperationGroup](#)
- class [FaceDbSchemaUpdater](#)
- class [FaceDetector](#)
- class [FaceGroup](#)
- class [FaceItem](#)
- class [FaceItemRetriever](#)
- class [FacePipeline](#)
- class [FacePipelineBase](#)
- class [FacePipelineDetect](#)
- class [FacePipelineDetectRecognize](#)
- class [FacePipelineEdit](#)
- class [FacePipelineExtendedPackage](#)
- class [FacePipelineFaceTagsIface](#)
- class [FacePipelineFaceTagsIfaceList](#)
- class [FacePipelinePackage](#)
- class [FacePipelinePackageBase](#)
- class [FacePipelineRecognize](#)
- class [FacePipelineReset](#)
- class [FacePipelineRetrain](#)
- class [FacePreprocessor](#)
- class [FacePreviewLoader](#)
- class [FaceRejectionOverlay](#)
- class [FaceRejectionOverlayButton](#)
- class [FaceScanSettings](#)
- class [FaceScanWidget](#)
- class [FacesDetector](#)
- class [FacesEngine](#)

- class [FaceTags](#)
- class [FaceTagsEditor](#)
- class [FaceTagsIface](#)
- class [FaceUtils](#)
- class [FacialRecognitionWrapper](#)
- class [FFmpegBinary](#)
- class [FFmpegConfigHelper](#)
- class [FFmpegLauncher](#)
- class [FieldQueryBuilder](#)
- class [FileActionItemInfoList](#)
- class [FileActionMngr](#)
- class [FileActionMngrDatabaseWorker](#)
- class [FileActionMngrFileWorker](#)
- class [FileActionProgress](#)
- class [FileActionProgressItemContainer](#)
- class [FileActionProgressItemCreator](#)
- class [FilePropertiesOption](#)
- class [FileReadLocker](#)
- class [FileReadWriteLockKey](#)
- class [FileSaveConflictBox](#)
- class [FileSaveOptionsBox](#)
- class [FileSaveOptionsDlg](#)
- class [FilesDownloader](#)
- class [FileWorkerInterface](#)
- class [FileWriteLocker](#)
- class [FilmContainer](#)
- class [FilmFilter](#)
- class [FilmGrainContainer](#)
- class [FilmGrainFilter](#)
- class [FilmGrainSettings](#)
- class [FilmProfile](#)
- class [Filter](#)
- class [FilterAction](#)
- class [FilterActionFilter](#)
- class [FiltersHistoryWidget](#)
- class [FilterSideBarWidget](#)
- class [FilterStatusBar](#)
- class [FindDuplicatesAlbum](#)
  - *The [FindDuplicatesAlbum](#) class Widgets used to show all reference images.*
- class [FindDuplicatesAlbumItem](#)
- class [FindDuplicatesView](#)
- class [FingerPrintsGenerator](#)
- class [FingerprintsTask](#)
- class [FirstRunDlg](#)
- class [FocusPoint](#)
- class [FocusPointGroup](#)
- class [FocusPointItem](#)
- class [FocusPointsExtractor](#)
- class [FocusPointsWriter](#)
- class [FrameOsd](#)
- class [FrameOsdSettings](#)
- class [FrameOsdWidget](#)
- class [FrameUtils](#)
- class [FreeRotationContainer](#)

- class [FreeRotationFilter](#)
- class [FreeRotationSettings](#)
- class [FreeSpaceToolTip](#)
- class [FreeSpaceWidget](#)
- class [FullObjectDetection](#)
- class [FullScreenSettings](#)
- class [FuzzySearchSideBarWidget](#)
- class [FuzzySearchView](#)
- class [GeoCoordinates](#)
- class [GeodeticCalculator](#)
- class [GeoDragDropHandler](#)
- class [GeofaceCluster](#)
- class [GeofaceGlobalObject](#)
  - *Global object for geolocation interface to hold items common to all geolocation interface Widget instances.*
- class [GeofaceInternalWidgetInfo](#)
  - *Class to hold information about map widgets stored in the [GeofaceGlobalObject](#).*
- class [GeofaceSharedData](#)
- class [GeolocationFilter](#)
- class [GeolocationSettings](#)
- class [GeolocationSettingsContainer](#)
- class [GeoModelHelper](#)
  - *Helper class to access data in models.*
- class [GeoPluginAboutDlg](#)
- class [GPCamera](#)
- class [GPSBookmarkModelHelper](#)
- class [GPSBookmarkOwner](#)
- class [GPSCorrelatorWidget](#)
- class [GPSDataContainer](#)
- class [GPSDBJobInfo](#)
- class [GPSDBJobsThread](#)
- class [GPSGeofaceModelHelper](#)
- class [GPSItemContainer](#)
- class [GPSItemDelegate](#)
- class [GPSItemInfo](#)
- class [GPSItemInfoSorter](#)
- class [GPSItemList](#)
- class [GPSItemListContextMenu](#)
- class [GPSItemListDragDropHandler](#)
- class [GPSItemModel](#)
- class [GPSItemSortProxyModel](#)
- class [GPSJob](#)
- class [GPSLinkItemSelectionModel](#)
- class [GPSMarkerTiler](#)
  - *Marker model for storing data needed to display markers on the map. The data is retrieved from [Digikam's](#) database.*
- class [GPSModelIndexProxyMapper](#)
- class [GPSSearchSideBarWidget](#)
- class [GPSSearchView](#)
- class [GPSUndoCommand](#)
- class [Graph](#)
- class [GraphicsDImgItem](#)
- class [GraphicsDImgView](#)
- class [GreycstorationContainer](#)
- class [GreycstorationFilter](#)
- class [GreycstorationSettings](#)

- class [GroupedImagesFinder](#)
- class [GroupIndicatorOverlay](#)
- class [GroupIndicatorOverlayWidget](#)
- class [GroupingViewImplementation](#)
- class [GroupItemFilterSettings](#)
- class [GroupStateComputer](#)
- class [Haarface](#)
- class [HaarProgressObserver](#)
- class [HidingStateChanger](#)
- class [Highlighter](#)
- class [HistogramBox](#)
- class [HistogramPainter](#)
- class [HistogramWidget](#)
- class [HistoryEdgeProperties](#)
- class [HistoryImageId](#)
- class [HistoryVertexProperties](#)
- class [HotPixelContainer](#)
- class [HotPixelFixer](#)
- class [HotPixelProps](#)
- class [HotPixelSettings](#)
- class [HotPixelsWeights](#)
- class [HoverButtonDelegateOverlay](#)
- class [HSLContainer](#)
- class [HSLFilter](#)
- class [HSLSettings](#)
- class [HSPreviewWidget](#)
- class [HTMLWidget](#)
- class [HTMLWidgetPage](#)
- class [IccManager](#)
- class [IccPostLoadingManager](#)
- class [ICCPreviewWidget](#)
- class [IccProfile](#)
- class [ICCProfileInfoDlg](#)
- class [IccProfilesComboBox](#)
- class [IccProfilesMenuAction](#)
- class [IccProfilesSettings](#)
- class [ICCProfileWidget](#)
- class [IccRenderingIntentComboBox](#)
- class [IccSettings](#)
- class [ICCSettingsContainer](#)
- class [IccTransform](#)
- class [IccTransformFilter](#)
- class [Identity](#)
- class [IdentityProvider](#)
- class [ImageChangeset](#)
- class [ImageCommonContainer](#)
- class [ImageCurves](#)
- class [ImageDialog](#)
- class [ImageDialogIconProvider](#)
- class [ImageDialogPreview](#)
- class [ImageDialogToolTip](#)
- class [ImageGuideWidget](#)
- class [ImageHistogram](#)
- class [ImageHistoryEntry](#)
- class [ImageIface](#)

- class [ImageLevels](#)
- class [ImageListProvider](#)
- class [ImageMetadataContainer](#)
- class [ImagePreviewItem](#)
- class [ImageQualityCalculator](#)
- class [ImageQualityConfSelector](#)
- class [ImageQualityContainer](#)
- class [ImageQualityParser](#)
- class [ImageQualitySettings](#)
- class [ImageQualitySorter](#)
- class [ImageQualityTask](#)
- class [ImageQualityThread](#)
- class [ImageQualityThreadPool](#)
- class [ImageRegionItem](#)
- class [ImageRegionWidget](#)
- class [ImageRelation](#)
- class [ImageSortFilterModel](#)
- class [ImageTagChangeset](#)
- class [ImageTagProperty](#)
- class [ImageTagPropertyName](#)
- class [ImageWindow](#)
- class [ImageZoomSettings](#)
- class [ImportCategorizedView](#)
- class [ImportCategoryDrawer](#)
- class [ImportContextMenuHelper](#)
- class [ImportCoordinatesOverlay](#)
- class [ImportDelegate](#)
- class [ImportDownloadOverlay](#)
- class [ImportDragDropHandler](#)
- class [ImportFilterComboBox](#)
- class [ImportFilterDlg](#)
- class [ImportFilterModel](#)
- class [ImportIconView](#)
- class [ImportItemModel](#)
- class [ImportItemPropertiesSideBarImport](#)
- class [ImportItemPropertiesTab](#)
- class [ImportLockOverlay](#)
- class [ImportNormalDelegate](#)
- class [ImportNormalDelegatePrivate](#)
- class [ImportOverlayWidget](#)
- class [ImportPreviewView](#)
- class [ImportRatingOverlay](#)
- class [ImportRenameParser](#)
- class [ImportRotateOverlay](#)
- class [ImportRotateOverlayButton](#)
- class [ImportSettings](#)
- class [ImportSortFilterModel](#)
- class [ImportStackedView](#)
- class [ImportThumbnailBar](#)
- class [ImportThumbnailDelegate](#)
- class [ImportThumbnailDelegatePrivate](#)
- class [ImportThumbnailModel](#)
- class [ImportUI](#)
- class [ImportView](#)
- class [InfoDlg](#)

- class [InfraredContainer](#)
- class [InfraredFilter](#)
- class [InitializationObserver](#)
- class [InsertBookmarksCommand](#)
- class [InternalTagName](#)
- class [InvertFilter](#)
- class [IOFileSettings](#)
- class [IOJob](#)
- class [IOJobData](#)
- class [IOJobsManager](#)
- class [IOJobsThread](#)
- class [IptcCoreContactInfo](#)
- class [IptcCoreLocationInfo](#)
- class [IptcMetaEngineMergeHelper](#)
- class [IptcWidget](#)
- class [ItemAlbumFilterModel](#)
- class [ItemAlbumModel](#)
- class [ItemAttributesWatch](#)
- class [ItemCategorizedView](#)
- class [ItemCategoryDrawer](#)
- class [ItemChangeHint](#)
- class [ItemComments](#)
- class [ItemCoordinatesOverlay](#)
- class [ItemCopyMoveHint](#)
- class [ItemCopyright](#)
- class [ItemDelegate](#)
- class [ItemDelegateOverlay](#)
- class [ItemDelegateOverlayContainer](#)
- class [ItemDescEditTab](#)
- class [ItemDragDropHandler](#)
- class [ItemExtendedProperties](#)
- class [ItemFaceDelegate](#)
- class [ItemFaceDelegatePrivate](#)
- class [ItemFilterModel](#)
- class [ItemFilterModelFilterer](#)
- class [ItemFilterModelPrepareHook](#)
- class [ItemFilterModelPreparer](#)
- class [ItemFilterModelTodoPackage](#)
- class [ItemFilterModelWorker](#)
- class [ItemFilterSettings](#)
- class [ItemFiltersHistoryItemDelegate](#)
- class [ItemFiltersHistoryModel](#)
- class [ItemFiltersHistoryTreeItem](#)
- class [ItemFullScreenOverlay](#)
- class [ItemFullScreenOverlayButton](#)
- class [ItemGPS](#)
- class [ItemGPSModelHelper](#)
- class [ItemHistoryGraph](#)
- class [ItemHistoryGraphData](#)
- class [ItemHistoryGraphModel](#)
- class [ItemIconView](#)
- class [ItemInfo](#)

*The [ItemInfo](#) class contains provides access to the database for a single image. The properties can be read and written. Information will be cached.*



- class [ItemInfoAlbumsJob](#)
- class [ItemInfoCache](#)
- class [ItemInfoData](#)
- class [ItemInfoJob](#)
- class [ItemInfoList](#)
- class [ItemInfoReadLocker](#)
- class [ItemInfoSet](#)
- class [ItemInfoStatic](#)
- class [ItemInfoTaskSplitter](#)
- class [ItemInfoWriteLocker](#)
- class [ItemListDragDropHandler](#)
- class [ItemLister](#)
- class [ItemListerJobGrowingPartsSendingReceiver](#)
- class [ItemListerJobPartsSendingReceiver](#)
- class [ItemListerJobReceiver](#)
- class [ItemListerReceiver](#)
- class [ItemListerRecord](#)
- class [ItemListerValueListReceiver](#)
- class [ItemListModel](#)
- class [ItemMarkerTiler](#)
- class [ItemMetadataAdjustmentHint](#)
- class [ItemModel](#)
- class [ItemPosition](#)
- class [ItemPreviewCanvas](#)
- class [ItemPreviewView](#)
- class [ItemPropertiesColorsTab](#)
- class [ItemPropertiesGPSTab](#)
- class [ItemPropertiesHistoryTab](#)
- class [ItemPropertiesMetadataTab](#)
- class [ItemPropertiesSideBar](#)
- class [ItemPropertiesSideBarDB](#)
- class [ItemPropertiesTab](#)
- class [ItemPropertiesVersionsTab](#)
- class [ItemQueryBuilder](#)
- class [ItemQueryPostHook](#)
- class [ItemQueryPostHooks](#)
- class [ItemRatingOverlay](#)
- class [ItemRotateOverlay](#)
- class [ItemRotateOverlayButton](#)
- class [ItemScanInfo](#)
- class [ItemScanner](#)
- class [ItemScannerCommit](#)
- class [ItemSelectionOverlay](#)
- class [ItemSelectionOverlayButton](#)
- class [ItemSelectionPropertiesTab](#)
- class [ItemShortInfo](#)
- class [ItemSortCollator](#)
- class [ItemSortSettings](#)
- class [ItemTagPair](#)
- class [ItemThumbnailBar](#)
- class [ItemThumbnailDelegate](#)
- class [ItemThumbnailDelegatePrivate](#)
- class [ItemThumbnailModel](#)
- class [ItemVersionsModel](#)
- class [ItemViewCategorized](#)

- class [ItemViewDelegate](#)
- class [ItemViewDelegatePrivate](#)
- class [ItemViewHoverButton](#)
- class [ItemViewImportDelegate](#)
- class [ItemViewImportDelegatePrivate](#)
- class [ItemViewToolTip](#)
- class [ItemViewUtilities](#)
- class [ItemVisibilityController](#)
- class [ItemVisibilityControllerPropertyObject](#)
- class [KDNNodeBase](#)
- class [KDNNodeOpenFace](#)
- class [KDNNodeSFace](#)
- class [KDTreeBase](#)
- class [KDTreeOpenFace](#)
- class [KDTreeSFace](#)
- class [KeywordSearchReader](#)
- class [KeywordSearchWriter](#)
- class [LabelsSideBarWidget](#)
- class [LabelsTreeView](#)
- class [LanguagesList](#)
- class [LcmsLock](#)
- class [LensDistortionFilter](#)
- class [LensDistortionPixelAccess](#)
- class [LensFunCameraSelector](#)
- class [LensFunContainer](#)
- class [LensFunFilter](#)
- class [LensFunIface](#)
- class [LensFunSettings](#)
- class [LessThanByProximityToSubject](#)
- class [LevelsContainer](#)
- class [LevelsFilter](#)
- class [LibsInfoDlg](#)
- class [LightTablePreview](#)
- class [LightTableThumbBar](#)
- class [LightTableView](#)
- class [LightTableWindow](#)
- class [ListItem](#)
- class [ListViewComboBox](#)
- class [LoadingCache](#)
- class [LoadingCacheFileWatch](#)
- class [LoadingCacheInterface](#)
- class [LoadingDescription](#)
- class [LoadingProcess](#)
- class [LoadingProcessListener](#)
- class [LoadingTask](#)
- class [LoadSaveFileInfoProvider](#)
- class [LoadSaveNotifier](#)
- class [LoadSaveTask](#)
- class [LoadSaveThread](#)
- class [LocalContrastContainer](#)
- class [LocalContrastFilter](#)
- class [LocalContrastSettings](#)
- class [LocalizeConfig](#)
- class [LocalizeContainer](#)
- class [LocalizeSelector](#)

- class [LocalizeSelectorList](#)
- class [LocalizeSettings](#)
- class [LookupAltitude](#)
- class [LookupAltitudeGeonames](#)
- class [LookupFactory](#)
- class [MaintenanceData](#)
- class [MaintenanceDlg](#)
- class [MaintenanceMngr](#)
- class [MaintenanceSettings](#)
- class [MaintenanceThread](#)
- class [MaintenanceTool](#)
- class [MakerNoteWidget](#)
- class [ManagedLoadSaveThread](#)
- class [MapBackend](#)
- class [MapDragData](#)
- class [MapDragDropHandler](#)
- class [MapViewModelHelper](#)
- class [MapWidget](#)

*The central map view class of geolocation interface.*

- class [MapWidgetView](#)

*Class containing digiKam's central map view.*

- struct [Mat](#)
  - class [MdKeyListViewItem](#)
  - class [MediaPlayerView](#)
  - class [MetadataHub](#)
  - class [MetadataHubMngr](#)
  - class [MetadataKeys](#)
  - class [MetadataListView](#)
  - class [MetadataListViewItem](#)
  - class [MetadataOption](#)
  - class [MetadataOptionDialog](#)
  - class [MetadataPage](#)
  - class [MetadataPanel](#)
  - class [MetadataRemover](#)
  - class [MetadataRemoveTask](#)
  - class [MetadataSelector](#)
  - class [MetadataSelectorItem](#)
  - class [MetadataSelectorView](#)
  - class [MetadataStatusBar](#)
  - class [MetadataSynchronizer](#)
  - class [MetadataSyncTask](#)
  - class [MetadataWidget](#)
  - class [MetaEngine](#)
  - class [MetaEngineData](#)
  - class [MetaEngineMergeHelper](#)
  - class [MetaEnginePreviews](#)
  - class [MetaEngineRotation](#)
  - class [MetaEngineSettings](#)
  - class [MetaEngineSettingsContainer](#)
- The class [MetaEngineSettingsContainer](#) encapsulates all metadata related settings.*
- class [MigrateFromDigikam4Page](#)
  - class [MimeFilter](#)
  - class [MixerContainer](#)
  - class [MixerFilter](#)

- class [MixerSettings](#)
- class [MLClassifierFoundation](#)
- class [MLPipelineFoundation](#)
- class [MLPipelinePackageFoundation](#)
- class [MLPipelinePackageNotify](#)
- class [ModelCompleter](#)
- class [ModelIndexBasedComboBox](#)
- class [ModelMenu](#)
- class [Modifier](#)
- class [MonthWidget](#)
- class [MysqlAdminBinary](#)
- class [MysqlInitBinary](#)
- class [MysqlServerBinary](#)
- class [MysqlUpgradeBinary](#)
- class [NamespaceEditDlg](#)
- class [NamespaceEntry](#)

*The [NamespaceEntry](#) class provide a simple container for dmetadata namespaces variables, such as names, what types of data expects and extra xml tags.*

- class [NamespaceListView](#)
- class [NetworkManager](#)
- class [NewItemFinder](#)
- class [NewlyAppearedFile](#)
- class [NoDuplicatesImportFilterModel](#)
- class [NoDuplicatesItemFilterModel](#)
- class [NoiseDetector](#)
- class [NonDeterministicRandomData](#)
- class [NormalizeFilter](#)
- class [NormalSearchTreeView](#)
- class [NRContainer](#)
- class [NREstimate](#)
- class [NRFilter](#)
- class [NRSettings](#)
- class [OilPaintFilter](#)
- class [OnlineVersionChecker](#)
- class [OnlineVersionDlg](#)
- class [OnlineVersionDwnl](#)
- class [OpenCVDNNFaceDetector](#)
- class [OpenCVDNNFaceRecognizer](#)
- class [OpenfacePreprocessor](#)
- class [OpenFilePage](#)
- class [Option](#)
- class [OverlayWidget](#)
- class [PackageLoadingDescriptionList](#)
- class [PageItem](#)
- class [PAlbum](#)
- class [PAlbumPath](#)
- class [PanIconFrame](#)
- class [PanIconWidget](#)
- class [ParallelAdapter](#)
- class [ParallelPipes](#)
- class [ParallelWorkers](#)
- class [Parser](#)
- class [ParseResults](#)
- class [ParseSettings](#)

- class [PeopleSideBarWidget](#)
- class [PersistentWidgetDelegateOverlay](#)
- class [PhotoInfoContainer](#)
- class [PickLabelFilter](#)
- class [PickLabelMenuAction](#)
- class [PickLabelSelector](#)
- class [PickLabelWidget](#)
- class [PlaceholderWidget](#)
- class [PointTransformAffine](#)
- class [PositionKeys](#)
- class [PreviewList](#)
- class [PreviewListItem](#)
- class [PreviewLoadingTask](#)
- class [PreviewLoadThread](#)
- class [PreviewPage](#)
- class [PreviewSettings](#)
- class [PreviewThreadWrapper](#)
- class [PreviewToolBar](#)
- class [PrivateProgressItemCreator](#)
- class [ProcessLauncher](#)
- class [ProgressEntry](#)
- class [ProgressItem](#)
- class [ProgressManager](#)

The [ProgressManager](#) singleton keeps track of all ongoing transactions and notifies observers (progress dialogs) when their progress percent value changes, when they are completed (by their owner), and when they are canceled. Each [ProgressItem](#) emits those signals individually and the singleton broadcasts them. Use the [createProgressItem\(\)](#) statics to acquire an item and then call `->setProgress( int percent )` on it every time you want to update the item and `->setComplete()` when the operation is done. This will delete the item. Connect to the item's [progressItemCanceled\(\)](#) signal to be notified when the user cancels the transaction using one of the observing progress dialogs or by calling `item->cancel()` in some other way. The owner is responsible for calling `setComplete()` on the item, even if it is canceled. Use the `standardCancelHandler()` slot if that is all you want to do on cancel.

- class [ProgressView](#)
- class [ProxyClickLineEdit](#)
- class [ProxyLineEdit](#)
- class [QListImageListProvider](#)
- class [QMapForAdaptors](#)
- class [QueueListView](#)
- class [QueueListViewItem](#)
- class [QueueMgrWindow](#)
- class [QueuePool](#)
- class [QueuePoolBar](#)
- class [QueueSettings](#)
- class [QueueSettingsView](#)
- class [QueueToolTip](#)
- class [RadioButtonHBox](#)
- class [RainDropFilter](#)
- class [RandomNumberGenerator](#)
- class [RangeDialog](#)
- class [RangeModifier](#)
- class [RatingBox](#)
- class [RatingComboBox](#)
- class [RatingComboBoxDelegate](#)
- class [RatingComboBoxModel](#)
- class [RatingComboBoxWidget](#)
- class [RatingFilter](#)

- class [RatingFilterWidget](#)
- class [RatingMenuAction](#)
- class [RatingStarDrawer](#)
- class [RatingWidget](#)
- class [RawCameraDlg](#)
- class [RawPage](#)
- class [RawProcessingFilter](#)
- class [RecognitionBenchmark](#)
- class [RecognitionPreprocessor](#)
- class [RecognitionTrainingProvider](#)
- class [RecognitionTrainingUpdateQueue](#)
- class [RecognitionWorker](#)
- class [RedEyeCorrectionContainer](#)
- class [RedEyeCorrectionFilter](#)
- class [RedEyeCorrectionSettings](#)
- class [RefocusFilter](#)
- class [RefocusMatrix](#)
- class [RegionFrameItem](#)
- class [RemoveBookmarksCommand](#)
- class [RemoveDoublesModifier](#)
- class [RemoveFilterAction](#)
- class [RenameCustomizer](#)
- class [RenameFileJob](#)
- class [ReplaceDialog](#)
- class [ReplaceModifier](#)
- class [RestoreDTrashItemsJob](#)
- class [RGBackend](#)

*This class is a base class for Open Street Map and Geonames backends.*

- class [RGInfo](#)

*This class contains data needed in reverse geocoding process.*

- class [RGTagModel](#)

*The model that holds data for the tag tree displayed in ReverseGeocodingWidget.*

- class [RGWidget](#)

*The [RGWidget](#) class represents the main widget for reverse geocoding.*

- class [RubberItem](#)
- class [Rule](#)
- class [RuleDialog](#)
- class [RuleType](#)
- class [RuleTypeForConversion](#)
- class [SafeTemporaryFile](#)
- class [SAlbum](#)
- class [SaveProperties](#)
- class [SavingContext](#)
- class [SavingTask](#)
- class [ScanController](#)
- class [ScanControllerCreator](#)
- class [ScanControllerLoadingCacheFileWatch](#)
- class [ScanStateFilter](#)
- class [SchemeManager](#)
- class [ScriptingSettings](#)
- class [SearchChangeset](#)
- class [SearchesDBJobInfo](#)
- class [SearchesDBJobsThread](#)
- class [SearchesJob](#)

- class [SearchField](#)
- class [SearchFieldAlbum](#)
- class [SearchFieldCheckBox](#)
- class [SearchFieldChoice](#)
- class [SearchFieldColorDepth](#)
- class [SearchFieldComboBox](#)
- class [SearchFieldGroup](#)
- class [SearchFieldGroupLabel](#)
- class [SearchFieldKeyword](#)
- class [SearchFieldLabels](#)
- class [SearchFieldMonthDay](#)
- class [SearchFieldPageOrientation](#)
- class [SearchFieldRangeDate](#)
- class [SearchFieldRangeDouble](#)
- class [SearchFieldRangeInt](#)
- class [SearchFieldRangeTime](#)
- class [SearchFieldRating](#)
- class [SearchFieldText](#)
- class [SearchFilterModel](#)
- class [SearchGroup](#)
- class [SearchGroupLabel](#)
- class [SearchInfo](#)
- class [SearchModel](#)
- class [SearchModificationHelper](#)
- class [SearchSideBarWidget](#)
- class [SearchTabHeader](#)
- class [SearchTextBar](#)
- class [SearchTextBarDb](#)
- class [SearchTextFilterSettings](#)
- class [SearchTextSettings](#)
- class [SearchTreeView](#)
- class [SearchView](#)
- class [SearchViewBottomBar](#)
- class [SearchViewThemedPartsCache](#)
- class [SearchWindow](#)
- class [SearchXmlCachingReader](#)
- class [SearchXmlReader](#)
- class [SearchXmlWriter](#)
- class [SequenceNumberDialog](#)
- class [SequenceNumberOption](#)
- class [Setup](#)
- class [SetupAlbumView](#)
- class [SetupCamera](#)
- class [SetupCategory](#)
- class [SetupCollectionDelegate](#)
- class [SetupCollectionModel](#)
- class [SetupCollections](#)
- class [SetupCollectionTreeView](#)
- class [SetupDatabase](#)
- class [SetupEditor](#)
- class [SetupEditorIface](#)
- class [SetupGeolocation](#)
- class [SetupICC](#)
- class [SetupImageQualitySorter](#)
- class [SetupIOFiles](#)

- class [SetupLightTable](#)
- class [SetupMetadata](#)
- class [SetupMime](#)
- class [SetupMisc](#)
- class [SetupPlugins](#)
- class [SetupRaw](#)
- class [SetupTemplate](#)
- class [SetupToolTip](#)
- class [SetupVersioning](#)
- class [SharedLoadingTask](#)
- class [SharedLoadSaveThread](#)
- class [SharedQueue](#)
- class [SharpContainer](#)
- class [SharpenFilter](#)
- class [SharpSettings](#)
- class [ShearFilter](#)
- class [ShowHideVersionsOverlay](#)
- class [Sidebar](#)
- class [SidebarSplitter](#)
- class [SidebarState](#)
- class [SidebarWidget](#)
- class [SidecarFinder](#)
- class [SimilarityDb](#)
- class [SimilarityDbAccess](#)
- class [SimilarityDbBackend](#)
- class [SimilarityDbSchemaUpdater](#)
- class [SimpleCollectionScannerObserver](#)
- class [SimpleTreeModel](#)
- class [SinglePhotoPreviewLayout](#)
- class [SketchWidget](#)
- class [SlideVideo](#)
- class [SoftProofDialog](#)
- class [SolidHardwareDlg](#)
- class [SolidVolumeInfo](#)
- class [SparseModelIndexVector](#)
- class [SpellCheckConfig](#)
- class [SqueezedComboBox](#)
- class [StackedView](#)
- class [StartScanPage](#)
- struct [State](#)
- class [StateSavingObject](#)
- class [StatusbarProgressWidget](#)
- class [StatusProgressBar](#)
- class [StayPoppedUpComboBox](#)
- class [StretchFilter](#)
- class [StyleSheetDebugger](#)
- class [SubjectData](#)
- class [SubjectEdit](#)
- class [SubjectWidget](#)
- class [SubQueryBuilder](#)
- class [SyncJob](#)
- class [SystemSettings](#)
- class [SystemSettingsWidget](#)
- class [TableView](#)
- class [TableViewColumn](#)



- class [TableViewColumnConfiguration](#)
- class [TableViewColumnConfigurationWidget](#)
- class [TableViewColumnDescription](#)
- class [TableViewColumnFactory](#)
- class [TableViewColumnProfile](#)
- class [TableViewConfigurationDialog](#)
- class [TableViewItemDelegate](#)
- class [TableViewModel](#)
- class [TableViewSelectionModelSyncer](#)
- class [TableViewShared](#)
- class [TableViewTreeView](#)
- class [TagChangeset](#)
- class [TagCheckView](#)
- class [TagCompleter](#)
- struct [TagData](#)
- class [TagDragDropHandler](#)
- class [TagEditDlg](#)
- class [TagFilterView](#)
- class [TagFolderView](#)
- class [TaggingAction](#)
- class [TaggingActionFactory](#)
- class [TagInfo](#)
- class [TagList](#)
- class [TagMngrListModel](#)
- class [TagMngrListView](#)
- class [TagMngrTreeView](#)
- class [TagModel](#)
- class [TagModificationHelper](#)
- class [TagProperties](#)
- class [TagPropertiesFilterModel](#)
- class [TagProperty](#)
- class [TagPropertyName](#)
- class [TagPropWidget](#)
- class [TagRegion](#)
- class [TagsActionMngr](#)
- class [TagsCache](#)
- class [TagsDBJobInfo](#)
- class [TagsDBJobsThread](#)
- class [TagsEdit](#)
- class [TagShortInfo](#)
- class [TagsJob](#)
- class [TagsLineEditOverlay](#)
- class [TagsManager](#)
- class [TagsManagerFilterModel](#)
- class [TagsPopupMenu](#)
- class [TagTreeView](#)
- class [TagTreeViewSelectComboBox](#)
- class [TagViewSideBarWidget](#)
- class [TAlbum](#)
- class [Template](#)
- class [TemplateList](#)
- class [TemplateListItem](#)
- class [TemplateManager](#)
- class [TemplatePanel](#)
- class [TemplateSelector](#)

- class [TemplateViewer](#)
- class [TextFilter](#)
- class [TextureContainer](#)
- class [TextureFilter](#)
- class [TextureSettings](#)
- class [ThemeManager](#)
- class [ThreadManager](#)
- class [ThumbBarDock](#)
- class [ThumbnailAligningDelegate](#)
- class [ThumbnailCreator](#)
- class [ThumbnailIdentifier](#)
- class [ThumbnailImage](#)
- class [ThumbnailImageCatcher](#)
- class [ThumbnailInfo](#)
- class [ThumbnailInfoProvider](#)
- class [ThumbnailLoadingTask](#)
- class [ThumbnailLoadThread](#)
- class [ThumbnailLoadThreadStaticPriv](#)
- class [ThumbnailResult](#)
- class [ThumbnailSize](#)
- class [ThumbsDb](#)
- class [ThumbsDbAccess](#)
- class [ThumbsDbBackend](#)
- class [ThumbsDbInfo](#)
- class [ThumbsDbInfoProvider](#)
- class [ThumbsDbSchemaUpdater](#)
- class [ThumbsGenerator](#)
- class [ThumbsTask](#)
- class [TileGrouper](#)
- class [TileIndex](#)
- class [TimeAdjustContainer](#)
- class [TimeAdjustSettings](#)
- class [TimelineSideBarWidget](#)
- class [TimeLineWidget](#)
- class [TimeZoneComboBox](#)
- class [Token](#)
  - *Token is the smallest parsing unit in AdvancedRename utility*
- class [TonalityContainer](#)
- class [TonalityFilter](#)
- class [ToolListViewGroup](#)
- class [ToolListViewItem](#)
- class [ToolSettingsView](#)
- class [ToolsListView](#)
- class [ToolsView](#)
- class [TooltipCreator](#)
- class [TooltipDialog](#)
- class [TooltipsPage](#)
- class [TrackCorrelator](#)
- class [TrackCorrelatorThread](#)
- class [TrackListModel](#)
- class [TrackManager](#)
- class [TrackReader](#)
- class [TrainerWorker](#)
- class [TrainingDataProvider](#)

- class [TransactionItem](#)
- class [TransactionItemView](#)
- class [TransitionMngr](#)
- class [TransitionPreview](#)
- class [TrashView](#)
- class [TreeBranch](#)
- class [TreeProxyModel](#)
- class [TreeViewComboBox](#)
- class [TreeViewLineEditComboBox](#)
- class [TrimmedModifier](#)
- class [TwoProgressItemsContainer](#)
- class [UMSCamera](#)
- class [UndoAction](#)
- class [UndoActionIrreversible](#)
- class [UndoActionReversible](#)
- class [UndoCache](#)
- class [UndoManager](#)
- class [UndoMetadataContainer](#)
- class [UndoState](#)
- class [UniqueModifier](#)
- class [UnsharpMaskFilter](#)
- class [VersionFileInfo](#)
- class [VersionFileOperation](#)
- class [VersioningPromptUserSaveDialog](#)
- class [VersionItemFilterSettings](#)
- class [VersionManager](#)
- class [VersionManagerSettings](#)
- class [VersionNamingScheme](#)
- class [VersionsDelegate](#)
- class [VersionsTreeView](#)
- class [VersionsWidget](#)
- class [VideoFrame](#)
- class [VideoInfoContainer](#)
- class [VideoMetadataContainer](#)
- class [VideoStripFilter](#)
- class [VideoThumbDecoder](#)
- class [VideoThumbnailer](#)
- class [VideoThumbWriter](#)
- class [VidPlayerDlg](#)
- class [VidSlideSettings](#)
- class [VidSlideTask](#)
- class [VidSlideThread](#)
- class [VisibilityController](#)
- class [VisibilityObject](#)
- class [WBContainer](#)
- class [WBFilter](#)
- class [WBSettings](#)
- class [WebBrowserDlg](#)
- class [WebWidget](#)
- class [WelcomePage](#)
- class [WelcomePageView](#)
- class [WelcomePageViewPage](#)
- class [WorkerObject](#)
- class [Workflow](#)
- class [WorkflowDlg](#)

- class [WorkflowItem](#)
- class [WorkflowList](#)
- class [WorkflowManager](#)
- class [WorkingWidget](#)
- class [WSAlbum](#)
- class [WSComboBoxIntermediate](#)
- class [WSLoginDialog](#)
- class [WSNewAlbumDialog](#)
- class [WSSelectUserDlg](#)
- class [WSSettings](#)
- class [WSSettingsWidget](#)
- class [WSToolDialog](#)
- class [WSToolUtils](#)
- class [XbelReader](#)
- class [XbelWriter](#)
- class [XmpWidget](#)

## Typedefs

- typedef QHash< [ActionJob](#) \*, int > [ActionJobCollection](#)
- typedef QMap< qlonglong, int > [AlbumCache](#)
- typedef QList< [Album](#) \* > [AlbumList](#)
- typedef QMap< int, QPixmap > [AlbumThumbnailMap](#)
- typedef QList< [BatchToolSet](#) > [BatchSetList](#)
- typedef QMap< QString, QVariant > [BatchToolSettings](#)
- typedef QList< [BatchTool](#) \* > [BatchToolsList](#)
- typedef QPair< [CamItemInfo](#), QPixmap > [CachedItem](#)
- typedef QList< [CamItemInfo](#) > [CamItemInfoList](#)
- typedef QPair< QByteArray, CHUpdateItemMap > [CHUpdateItem](#)
- typedef QMultiMap< QDateTime, [CamItemInfo](#) > [CHUpdateItemMap](#)
- typedef QPair< QDateTime, QDateTime > [DateRange](#)
- typedef QList< [DateRange](#) > [DateRangeList](#)
- typedef QMap< QString, QString > [DbKeyIdsMap](#)
- typedef QMap< QString, [DbKeysCollection](#) \* > [DbOptionKeysMap](#)
- typedef QMap< QString, QVariant > [DImgLoaderPrms](#)
- typedef bool(\* [DItemsListsLessThanHandler](#)) (const QTreeWidgetItem \*current, const QTreeWidgetItem &other)
- typedef enum Digikam::\_DNNLoaderType [DNNLoaderType](#)
- typedef enum Digikam::\_DNNModelUsage [DNNModelUsage](#)
- typedef QList< [DNNModelUsage](#) > [DNNModelUsageList](#)
- typedef QList< [DownloadSettings](#) > [DownloadSettingsList](#)
- typedef QList< [DTrashItemInfo](#) > [DTrashItemInfoList](#)
- typedef FileReadWriteLockPriv [Entry](#)
- typedef QMap< QString, QStringList > [FFMpegProperties](#)
- typedef QList< [Filter](#) \* > [FilterList](#)
- typedef [Graph](#) < [HistoryVertexProperties](#), [HistoryEdgeProperties](#) > [HistoryGraph](#)
- typedef QMap< QString, ICCTagInfo > [ICCTagInfoMap](#)
- typedef QMap< QPair< qlonglong, QString >, QList< int > > [IdAlbumMap](#)
- typedef QSharedPointer< [DImgFilterGenerator](#) > [ImgFilterPtr](#)
- typedef QPair< int, int > [IntPair](#)
- typedef QList< [IntPair](#) > [IntPairList](#)
- typedef ItemInfoList::iterator [ItemInfoListIterator](#)
- typedef QExplicitlySharedDataPointer< ItemTagPairPriv > [ItemTagPairPrivSharedPointer](#)
- typedef QMap< QString, KLazyLocalizedString > [LanguageCodeMap](#)

- typedef QList< MetadataInfo::Field > **MetadataFields**
- typedef QHash< QString, QByteArray > **MyHash**
- typedef QPair< QUrl, QString > **NewNameInfo**
- typedef QList< NewNameInfo > **NewNamesList**
- typedef QPair< QPointF, [HudSide](#) > **OptimalPosition**
- typedef QPair< QString, QVariant > **PathValuePair**
- typedef QList< int > **QIntList**
- typedef QList< [ItemInfoSet](#) > **QueuePoolItemsList**
- typedef QList< [Rule](#) \* > **RulesList**
- typedef QMap< qlonglong, [Haar::SignatureData](#) > **SignatureCache**
- typedef struct [Digikam::TagData](#) **TagData**
- typedef QList< [TagProperty](#) >::const\_iterator **TagPropertiesConstIterator**
- typedef QExplicitlySharedDataPointer< TagProperties::TagPropertiesPriv > **TagPropertiesPrivShared**↔  
**Pointer**
- typedef QPair< TagPropertiesConstIterator, TagPropertiesConstIterator > **TagPropertiesRange**
- typedef QList< [Token](#) \* > **TokenList**
- typedef QPair< int, int > **YearMonth**

## Enumerations

- enum { **TaggingActionRole** = Qt::UserRole + 1 , **CompletionRole** = Qt::UserRole + 2 }
- enum **\_DNNLoaderType** { **DNNLoaderNet** , **DNNLoaderConfig** , **DNNLoaderYuNet** , **DNNLoaderSFace** }
- enum **\_DNNModelUsage** { **DNNUsageFaceDetection** , **DNNUsageFaceRecognition** , **DNNUsageRedeyeDetection** , **DNNUsage**↔  
**ObjectDetection** , **DNNUsageAesthetics** }
- enum **ChannelType** { **LuminosityChannel** = 0 , **RedChannel** , **GreenChannel** , **BlueChannel** , **AlphaChannel** , **ColorChannels** }
- enum **ClickDragState** { **HoverState** , **PressedState** , **PressDragState** , **ClickedMoveState** }
- enum **ColorLabel** { **NoColorLabel** = 0 , **RedLabel** , **OrangeLabel** , **YellowLabel** , **GreenLabel** , **BlueLabel** , **MagentaLabel** , **GrayLabel** , **BlackLabel** , **WhiteLabel** , **FirstColorLabel** = NoColorLabel , **LastColorLabel** = WhiteLabel , **NumberOfColorLabels** = LastColorLabel + 1 }
- enum **CropHandleFlag** { **CH\_None** , **CH\_Top** = 1 , **CH\_Left** = 2 , **CH\_Right** = 4 , **CH\_Bottom** = 8 , **CH\_TopLeft** = CH\_Top | CH\_Left , **CH\_BottomLeft** = CH\_Bottom | CH\_Left , **CH\_Top**↔  
**Right** = CH\_Top | CH\_Right , **CH\_BottomRight** = CH\_Bottom | CH\_Right , **CH\_Content** = 16 }
- enum **DColorChooserMode** { **ChooserClassic** = 0x0000 , **ChooserHue** = 0x0001 , **ChooserSaturation** = 0x0002 , **ChooserValue** = 0x0003 , **ChooserRed** = 0x0004 , **ChooserGreen** = 0x0005 , **ChooserBlue** = 0x0006 }
- enum [DetectorModel](#) { [YOLOV5NANO](#) = 0 , [YOLOV5XLARGE](#) , **RESNET50** }
- enum [DetectorNNModel](#) { [DNNDetectorSSD](#) = 0 , [DNNDetectorYOLOv3](#) , [DNNDetectorYuNet](#) }
- enum **DropAction** { **NoAction** , **CopyAction** , **MoveAction** , **GroupAction** , **SortAction** , **GroupAndMoveAction** , **AssignTagAction** }
- enum **FaceGroupState** { **NoFaces** , **FacesLoaded** }
- enum **FilterType** { **TEXT** = 0 , **MIME** , **GEOLOCATION** , **TAGS** , **LABELS** }
- enum **FocusPointGroupState** { **NoPoints** , **LoadingPoints** , **PointsLoaded** }

- enum `FullScreenOptions` {  
`FS_TOOLBARS` = 0x00000001 , `FS_THUMBBAR` = 0x00000002 , `FS_SIDEBARS` = 0x00000004 ,  
`FS_STATUSBAR` = 0x00000008 ,  
`FS_NONE` = 0x00000010 , `FS_ALBUMGUI` = `FS_TOOLBARS` | `FS_THUMBBAR` | `FS_SIDEBARS` | `FS_↔`  
`_STATUSBAR` , `FS_EDITOR` = `FS_TOOLBARS` | `FS_THUMBBAR` | `FS_SIDEBARS` | `FS_STATUSBAR` ,  
`FS_LIGHTTABLE` = `FS_TOOLBARS` | `FS_SIDEBARS` | `FS_STATUSBAR` ,  
`FS_IMPORTUI` = `FS_TOOLBARS` | `FS_THUMBBAR` | `FS_SIDEBARS` | `FS_STATUSBAR` }
  - enum class `FuzzyAlgorithm` { `Unknown` = 0 , `Haar` = 1 , `Tfidf` = 2 }
  - enum `GeoExtraAction` { `ExtraActionSticky` = 1 , `ExtraLoadTracks` = 2 }
  - enum `GeoGroupStateEnum` {  
`SelectedMask` = 0x03 << 0 , `SelectedNone` = 0x00 << 0 , `SelectedSome` = 0x03 << 0 , `SelectedAll` =  
0x02 << 0 ,  
`FilteredPositiveMask` = 0x03 << 2 , `FilteredPositiveNone` = 0x00 << 2 , `FilteredPositiveSome` = 0x03  
<< 2 , `FilteredPositiveAll` = 0x02 << 2 ,  
`RegionSelectedMask` = 0x03 << 4 , `RegionSelectedNone` = 0x00 << 4 , `RegionSelectedSome` = 0x03  
<< 4 , `RegionSelectedAll` = 0x02 << 4 }
- Representation of possible tile or cluster states.*
- enum `GeoMouseMove` {  
`MouseMovePan` = 1 , `MouseMoveRegionSelection` = 2 , `MouseMoveRegionSelectionFromIcon` = 4 ,  
`MouseMoveFilter` = 8 ,  
`MouseMoveSelectThumbnail` = 16 , `MouseMoveZoomIntoGroup` = 32 , `MouseMoveLast` = 32 }
  - enum `GroupAction` { `AddToGroup` , `RemoveFromGroup` , `Ungroup` }
  - enum `HistogramBoxType` {  
`RGB` = 0 , `RGBA` , `LRGB` , `LRGBA` ,  
`LRGBC` , `LRGBAC` }
  - enum `HistogramRenderingType` { `FullImageHistogram` = 0 , `ImageSelectionHistogram` }
  - enum `HistogramScale` { `LinScaleHistogram` = 0 , `LogScaleHistogram` }
  - enum `HudSide` {  
`HS_None` = 0 , `HS_Top` = 1 , `HS_Bottom` = 2 , `HS_Inside` = 4 ,  
`HS_TopInside` = `HS_Top` | `HS_Inside` , `HS_BottomInside` = `HS_Bottom` | `HS_Inside` }
  - enum `ImportRotateOverlayDirection` { `ImportRotateOverlayLeft` , `ImportRotateOverlayRight` }
  - enum `ItemRotateOverlayDirection` { `ItemRotateOverlayLeft` , `ItemRotateOverlayRight` }
  - enum `MapLayout` { `MapLayoutOne` = 0 , `MapLayoutHorizontal` = 1 , `MapLayoutVertical` = 2 }
  - enum `MatColorOrder` {  
`MCO_BGR` , `MCO_RGB` , `MCO_BGRA` = `MCO_BGR` , `MCO_RGBA` = `MCO_RGB` ,  
`MCO_ARGB` , `MCO_INVALID` }
  - enum `MeaningOfDirection` { `ParentToChild` , `ChildToParent` }
  - enum `OperationType` {  
`MetadataOps` = 0 , `ImportExportOps` , `BQMops` , `LightTableOps` ,  
`SlideshowOps` , `RenameOps` , `ToolsOps` , `UnspecifiedOps` }
  - enum `PickLabel` {  
`NoPickLabel` = 0 , `RejectedLabel` , `PendingLabel` , `AcceptedLabel` ,  
`FirstPickLabel` = `NoPickLabel` , `LastPickLabel` = `AcceptedLabel` , `NumberOfPickLabels` = `LastPickLabel` +  
1 }
  - enum `PreprocessorSelection` { `OPENFACE` = 0 }
  - enum `SKey` {  
`ALBUM` = 0 , `ALBUMNAME` , `ALBUMCAPTION` , `ALBUMCOLLECTION` ,  
`TAG` , `TAGNAME` , `IMAGENAME` , `IMAGECAPTION` ,  
`IMAGEDATE` , `KEYWORD` , `RATING` }
  - enum `SOperator` {  
`EQ` = 0 , `NE` , `LT` , `GT` ,  
`LIKE` , `NLIKE` , `LTE` , `GTE` }
  - enum `StdActionType` {  
`StdCopyAction` = 0 , `StdPasteAction` , `StdCutAction` , `StdQuitAction` ,  
`StdCloseAction` , `StdZoomInAction` , `StdZoomOutAction` , `StdOpenAction` ,  
`StdSaveAction` , `StdSaveAsAction` , `StdRevertAction` , `StdBackAction` ,  
`StdForwardAction` }

- enum **TrackColumns** { **ColumnVisible** = 0 , **ColumnNPoints** = 1 , **ColumnFilename** = 2 , **ColumnCount** = 3 }
- enum **Type** { **TypeChild** = 1 , **TypeSpacer** = 2 , **TypeNewChild** = 4 }
- enum class **YoloVersions** { **YOLOV5NANO** = 0 , **YOLOV5XLARGE** , **RESNET50** }

## Functions

- const QString **additionalInformation** ()
- DIGIKAM\_EXPORT QProcessEnvironment **adjustedEnvironmentForApplImage** ()
- QDateTime **asDateTimeLocal** (const QDateTime &dt)
- QDateTime **asDateTimeUTC** (const QDateTime &dt)
- bool **checkSidecarSettings** ()
- void **checkTree** (**TreeBranch** \*const checkBranch, int level)
- QStringList **cleanUserFilterString** (QString filterString, const bool caseSensitive, const bool useSemicolon)
- template<class ContainerA , class ContainerB >  
bool **containsAnyOf** (const ContainerA &listA, const ContainerB &listB)
- template<class ContainerA , typename Value , class ContainerB >  
bool **containsNoneOfExcept** (const ContainerA &list, const ContainerB &noneOfList, const Value &exception)
- void **coordinatesToClipboard** (const **GeoCoordinates** &coordinates, const QUrl &url, const QString &title)
- MetadataInfo::Field **DatabasImageMetadataFieldsToMetadataInfoField** (const DatabaseFields::Image↔ Metadata imageMetadataField)
- MetadataInfo::Field **DatabaseVideoMetadataFieldsToMetadataInfoField** (const DatabaseFields::Video↔ Metadata videoMetadataField)
- QShortcut \* **defineShortcut** (QWidget \*const w, const QKeySequence &key, const QObject \*receiver, const char \*slot)
- const QDateTime **digikamBuildDate** ()
- int **digikamMakeIntegerVersion** (int major, int minor, int patch)
- const QString **digikamVersion** ()
- void **DNotificationWrapper** (const QString &eventId, const QString &message, QWidget \*const parent, const QString &>windowTitle, const QPixmap &pixmap=QPixmap())  
*Show a notification using KNotify, or KPassivePopup if KNotify is unavailable.*
- template<typename T >  
**PointTransformAffine findAffineTransform** (const std::vector< std::vector< T > > &fromPoints, const std::vector< std::vector< T > > &toPoints)
- **PointTransformAffine findSimilarityTransform** (const std::vector< std::vector< float > > &fromPoints, const std::vector< std::vector< float > > &toPoints)
- void **Geoface\_assert** (const char \*const condition, const char \*const filename, const int lineNumber)
- GeoCoordinates::PairList **GeofaceHelperNormalizeBounds** (const GeoCoordinates::Pair &boundsPair)  
*Split bounds crossing the dateline into parts which do not cross the dateline.*
- bool **GeofaceHelperParseBoundsString** (const QString &boundsString, QPair< **GeoCoordinates**, **GeoCoordinates** > \*const boundsCoordinates)  
*Parses a '((lat1, lon1), (lat2, lon2))' bounds string as returned by the JavaScript parts.*
- bool **GeofaceHelperParseLatLonString** (const QString &latLonString, **GeoCoordinates** \*const coordinates)  
*Parses a 'lat,lon' string as returned by the JavaScript parts.*
- bool **GeofaceHelperParseXYStringToPoint** (const QString &xyString, QPoint \*const point)  
*Parses a '(X.xxx,Y.yyy)' string as returned by the JavaScript parts.*
- qreal **getColorComponentValue** (const QColor &color, DColorChooserMode chooserMode)
- std::vector< cv::Rect > **getEyes** (const **FullObjectDetection** &shape)
- QString **getUserAgentName** ()
- int **getWarningLevelFromGPSDataContainer** (const **GPSDataContainer** &data)
- bool **iccProfileLessThan** (**IccProfile** a, **IccProfile** b)
- cv::Mat **image2Mat** (const QImage &img, int requiredMatType, MatColorOrder requiredOrder)
- cv::Mat **image2Mat\_shared** (const QImage &img, MatColorOrder \*const order)

- NoiseDetector::Mat3D **initFiltersHaar** ()
- DIGIKAM\_EXPORT void **installQtTranslationFiles** (QApplication &app)
- **PointTransformAffine inv** (const [PointTransformAffine](#) &trans)
- bool **isCursorClicked** (const QPoint &pos, double cursorPos, int width, int height, int gradientWidth)
- bool **isReadableImageFile** (const QString &filePath)
- DIGIKAM\_EXPORT bool **isRunningInApplImageBundle** ()
- bool **isRunningOnMacOS** ()
- DIGIKAM\_EXPORT bool **isRunningOnNativeKDE** ()
- int **layoutMargin** ()
- int **layoutSpacing** ()
- template<class T >  
T **length\_squared** (const std::vector< T > &diff)
- bool **lessThanByTitle** (const [Album](#) \*first, const [Album](#) \*second)  
for *qSort*
- DIGIKAM\_EXPORT void **loadEcmQtTranslationFiles** (QApplication &app)
- DIGIKAM\_EXPORT void **loadStdQtTranslationFiles** (QApplication &app)
- DIGIKAM\_EXPORT QString **macOSBundlePrefix** ()
- QStringList **makeTagString** (const [RGInfo](#) &info, const QString &inputFormat, const QString &backend←  
Name)
- QImage **mat2Image** (const cv::Mat &mat, MatColorOrder order, QImage::Format formatHint)
- QImage **mat2Image\_shared** (const cv::Mat &mat, QImage::Format formatHint)
- void **openOnlineDocumentation** (const QString &section, const QString &chapter, const QString &reference)
- [FocusPoint::TypePoint](#) **operator&** ([FocusPoint::TypePoint](#) type1, [FocusPoint::TypePoint](#) type2)
- [FocusPoint::TypePoint](#) & **operator&=** ([FocusPoint::TypePoint](#) &type1, [FocusPoint::TypePoint](#) type2)
- [PointTransformAffine](#) **operator\*** (const [PointTransformAffine](#) &lhs, const [PointTransformAffine](#) &rhs)
- template<class T >  
std::vector< std::vector< T > > **operator\*** (const std::vector< std::vector< T > > &v1, const std::vector<  
std::vector< T > > &v2)
- template<class T >  
std::vector< T > **operator\*** (const std::vector< std::vector< T > > &v1, const std::vector< T > &v2)
- template<class T >  
std::vector< std::vector< T > > **operator\*** (const std::vector< T > &v1, const std::vector< T > &v2)
- template<class T >  
std::vector< T > **operator\*** (const std::vector< T > &v1, float d)
- template<class T >  
std::vector< std::vector< T > > **operator+** (const std::vector< std::vector< T > > &v1, const std::vector<  
std::vector< T > > &v2)
- template<class T >  
std::vector< std::vector< T > > **operator+** (const std::vector< std::vector< T > > &v1, float d)
- template<class T >  
std::vector< T > **operator+** (const std::vector< T > &v1, const std::vector< T > &v2)
- template<class T >  
std::vector< T > **operator-** (const std::vector< T > &v1)
- template<class T >  
std::vector< T > **operator-** (const std::vector< T > &v1, const std::vector< T > &v2)
- template<class T >  
std::vector< std::vector< T > > **operator/** (const std::vector< std::vector< T > > &v1, int divisor)
- template<class T >  
std::vector< T > **operator/** (const std::vector< T > &v1, int divisor)
- bool **operator<** (const [ThumbnailIdentifier](#) &a, const [ThumbnailIdentifier](#) &b)
- QDataStream & **operator<<** (QDataStream &ds, const [CamItemInfo](#) &info)
- QDataStream & **operator<<** (QDataStream &ds, const [PhotoInfoContainer](#) &info)
- QDataStream & **operator<<** (QDataStream &ds, const [VideoInfoContainer](#) &info)
- QDebug **operator<<** (QDebug dbg, const [BatchToolSet](#) &s)  
*QDebug() stream operator. Writes property t to the debug output in a nicely formatted way.*



- QDebug **operator**<< (QDebug dbg, const [CamItemInfo](#) &info)  
*QDebug()* stream operator. Writes property info to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [CaptionValues](#) &val)  
*QDebug()* stream operator. Writes values val to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [DbEngineParameters](#) &p)
- QDebug **operator**<< (QDebug dbg, const [DMetadataSettingsContainer](#) &inf)  
*QDebug()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [DRawDecoderSettings](#) &s)  
*QDebug()* stream operator. Writes settings s to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [DRawInfo](#) &c)  
*QDebug()* stream operator. Writes container c to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [DTrashItemInfo](#) &info)  
*QDebug()* stream operator. Writes property info to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [FaceTagsIface](#) &f)
- QDebug **operator**<< (QDebug dbg, const [FocusPoint](#) &fp)  
*QDebug()* stream operator. Writes [FocusPoint](#) to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [GeolocationSettingsContainer](#) &inf)  
*QDebug(DIGIKAM\_GEOENGINE\_LOG) << QString::fromUtf8()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [HistoryImageId](#) &id)
- QDebug **operator**<< (QDebug dbg, const [HistoryVertexProperties](#) &props)
- QDebug **operator**<< (QDebug dbg, const [ImageQualityContainer](#) &s)  
*QDebug()* stream operator. Writes property s to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [IptcCoreContactInfo](#) &inf)  
*QDebug()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [IptcCoreLocationInfo](#) &inf)  
*QDebug()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [ItemHistoryGraph](#) &g)
- QDebug **operator**<< (QDebug dbg, const [LocalizeContainer](#) &inf)  
*QDebug()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [MaintenanceSettings](#) &s)  
*QDebug()* stream operator. Writes property s to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [MetaEngineSettingsContainer](#) &inf)  
*QDebug()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [NamespaceEntry](#) &inf)  
*QDebug()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [NRContainer](#) &inf)  
*QDebug()* stream operator. Writes property inf to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [PhotoInfoContainer](#) &t)  
*QDebug()* stream operator. Writes property t to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [TagRegion](#) &r)
- QDebug **operator**<< (QDebug dbg, const [Template](#) &t)  
*QDebug()* stream operator. Writes [Template](#) to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug dbg, const [VideoInfoContainer](#) &t)  
*QDebug()* stream operator. Writes property t to the debug output in a nicely formatted way.
- QDebug **operator**<< (QDebug stream, const [ItemInfo](#) &info)  
*QDebug()* stream operator. Writes property info to the debug output in a nicely formatted way.
- bool **operator**== (const [SearchTextSettings](#) &a, const [SearchTextSettings](#) &b)
- QDataStream & **operator**>> (QDataStream &ds, [CamItemInfo](#) &info)
- QDataStream & **operator**>> (QDataStream &ds, [PhotoInfoContainer](#) &info)
- QDataStream & **operator**>> (QDataStream &ds, [VideoInfoContainer](#) &info)
- [FocusPoint::TypePoint](#) **operator**| ([FocusPoint::TypePoint](#) type1, [FocusPoint::TypePoint](#) type2)

- [FocusPoint::TypePoint](#) & **operator|=** ([FocusPoint::TypePoint](#) &type1, [FocusPoint::TypePoint](#) type2)
- **Q\_GLOBAL\_STATIC\_WITH\_ARGS** ([DbEngineConfigSettingsLoader](#), dbcoreloader,(QStandardPaths::locate(QStandardPaths::GenericDataLocation, QLatin1String("digikam/database/dbconfig.xml")), dbcoreconfig←\_xml\_version)) [DbEngineConfigSettings](#) [DbEngineConfig](#)
- `size_t` **qHash** (const [CollectionLocation](#) &loc)
- `QT_HASH_TYPE` **qHash** (const [Digikam::AlbumCopyMoveHint](#) &hint)
- `size_t` **qHash** (const [ItemInfo](#) &info)
- `size_t` **qHash** (const [ItemLISTERRecord](#) &key)
- `size_t` **qHash** (const [NewlyAppearedFile](#) &file)
- `size_t` **qHash** (const [PAlbumPath](#) &id)
- `int` [QPointSquareDistance](#) (const [QPoint](#) &a, const [QPoint](#) &b)  
*Helper function, returns the square of the distance between two points.*
- `template<typename T, class Container >`  
`void` **removeAnyInInterval** ([Container](#) &list, const T &begin, const T &end)
- `LqrRetVal` **s\_carverProgressEnd** (const `gchar` \*end\_message)
- `LqrRetVal` **s\_carverProgressInit** (const `gchar` \*init\_message)
- `LqrRetVal` **s\_carverProgressUpdate** (`gdouble` percentage)
- `bool` **s\_checkSolidCamera** (const [Solid::Device](#) &cameraDevice)
- `bool` **s\_dmcompare** (const [NamespaceEntry](#) &e1, const [NamespaceEntry](#) &e2)
- `void` **s\_exifParserCallbackForLibRaw** (`void` \*context, `int` tag, `int` type, `int` len, `unsigned int` ord, `void` \*ifp, `INT64` base)
- `bool` [s\\_inlineTranslateString](#) (const `QString` &text, const `QString` &trCode, `QString` &tr, `QString` &error)
- `bool` **s\_isHeifSuccess** (const `struct heif_error` \*const error)
- `QStringList` **s\_keywordsSeparation** (const `QString` &data)
- `QString` **s\_labelForSolidCamera** (const [Solid::Device](#) &cameraDevice)
- `bool` **s\_modificationDateEquals** (const `QDateTime` &a, const `QDateTime` &b)
- `int` **s\_progressCallbackForLibRaw** (`void` \*context, `enum LibRaw_progress` p, `int` iteration, `int` expected)
- `QString` **s\_rawFileExtensions** ()
- `QMap< QString, QString >` **s\_rawFileExtensionsdWithDesc** ()
- `int` **s\_rawFileExtensionsVersion** ()
- `void` **s\_readHEICMetadata** (`struct heif_context` \*const heif\_context, `heif_item_id` image\_id, `QByteArray` &exif, `QByteArray` &iptc, `QByteArray` &xmp)
- `qint64` **s\_secondsSinceJanuary1904** (const `QDateTime` &dt)
- `QString` **s\_setXmpTagStringFromEntry** (const [DMetadata](#) \*const meta, const `QStringList` &lst, const [DMetadata::MetaDatumMap](#) &map, const `QStringList` &xmpTags=`QStringList`())
- `void` **setComponentValue** (`QColor` &color, `DColorChooserMode` chooserMode, `qreal` value)
- `bool` **setExifXmpTagDataVariant** ([DMetadata](#) \*const meta, const `char` \*const exifTagName, const `char` \*const xmpTagName, const `QVariant` &value)
- `DIGIKAM_EXPORT` `void` **setMacOSEnvironment** ()
- `void` **setOpenCLEnvironment** (`bool` b)
- `DIGIKAM_EXPORT` `void` **setWindowsEnvironment** ([QApplication](#) &app)
- `void` **showDigikamComponentsInfo** ()
- `void` **showDigikamDatabaseStat** ()
- `void` **showRawCameraList** ()
- `QDateTime` **startOfDay** (const `QDate` &date)
- `QStringList` **supportedImageMimeTypes** ([QIODevice::OpenModeFlag](#) mode, `QString` &allTypes)
- `QString` **toolButtonStyleSheet** ()
- `bool` **TrackCorrelationLessThan** (const [TrackCorrelator::Correlation](#) &a, const [TrackCorrelator::Correlation](#) &b)
- `DIGIKAM_EXPORT` `void` **unloadQtTranslationFiles** ([QApplication](#) &app)

## Variables

- auto [accessCol](#)
- auto [accessRow](#)
- ImageCurves::CRMatrix [CR\\_basis](#)
- const int **DNN\_MODEL\_THRESHOLD\_NOT\_SET** = 1000
- const std::map< [FaceScanSettings::FaceDetectionSize](#), int > [faceenum2size](#)
- const int **FilterChunkSize** = 2001
- const int [GeofaceMinMarkerGroupingRadius](#) = 1
- const int **GeofaceMinThumbnailGroupingRadius** = 15
- const int **GeofaceMinThumbnailSize** = [GeofaceMinThumbnailGroupingRadius](#) \* 2
- const int **MAX\_MATRIX\_SIZE** = 25
- const int **PrepareChunkSize** = 101
- const float **RATIO\_POINT\_IMAGE** = 1 / 120
- const int **RoleGPSItemInfo** = Qt::UserRole + 1
- bool **s\_hResize** = false
- QRecursiveMutex [s\\_metaEngineMutex](#)  
*Mutex to fix no re-entrancy from Exiv2.*
- bool [s\\_metaEngineSupportBmff](#) = false  
*Flag for Exiv2 Base Media File Format support.*
- bool [s\\_metaEngineWarnOrError](#) = false  
*Flag for Exiv2 has printed a warning or error.*
- [ContentAwareFilter](#) \* **s\_resiser** = nullptr
- const LanguageCodeMap [s\\_rfc3066ForXMP](#)
- bool [s\\_stage](#) = false
- bool **s\_wResize** = false
- const int **SIZE\_FILTER** = 4

### 5.1.1 Detailed Description

NOTE: This is because of the [CollectionManager](#) private slot.

References about DNG: DNG SDK tutorial: [www.adobeforums.com/webx/.3bc2944e](http://www.adobeforums.com/webx/.3bc2944e) [www.adobeforums.com/webx/.3c054bde](http://www.adobeforums.com/webx/.3c054bde) DNG review: [www.barrypearson.co.uk/articles/dng/index.htm](http://www.barrypearson.co.uk/articles/dng/index.htm) DNG intro: [www.adobe.com/digitalimag/pdfs/dng\\_primer.pdf](http://www.adobe.com/digitalimag/pdfs/dng_primer.pdf) [www.adobe.com/products/dng/pdfs/DNG\\_primer\\_manufacturers.pdf](http://www.adobe.com/products/dng/pdfs/DNG_primer_manufacturers.pdf) DNG Specification: [www.images.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng\\_spec\\_1.5.0.0.pdf](http://www.images.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng_spec_1.5.0.0.pdf) TIFF/EP Spec.: [www.map.tu.chiba-u.ac.jp/IEC/100/TA2/reccdoc/N4378.pdf](http://www.map.tu.chiba-u.ac.jp/IEC/100/TA2/reccdoc/N4378.pdf) DNG SDK reference: [www.thomasdideriksen.dk/misc/File%20Formats/dng\\_sdk\\_refman.pdf](http://www.thomasdideriksen.dk/misc/File%20Formats/dng_sdk_refman.pdf) DNG SDK tarball: [helpx.adobe.com/photoshop/digital-negative.html::dng\\_sdk\\_download](http://helpx.adobe.com/photoshop/digital-negative.html::dng_sdk_download) DNG users forum: [www.adobeforums.com/webx/.3bb5f0ec](http://www.adobeforums.com/webx/.3bb5f0ec)

Applications using DNG SDK: DNG4PS2: [dng4ps2.chat.ru/index\\_en.html](http://dng4ps2.chat.ru/index_en.html) CORNERFIX: [sourceforge.net/projects/cornerfix](http://sourceforge.net/projects/cornerfix) ADOBE DNG CONVERTER: [helpx.adobe.com/photoshop/using/adobe-dng-converter.html](http://helpx.adobe.com/photoshop/using/adobe-dng-converter.html) DNGCONVERT: [github.com/jmuedngconvert](https://github.com/jmuedngconvert) MOVIE2DNG: [elphel.svn.sourceforge.net/svnroot/elphel/tools/](http://elphel.svn.sourceforge.net/svnroot/elphel/tools/) Movie2DNG RAW2DNG: [github.com/Fimagera/raw2dng](https://github.com/Fimagera/raw2dng)

NOTE: Good explanations about GPS (in French) can be found at this url : [www.gpspassion.com/forumsen/topic.asp?TOPIC\\_ID=16593](http://www.gpspassion.com/forumsen/topic.asp?TOPIC_ID=16593)

### 5.1.2 Typedef Documentation

#### 5.1.2.1 ActionJobCollection

```
typedef QHash<ActionJob*, int> Digikam::ActionJobCollection
```

Define a QHash of job/priority to process by [ActionThreadBase](#) manager. Priority value can be used to control the run queue's order of execution. Zero priority want mean to process job with higher priority.

### 5.1.2.2 BatchSetList

```
typedef QList<BatchToolSet> Digikam::BatchSetList
```

An indexed map of batch tools with settings.

### 5.1.2.3 BatchToolSettings

```
typedef QMap<QString, QVariant> Digikam::BatchToolSettings
```

A map of batch tool settings (setting key, setting value).

### 5.1.2.4 BatchToolsList

```
typedef QList<BatchTool*> Digikam::BatchToolsList
```

A list of batch tool instances.

### 5.1.2.5 DateRange

```
typedef QPair<QDateTime, QDateTime> Digikam::DateRange
```

Range of a contiguous dates selection <start date, end date>.

### 5.1.2.6 DateRangeList

```
typedef QList<DateRange> Digikam::DateRangeList
```

List of dates range selected.

### 5.1.2.7 DImgLoaderPrms

```
typedef QMap<QString, QVariant> Digikam::DImgLoaderPrms
```

Map container of widget parameter name/value.

### 5.1.2.8 DItemsListIsLessThanHandler

```
typedef bool(* Digikam::DItemsListIsLessThanHandler) (const QTreeWidgetItem *current, const
QTreeWidgetItem &other)
```

Type of static function used to customize sort items in list. Sort items call this method in `DItemsListViewItem::operator<`. To setup this method, uses `DItemList::setIsLessThanHandler()`.

### 5.1.2.9 QueuePoolItemsList

```
typedef QList<ItemInfoSet> Digikam::QueuePoolItemsList
```

A list of all queued items from the pool.

## 5.1.3 Enumeration Type Documentation

### 5.1.3.1 DetectorModel

```
enum Digikam::DetectorModel
```

## Enumerator

YOLOV5NANO	YOLO nano neural network model.
YOLOV5XLARGE	YOLO large neural network model.

## 5.1.3.2 DetectorNNModel

```
enum Digikam::DetectorNNModel
```

## Enumerator

DNNDetectorSSD	SSD MobileNet neural network inference [ <a href="https://github.com/arunponnusamy/cvlib">https://github.com/arunponnusamy/cvlib</a> ].
DNNDetectorYOLOv3	YOLO neural network inference [ <a href="https://github.com/sthanhng/yoloface">https://github.com/sthanhng/yoloface</a> ].
DNNDetectorYuNet	YuNet neural network inference [ <a href="https://github.com/opencv/opencv_zoo/tree/main">https://github.com/opencv/opencv_zoo/tree/main</a> ].

## 5.1.3.3 FullScreenOptions

```
enum Digikam::FullScreenOptions
```

Optional parts which can be hidden or not from managed window configuration panel

## Enumerator

FS_TOOLBARS	Manage Tools bar in full-screen mode.
FS_THUMBBAR	Manage Thumb bar in full-screen mode.
FS_SIDEBARS	Manage Side bars in full-screen mode.
FS_STATUSBAR	Manage Status bar in full-screen mode.
FS_NONE	No full-screen options.
FS_ALBUMGUI	<a href="#">Album</a> GUI Config.
FS_EDITOR	Image Editor Config.
FS_LIGHTTABLE	Light Table Config.
FS_IMPORTUI	Import UI Config.

## 5.1.3.4 GeoGroupStateEnum

```
enum Digikam::GeoGroupStateEnum
```

The idea is that a group consists of more than one object. Thus the resulting state is that either none of the objects, some or all of them have a certain state. The constants for each state are set up such that they can be logically or'ed: If a group has the state `___All`, and another the state `___Some`, the bit representing `___Some` is always propagated along. You only have to make sure that once you reach an object with `___None`, and the computed state is `___All`, to set the `___Some` bit.

Selected`___`: An object is selected. FilteredPositive`___`: An object was highlighted by a filter. This usually means that not-positively-filtered objects should be hidden. RegionSelected`___`: An object is inside a region of interest on the map.

### 5.1.3.5 HistogramRenderingType

enum `Digikam::HistogramRenderingType`

#### Enumerator

FullImageHistogram	Full image histogram rendering.
ImageSelectionHistogram	Image selection histogram rendering.

### 5.1.3.6 HistogramScale

enum `Digikam::HistogramScale`

#### Enumerator

LinScaleHistogram	Linear scale.
LogScaleHistogram	Logarithmic scale.

### 5.1.3.7 HudSide

enum `Digikam::HudSide`

#### Enumerator

HS_None	Special value used to avoid initial animation.
---------	--

### 5.1.3.8 MeaningOfDirection

enum `Digikam::MeaningOfDirection`

Each edge is directed: "vertex1 -> vertex2". This direction has a meaning with methods such as roots() or leaves().

#### Enumerator

ParentToChild	Edges are directed from a parent to its child.
ChildToParent	Edges are direct from a child to its parent.

### 5.1.3.9 OperationType

enum `Digikam::OperationType`

Types of operations for [ApplicationSettings](#). Originally introduced for grouping to configure whether an operation should be done on all group members or only it's head.

## Enumerator

UnspecifiedOps	This element must always come last.
----------------	-------------------------------------

## 5.1.3.10 YoloVersions

```
enum class Digikam::YoloVersions [strong]
```

## Enumerator

YOLOV5NANO	yolov5n_batch_16_s320.onnx
YOLOV5XLARGE	yolov5x_batch_16_s320.onnx
RESNET50	resnet50.onnx

## 5.1.4 Function Documentation

## 5.1.4.1 adjustedEnvironmentForAppImage()

```
QProcessEnvironment Digikam::adjustedEnvironmentForAppImage ( )
```

If digiKam run into AppImage, return a cleaned environment for QProcess to execute a program outside the bundle without broken run-time dependencies. Use case : system based Hugin CLI tools called by Panorama wizard. If digiKam do not run as AppImage bundle, this method return a QProcessEnvironment instance based on system environment.

## 5.1.4.2 asDateTimeLocal()

```
DIGIKAM_EXPORT QDateTime Digikam::asDateTimeLocal (
    const QDateTime & dt )
```

This method returns QDateTime with Local timespec.

## 5.1.4.3 asDateTimeUTC()

```
DIGIKAM_EXPORT QDateTime Digikam::asDateTimeUTC (
    const QDateTime & dt )
```

This method returns QDateTime with UTC timespec.

## 5.1.4.4 coordinatesToClipboard()

```
void DIGIKAM_EXPORT Digikam::coordinatesToClipboard (
    const GeoCoordinates & coordinates,
    const QUrl & url,
    const QString & title )
```

NOTE: importing this representation into Marble does not show anything, but Merkaartor shows the point

importing this data into Marble and Merkaartor works

#### 5.1.4.5 DatabaseImageMetadataFieldsToMetadataInfoField()

```
MetadataInfo::Field Digikam::DatabaseImageMetadataFieldsToMetadataInfoField (
    const DatabaseFields::ImageMetadata imageMetadataField )
```

#### 5.1.4.6 DatabaseVideoMetadataFieldsToMetadataInfoField()

```
MetadataInfo::Field Digikam::DatabaseVideoMetadataFieldsToMetadataInfoField (
    const DatabaseFields::VideoMetadata videoMetadataField )
```

#### 5.1.4.7 defineShortcut()

```
DIGIKAM_EXPORT QShortcut * Digikam::defineShortcut (
    QWidget *const w,
    const QKeySequence & key,
    const QObject * receiver,
    const char * slot )
```

Convenience method for creating keyboard shortcuts.

#### 5.1.4.8 DNotificationWrapper()

```
void DIGIKAM_EXPORT Digikam::DNotificationWrapper (
    const QString & eventId,
    const QString & message,
    QWidget *const parent,
    const QString & windowTitle,
    const QPixmap & pixmap = QPixmap() )
```

##### Parameters

<i>eventId</i>	Event id for this notification, KNotification::Notification is used if this is empty. Events have to be configured in digikam.notifyrc
<i>message</i>	Message to display
<i>parent</i>	Widget which owns the notification
<i>windowTitle</i>	Title of the notification window (only used for KPassivePopup)
<i>pixmap</i>	Pixmap to show in the notification, in addition to the digikam logo.

#### 5.1.4.9 GeofaceHelperParseLatLonString()

```
DIGIKAM_EXPORT bool Digikam::GeoIfaceHelperParseLatLonString (
    const QString & latLonString,
    GeoCoordinates *const coordinates )
```

helper functions

##### Returns

true if the string could be parsed successfully



#### 5.1.4.10 `getComponentValue()`

```
qreal Digikam::getComponentValue (
    const QColor & color,
    DColorChooserMode chooserMode )
```

get/set color component

#### 5.1.4.11 `image2Mat()`

```
cv::Mat Digikam::image2Mat (
    const QImage & img,
    int requiredMatType,
    MatColorOrder requiriedOrder )
```

Convert QImage to cv::Mat

Convert QImage to/from cv::Mat

- cv::Mat
  - Supported channels
    - \* 1 channel
    - \* 3 channels (B G R), (R G B)
    - \* 4 channels (B G R A), (R G B A), (A R G B)
  - Supported depth
    - \* CV\_8U [0, 255]
    - \* CV\_16U [0, 65535]
    - \* CV\_32F [0, 1.0]
- QImage
  - All of the formats of QImage are supported.

#### 5.1.4.12 `image2Mat_shared()`

```
cv::Mat Digikam::image2Mat_shared (
    const QImage & img,
    MatColorOrder *const order )
```

Convert QImage to cv::Mat without data copy

Convert QImage to/from cv::Mat without data copy

- Supported QImage formats and cv::Mat types are:
  - QImage::Format\_Indexed8 <==> CV\_8UC1
  - QImage::Format\_Alpha8 <==> CV\_8UC1
  - QImage::Format\_Grayscale8 <==> CV\_8UC1
  - QImage::Format\_RGB888 <==> CV\_8UC3 (R G B)
  - QImage::Format\_RGB32 <==> CV\_8UC4 (A R G B or B G R A)

- QImage::Format\_ARGB32 <==> CV\_8UC4 (A R G B or B G R A)
  - QImage::Format\_ARGB32\_Premultiplied <==> CV\_8UC4 (A R G B or B G R A)
  - QImage::Format\_RGBX8888 <==> CV\_8UC4 (R G B A)
  - QImage::Format\_RGBA8888 <==> CV\_8UC4 (R G B A)
  - QImage::Format\_RGBA8888\_Premultiplied <==> CV\_8UC4 (R G B A)
- For QImage::Format\_RGB32 ,QImage::Format\_ARGB32 and QImage::Format\_ARGB32\_Premultiplied, the color channel order of cv::Mat will be (B G R A) in little endian system or (A R G B) in big endian system.
  - User must make sure that the color channels order is the same as the color channels order required by QImage.

#### 5.1.4.13 installQtTranslationFiles()

```
void Digikam::installQtTranslationFiles (
    QApplication & app )
```

For bundles only, main function to manage all Qt translation files at run-time in application instance.

#### 5.1.4.14 isReadableImageFile()

```
DIGIKAM_EXPORT bool Digikam::isReadableImageFile (
    const QString & filePath )
```

Return true if filePath is an image readable by application for thumbnail, preview, or edit.

#### 5.1.4.15 isRunningInApplmageBundle()

```
bool Digikam::isRunningInAppImageBundle ( )
```

Return true if application run in Applmage bundle.

#### 5.1.4.16 isRunningOnNativeKDE()

```
bool Digikam::isRunningOnNativeKDE ( )
```

Return true if application run on native KDE desktop.

#### 5.1.4.17 layoutMargin()

```
DIGIKAM_EXPORT int Digikam::layoutMargin ( )
```

Default margin to use in layout.

#### 5.1.4.18 layoutSpacing()

```
DIGIKAM_EXPORT int Digikam::layoutSpacing ( )
```

Default spacing to use in layout.

**5.1.4.19 loadEcmQtTranslationFiles()**

```
void Digikam::loadEcmQtTranslationFiles (
    QApplication & app )
```

For bundles only, load ECM Qt translation files at run-time in application instance.

**5.1.4.20 loadStdQtTranslationFiles()**

```
void Digikam::loadStdQtTranslationFiles (
    QApplication & app )
```

For bundles only, load standard Qt translation files at run-time in application instance.

**5.1.4.21 macOSBundlePrefix()**

```
QString Digikam::macOSBundlePrefix ( )
```

Prefix of macOS Bundle to access to internal Unix hierarchy.

**5.1.4.22 mat2Image()**

```
QImage Digikam::mat2Image (
    const cv::Mat & mat,
    MatColorOrder order,
    QImage::Format formatHint )
```

Convert cv::Mat to QImage

**5.1.4.23 mat2Image\_shared()**

```
QImage Digikam::mat2Image_shared (
    const cv::Mat & mat,
    QImage::Format formatHint )
```

Convert cv::Mat to QImage without data copy

**5.1.4.24 openOnlineDocumentation()**

```
DIGIKAM_EXPORT void Digikam::openOnlineDocumentation (
    const QString & section = QString(),
    const QString & chapter = QString(),
    const QString & reference = QString() )
```

Open online handbook at the section/chapter/reference page.

if section and chapter and reference are empty, fromt page is open. ( [https://en.wikipedia.org/wiki/Matrix\\_\(protocol\)#Bridges](https://en.wikipedia.org/wiki/Matrix_(protocol)#Bridges)) if only chapter and reference are empty, section page is open. (as: [https://docs.digikam.org/en/main\\_window.html](https://docs.digikam.org/en/main_window.html)) if only reference is empty, chapter from section page is open. (as: [https://docs.digikam.org/en/main\\_window/people\\_view.html](https://docs.digikam.org/en/main_window/people_view.html)) else reference at chapter from section page is open. (as: [https://docs.digikam.org/en/main\\_window/people\\_view.html#face-recognition](https://docs.digikam.org/en/main_window/people_view.html#face-recognition))

#### 5.1.4.25 operator<<()

```
QDebug Digikam::operator<< (
    QDebug dbg,
    const MaintenanceSettings & s )
```

QDebug(DIGIKAM\_GENERAL\_LOG) stream operator. Writes property *s* to the debug output in a nicely formatted way.

#### 5.1.4.26 operator" | ()

```
FocusPoint::TypePoint Digikam::operator| (
    FocusPoint::TypePoint type1,
    FocusPoint::TypePoint type2 ) [inline]
```

Boolean Operators over TypePoint type.

#### 5.1.4.27 qHash()

```
size_t Digikam::qHash (
    const ItemListerRecord & key ) [inline]
```

Used by QSet.

#### 5.1.4.28 QPointSquareDistance()

```
DIGIKAM_EXPORT int Digikam::QPointSquareDistance (
    const QPoint & a,
    const QPoint & b )
```

##### Parameters

<i>a</i>	Point a
<i>b</i>	Point b

##### Returns

Square of the distance between a and b

#### 5.1.4.29 s\_inlineTranslateString()

```
bool DIGIKAM_EXPORT Digikam::s_inlineTranslateString (
    const QString & text,
    const QString & trCode,
    QString & tr,
    QString & error )
```

Helper re-entrant static method to translate a string with online translator. Language from string is auto-detected, and target language is specified to 'trCode'. If string can be processed, translation is returned to 'tr' and function return true, else false is returned with a dysfunction description in 'error'.

**5.1.4.30 s\_rawFileExtensionsdWithDesc()**

```
DIGIKAM_EXPORT QMap< QString, QString > Digikam::s_rawFileExtensionsdWithDesc ( )
```

NOTE: extension list Version 1 and 2 are taken from [www.cybercom.net/~dcoffin/dcrawl/rawphoto.c](http://www.cybercom.net/~dcoffin/dcrawl/rawphoto.c)

Ext	Descriptions From
	<a href="http://www.file-extensions.org">www.file-extensions.org</a>
	<a href="http://en.wikipedia.org/wiki/RAW_file_format">en.wikipedia.org/wiki/RAW_file_format</a>
	<a href="http://filext.com">filext.com</a>

NOTE: VERSION 1

These images are based on the TIFF image standard.

For these models: Kodak DSC Pro SLR/c, Kodak DSC Pro SLR/n, Kodak DSC Pro 14N, Kodak DSC PRO 14nx.

DNG is publicly available archival format for the raw files generated by digital cameras. By addressing the lack of an open standard for the raw files created by individual camera models, DNG helps ensure that photographers will be able to access their files in the future.

For DSC-F828 8 megapixel digital camera or Sony DSC-R1.

For devices based on Foveon X3 direct image sensor.

For Alpha devices.

NOTE: VERSION 2

NOTE: VERSION 3

NOTE: VERSION 4

NOTE: VERSION 5

NOTE: VERSION 6

NOTE: VERSION 7

NOTE: VERSION 8

**5.1.4.31 s\_rawFileExtensionsVersion()**

```
DIGIKAM_EXPORT int Digikam::s_rawFileExtensionsVersion ( )
```

NOTE: increment this number whenever you change the above strings

**5.1.4.32 s\_setXmpTagStringFromEntry()**

```
QString Digikam::s_setXmpTagStringFromEntry (
    const DMetadata *const meta,
    const QStringList & lst,
    const DMetadata::MetaDataMap & map,
    const QStringList & xmpTags = QStringList() )
```

Search first occurrence of string in 'map' with keys given by 'lst'. Return the string match. If 'xmpTags' is not empty, register XMP tags value with string.

**5.1.4.33 setExifXmpTagDataVariant()**

```
bool Digikam::setExifXmpTagDataVariant (
    DMetadata *const meta,
    const char *const exifTagName,
    const char *const xmpTagName,
    const QVariant & value )
```

**5.1.4.34 setMacOSEnvironment()**

```
void Digikam::setMacOSEnvironment ( )
```

For MacOS bundles only, set necessary MacOS environment variables

**5.1.4.35 setOpenCLEnvironment()**

```
DIGIKAM_EXPORT void Digikam::setOpenCLEnvironment (
    bool b )
```

For OpenCV framework, set necessary environment variables to use OpenCL features.

**5.1.4.36 setWindowsEnvironment()**

```
void Digikam::setWindowsEnvironment (
    QApplication & app )
```

For Windows only, set necessary Windows environment variables

**5.1.4.37 showRawCameraList()**

```
DIGIKAM_EXPORT void Digikam::showRawCameraList ( )
```

Show a dialog with all RAW camera supported by digiKam, through libraw.

**5.1.4.38 startOfDay()**

```
DIGIKAM_EXPORT QDateTime Digikam::startOfDay (
    const QDate & date )
```

This method returns QDateTime from with date set to parameter date and time set to start of the day.

**5.1.4.39 supportedImageMimeTypes()**

```
DIGIKAM_EXPORT QStringList Digikam::supportedImageMimeTypes (
    QIODevice::OpenModeFlag mode,
    QString & allTypes )
```

Return list of supported image formats by Qt for reading or writing operations if suitable container used by QFile↔ Dialog. For simple container of type mime, use 'allTypes' string. Supported modes are QIODevice::ReadOnly, QIODevice::WriteOnly, and QIODevice::ReadWrite.

#### 5.1.4.40 toolButtonStyleSheet()

```
DIGIKAM_EXPORT QString Digikam::toolButtonStyleSheet ( )
```

Style sheet for transparent QToolButtons over image and video preview.

#### 5.1.4.41 unloadQtTranslationFiles()

```
void Digikam::unloadQtTranslationFiles (
    QApplication & app )
```

For bundles only, unload all Qt translation files at run-time in application instance.

### 5.1.5 Variable Documentation

#### 5.1.5.1 accessCol

```
auto Digikam::accessCol
```

##### Initial value:

```
= [] (const cv::Mat& mat)
{
    return [mat](int index)
    {
        return mat.col(index);
    };
}
```

#### 5.1.5.2 accessRow

```
auto Digikam::accessRow
```

##### Initial value:

```
= [] (const cv::Mat& mat)
{
    return [mat](int index)
    {
        return mat.row(index);
    };
}
```

#### 5.1.5.3 CR\_basis

```
ImageCurves::CRMatrix Digikam::CR_basis
```

##### Initial value:

```
=
{
    { -0.5, 1.5, -1.5, 0.5 },
    { 1.0, -2.5, 2.0, -0.5 },
    { -0.5, 0.0, 0.5, 0.0 },
    { 0.0, 1.0, 0.0, 0.0 },
}
```

#### 5.1.5.4 faceenum2size

```
const std::map<FaceScanSettings::FaceDetectionSize, int> Digikam::faceenum2size
```

##### Initial value:

```
{
  { FaceScanSettings::FaceDetectionSize::ExtraLarge, 420 },
  { FaceScanSettings::FaceDetectionSize::Large,      620 },
  { FaceScanSettings::FaceDetectionSize::Medium,    800 },
  { FaceScanSettings::FaceDetectionSize::Small,    1200 },
  { FaceScanSettings::FaceDetectionSize::ExtraSmall, 2000 }
}
```

#### 5.1.5.5 GeolfaceMinMarkerGroupingRadius

```
const int Digikam::GeoIfaceMinMarkerGroupingRadius = 1
```

#### 5.1.5.6 s\_metaEngineMutex

```
QRecursiveMutex Digikam::s_metaEngineMutex
```

This mutex is used to protect all Exiv2 API calls when [MetaEngine](#) is used with multi-threads.

#### 5.1.5.7 s\_metaEngineSupportBmff

```
bool Digikam::s_metaEngineSupportBmff = false
```

Boolean value about Bmff based file support (CR3, HEIF, HEIC, and AVIF). Initialized at run time by `initializeExiv2()`.

#### 5.1.5.8 s\_metaEngineWarnOrError

```
bool Digikam::s_metaEngineWarnOrError = false
```

Boolean value about Exiv2 warning or error in the `Exiv2::LogMsg`. Changed in `MetaEngine::Private::printExiv2MessageHandler`.

#### 5.1.5.9 s\_rfc3066ForXMP

```
const LanguageCodeMap Digikam::s_rfc3066ForXMP
```

##### Warning

: We cannot use `KLocale::allLanguagesList()` here because KDE framework only support 2 characters country codes. XMP require 2+2 characters language+country following RFC 3066.

The format is based on xx-XX using ISO-639 two-letter code with ISO-3166 two-letter country code.

List version 1 : [babelwiki.babelzilla.org/index.php?title=Language\\_codes](http://babelwiki.babelzilla.org/index.php?title=Language_codes) List version 2 update: [coverpages.org/TexinUsingLangID.html](http://coverpages.org/TexinUsingLangID.html) List version 3 update: List of ISO 639-1 codes ( [https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)) List of ISO 639-2 codes ( [https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-2\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-2_codes)) List of ISO 639 macrolanguage ( [https://en.wikipedia.org/wiki/ISO\\_639\\_macrolanguage](https://en.wikipedia.org/wiki/ISO_639_macrolanguage))

##### Note

: this lists of RFC3066 map is also used in conversion map from [DOnlineTranslator](#) class. Any changes here must be also backported to [DOnlineTranslator](#) class.



### 5.1.5.10 s\_stage

```
bool Digikam::s_stage = false
```

Resizement is decomposed in 2 stages: horizontal and vertical.

## 5.2 Digikam::Matrix Namespace Reference

### Functions

- [MetaEngineRotation matrix](#) ([MetaEngine::ImageOrientation](#) exifOrientation)
- [MetaEngineRotation matrix](#) ([MetaEngineRotation::TransformationAction](#) action)

### 5.2.1 Detailed Description

If the picture is displayed according to the exif orientation tag, the user will request rotating operations relative to what he sees, and that is the picture rotated according to the EXIF tag. So the operation requested and the given EXIF angle must be combined. E.g. if orientation is "6" (rotate 90 clockwise to show correctly) and the user selects 180 clockwise, the operation is 270. If the user selected 270, the operation would be None (and clearing the exif tag).

This requires to describe the transformations in a model which cares for both composing ( $180+90=270$ ) and eliminating ( $180+180=no\ action$ ), as well as the non-commutative nature of the operations ( $vflip+90$  is not  $90+vflip$ )

All 2D transformations can be described by a 2x3 matrix, see [QWMetaEngineRotation](#). All transformations needed here - rotate 90, 180, 270, flipV, flipH - can be described in a 2x2 matrix with the values 0,1,-1 (because flipping is expressed by changing the sign only, and sine and cosine of 90, 180 and 270 are either 0,1 or -1).

$$x' = m11 x + m12 y \quad y' = m21 x + m22 y$$

Moreover, all combinations of these rotate/flip operations result in one of the eight matrices defined below. This did not proof that mathematically, but empirically.

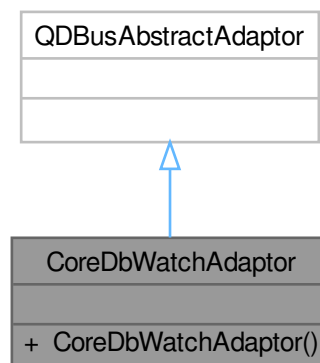


# Chapter 6

## Class Documentation

### 6.1 CoreDbWatchAdaptor Class Reference

Inheritance diagram for CoreDbWatchAdaptor:



#### Signals

- void **signalAlbumChangeDBus** (const QString &databaseldentifier, const QString &applicationIdentifier, const [Digikam::AlbumChangeset](#) &changeset)
- void **signalAlbumRootChangeDBus** (const QString &databaseldentifier, const QString &applicationIdentifier, const [Digikam::AlbumRootChangeset](#) &changeset)
- void **signalCollectionImageChangeDBus** (const QString &databaseldentifier, const QString &applicationIdentifier, const [Digikam::CollectionImageChangeset](#) &changeset)
- void **signalImageChangeDBus** (const QString &databaseldentifier, const QString &applicationIdentifier, const [Digikam::ImageChangeset](#) &changeset)
- void **signalImageTagChangeDBus** (const QString &databaseldentifier, const QString &applicationIdentifier, const [Digikam::ImageTagChangeset](#) &changeset)
- void **signalSearchChangeDBus** (const QString &databaseldentifier, const QString &applicationIdentifier, const [Digikam::SearchChangeset](#) &changeset)
- void **signalTagChangeDBus** (const QString &databaseldentifier, const QString &applicationIdentifier, const [Digikam::TagChangeset](#) &changeset)

Public Member Functions

- CoreDbWatchAdaptor ([Digikam::CoreDbWatch](#) \*const watch)

## 6.2 Digikam::AbstractAlbumModel Class Reference

Inheritance diagram for Digikam::AbstractAlbumModel:



## Public Types

- enum [AlbumDataRole](#) {  
[AlbumTitleRole](#) = Qt::UserRole , [AlbumTypeRole](#) = Qt::UserRole + 1 , [AlbumPointerRole](#) = Qt::UserRole + 2  
, [AlbumIdRole](#) = Qt::UserRole + 3 ,  
[AlbumGlobalIdRole](#) = Qt::UserRole + 4 , [AlbumSortRole](#) = Qt::UserRole + 5 }
- enum [RootAlbumBehavior](#) { [IncludeRootAlbum](#) , [IgnoreRootAlbum](#) }

## Signals

- void [rootAlbumAvailable](#) ()

## Public Member Functions

- [AbstractAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [Album](#) \* [albumForIndex](#) (const [QModelIndex](#) &index) const
- [Album::Type](#) [albumType](#) () const
- int **columnCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [AlbumModelDragDropHandler](#) \* [dragDropHandler](#) () const
- bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role=[Qt::DisplayRole](#)) const override
- [QModelIndex](#) **index** (int row, int column, const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QModelIndex](#) [indexForAlbum](#) ([Album](#) \*album) const
- bool [isFaceTagModel](#) () const
- [QMimeData](#) \* **mimeData** (const [QModelIndexList](#) &indexes) const override
- [QStringList](#) **mimeTypes** () const override
- [QModelIndex](#) **parent** (const [QModelIndex](#) &index) const override
- [Album](#) \* **rootAlbum** () const
- [RootAlbumBehavior](#) [rootAlbumBehavior](#) () const
- [QModelIndex](#) [rootAlbumIndex](#) () const
- int **rowCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- void [setDragDropHandler](#) ([AlbumModelDragDropHandler](#) \*handler)
- void [setDropIndex](#) (const [QModelIndex](#) &index)
- [Qt::DropActions](#) **supportedDropActions** () const override

## Static Public Member Functions

- static [Album](#) \* [retrieveAlbum](#) (const [QModelIndex](#) &index)

## Protected Slots

- void **slotAlbumAboutToBeAdded** ([Album](#) \*album, [Album](#) \*parent, [Album](#) \*prev)
- void **slotAlbumAboutToBeDeleted** ([Album](#) \*album)
- void **slotAlbumAdded** ([Album](#) \*)
- void **slotAlbumHasBeenDeleted** ([Album](#) \*album)
- void **slotAlbumIconChanged** ([Album](#) \*album)
- void **slotAlbumRenamed** ([Album](#) \*album)
- void **slotAlbumsCleared** ()

## Protected Member Functions

- virtual void `albumCleared` (`Album *`)  
*Notification when an entry is removed.*
- virtual QVariant `albumData` (`Album *a`, int role) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual void `allAlbumsCleared` ()  
*Notification when all entries are removed.*
- virtual QString `columnHeader` () const  
*For subclassing convenience: A part of the implementation of headerData()*
- virtual QVariant `decorationRoleData` (`Album *a`) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual bool `filterAlbum` (`Album *album`) const
- virtual QVariant `fontRoleData` (`Album *a`) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual Qt::ItemFlags `itemFlags` (`Album *album`) const  
*For subclassing convenience: A part of the implementation of itemFlags()*
- void `setEnabledDrag` (bool enable)
- void `setEnabledDrop` (bool enable)
- void `setFaceTagModel` (bool enable)
- virtual QVariant `sortRoleData` (`Album *a`) const  
*For subclassing convenience: A part of the implementation of data()*

## 6.2.1 Member Enumeration Documentation

### 6.2.1.1 AlbumDataRole

```
enum Digikam::AbstractAlbumModel::AlbumDataRole
```

#### Enumerator

AlbumTitleRole	Returns the album title. Principally the same as display role, but without any additions.
AlbumTypeRole	Returns the <code>Album::Type</code> of the associated album.
AlbumPointerRole	Returns a pointer to the associated <code>Album</code> object.
AlbumIdRole	Returns the id of the associated <code>Album</code> object.
AlbumGlobalIdRole	Returns the global id (unique across all album types)
AlbumSortRole	Returns the data to sort on.

### 6.2.1.2 RootAlbumBehavior

```
enum Digikam::AbstractAlbumModel::RootAlbumBehavior
```

`AbstractAlbumModel` is the abstract base class for all models that present `Album` objects as managed by `AlbumManager`. You will want to create an instance of the base classes.

#### Enumerator

IncludeRootAlbum	The root album will be included as a single parent item with all top-level album as children
IgnoreRootAlbum	The root album will not be included, but all top-level album are represented as top-level items in this view

## 6.2.2 Constructor & Destructor Documentation

### 6.2.2.1 AbstractAlbumModel()

```
Digikam::AbstractAlbumModel::AbstractAlbumModel (
    Album::Type albumType,
    Album *const rootAlbum,
    RootAlbumBehavior rootBehavior = IncludeRootAlbum,
    QObject *const parent = nullptr ) [explicit]
```

Create an [AbstractAlbumModel](#) object for albums with the given type. Pass the root album if it is already available. Do not use this class directly, but one of the subclasses.

## 6.2.3 Member Function Documentation

### 6.2.3.1 albumCleared()

```
virtual void Digikam::AbstractAlbumModel::albumCleared (
    Album * ) [inline], [protected], [virtual]
```

Reimplemented in [Digikam::AbstractCountingAlbumModel](#), and [Digikam::AbstractCheckableAlbumModel](#).

### 6.2.3.2 albumData()

```
QVariant Digikam::AbstractAlbumModel::albumData (
    Album * a,
    int role ) const [protected], [virtual]
```

#### Note

these can be reimplemented in a subclass

Reimplemented in [Digikam::AbstractCountingAlbumModel](#), [Digikam::AbstractCheckableAlbumModel](#), [Digikam::AlbumModel](#), [Digikam::TagModel](#), and [Digikam::SearchModel](#).

### 6.2.3.3 albumForIndex()

```
Album * Digikam::AbstractAlbumModel::albumForIndex (
    const QModelIndex & index ) const
```

Returns the album object associated with the given model index

### 6.2.3.4 albumType()

```
Album::Type Digikam::AbstractAlbumModel::albumType ( ) const
```

Returns the [Album::Type](#) of the contained albums

### 6.2.3.5 allAlbumsCleared()

```
virtual void Digikam::AbstractAlbumModel::allAlbumsCleared ( ) [inline], [protected], [virtual]
```

Reimplemented in [Digikam::AbstractCountingAlbumModel](#), and [Digikam::AbstractCheckableAlbumModel](#).

### 6.2.3.6 columnHeader()

```
QString Digikam::AbstractAlbumModel::columnHeader ( ) const [protected], [virtual]
```

Reimplemented in [Digikam::AbstractSpecificAlbumModel](#).

### 6.2.3.7 decorationRoleData()

```
QVariant Digikam::AbstractAlbumModel::decorationRoleData (
    Album * a ) const [protected], [virtual]
```

Reimplemented in [Digikam::AlbumModel](#), [Digikam::TagModel](#), and [Digikam::DateAlbumModel](#).

### 6.2.3.8 dragDropHandler()

```
AlbumModelDragDropHandler * Digikam::AbstractAlbumModel::dragDropHandler ( ) const
```

Returns the drag drop handler, or 0 if none is installed

### 6.2.3.9 filterAlbum()

```
bool Digikam::AbstractAlbumModel::filterAlbum (
    Album * album ) const [protected], [virtual]
```

Returns true for those and only those albums that shall be contained in this model. They must have a common root album, which is set in the constructor.

### 6.2.3.10 fontRoleData()

```
QVariant Digikam::AbstractAlbumModel::fontRoleData (
    Album * a ) const [protected], [virtual]
```

Reimplemented in [Digikam::TagModel](#).

### 6.2.3.11 indexForAlbum()

```
QModelIndex Digikam::AbstractAlbumModel::indexForAlbum (
    Album * album ) const
```

Return the QModelIndex for the given album, or an invalid index if the album is not contained in this model.



### 6.2.3.12 isFaceTagModel()

```
bool Digikam::AbstractAlbumModel::isFaceTagModel ( ) const
```

Returns true if the album model a face tag model

### 6.2.3.13 retrieveAlbum()

```
Album * Digikam::AbstractAlbumModel::retrieveAlbum (
    const QModelIndex & index ) [static]
```

Returns the album represented by the index. In contrast to [albumForIndex\(\)](#), the index can be from any proxy model, as long as an [AbstractAlbumModel](#) is at the end.

### 6.2.3.14 rootAlbumAvailable

```
void Digikam::AbstractAlbumModel::rootAlbumAvailable ( ) [signal]
```

This is initialized once after creation, if the root album becomes available, if it was not already available at time of construction. This is emitted regardless of root album policy.

### 6.2.3.15 rootAlbumBehavior()

```
AbstractAlbumModel::RootAlbumBehavior Digikam::AbstractAlbumModel::rootAlbumBehavior ( ) const
```

Returns the root album behavior set for this model

### 6.2.3.16 rootAlbumIndex()

```
QModelIndex Digikam::AbstractAlbumModel::rootAlbumIndex ( ) const
```

Return the index corresponding to the root album. If the policy is IgnoreRootAlbum, this is an invalid index.

### 6.2.3.17 setDragDropHandler()

```
void Digikam::AbstractAlbumModel::setDragDropHandler (
    AlbumModelDragDropHandler * handler )
```

Set a drag drop handler

### 6.2.3.18 setDropIndex()

```
void Digikam::AbstractAlbumModel::setDropIndex (
    const QModelIndex & index )
```

Set current index from QDragMoveEvent

### 6.2.3.19 `setEnabledDrag()`

```
void Digikam::AbstractAlbumModel::setEnabledDrag (
    bool enable ) [protected]
```

Switch on drag and drop globally for all items. Default is true. For per-item cases reimplement [itemFlags\(\)](#).

### 6.2.3.20 `sortRoleData()`

```
QVariant Digikam::AbstractAlbumModel::sortRoleData (
    Album * a ) const [protected], [virtual]
```

Reimplemented in [Digikam::DateAlbumModel](#).



## Public Types

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()  
*slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()  
*slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()  
*slotExpandNode - expands recursively selected nodes*

## Signals

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< [Album](#) \* > &selectedAlbums)

## Public Member Functions

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void [addContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractSpecificAlbumModel](#) \* [albumModel](#) () const
- QList< [ContextMenuElement](#) \* > [contextMenuElements](#) () const
- template<class A >  
QList< A \* > [currentAlbums](#) ()
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [expandMatches](#) (const QModelIndex &index)
- QModelIndex [indexVisuallyAt](#) (const QPoint &p)
- void [removeContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > [selectedItems](#) ()  
*selectedItems()* -
- void [setAlbumManagerCurrentAlbum](#) (const bool setCurrentAlbum)
- void [setContextMenuIcon](#) (const QPixmap &pixmap)
- void [setContextMenuTitle](#) (const QString &title)
- void [setEnableContextMenu](#) (const bool enable)
- void [setExpandNewCurrentItem](#) (const bool doThat)
- void [setExpandOnSingleClick](#) (const bool doThat)
- void [setSelectAlbumOnClick](#) (const bool selectOnClick)
- void [setSelectOnContextMenu](#) (const bool select)
- bool [viewportEvent](#) (QEvent \*event) override

## Public Member Functions inherited from Digikam::StateSavingObject

- [StateSavingDepth](#) `getStateSavingDepth ()` const
- void `loadState ()`
- void `saveState ()`
- virtual void `setConfigGroup (const KConfigGroup &group)`
- virtual void `setEntryPrefix (const QString &prefix)`
- void `setStateSavingDepth (const StateSavingDepth depth)`
- [StateSavingObject](#) (QObject \*const host)
- virtual `~StateSavingObject ()`

## Protected Slots

- void `albumSettingsChanged ()`
- void `slotCurrentChanged ()`
- virtual void `slotRootAlbumAvailable ()`
- void `slotSearchTextSettingsAboutToChange (bool searched, bool willSearch)`
- void `slotSearchTextSettingsChanged (bool wasSearching, bool searching)`
- void `slotSelectionChanged ()`

## Protected Member Functions

- virtual void `addCustomContextMenuActions (ContextMenuHelper &cmh, Album *album)`
- virtual QPixmap `contextMenuIcon ()` const
- virtual QString `contextMenuTitle ()` const
- void `dragEnterEvent (QDragEnterEvent *e)` override
- void `dragLeaveEvent (QDragLeaveEvent *e)` override
- void `dragMoveEvent (QDragMoveEvent *e)` override
- void `dropEvent (QDropEvent *e)` override
- virtual void `handleCustomContextMenuAction (QAction *action, const AlbumPointer< Album > &album)`
- virtual void `middleButtonPressed (Album *a)`
- void `mousePressEvent (QMouseEvent *e)` override

*Other helper methods.*

- virtual QPixmap  `pixmapForDrag (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)`
- void `rowsAboutToBeRemoved (const QModelIndex &parent, int start, int end)` override
- void `rowsInserted (const QModelIndex &index, int start, int end)` override
- void `setAlbumFilterModel (AlbumFilterModel *const filterModel)`
- void `setAlbumModel (AbstractSpecificAlbumModel *const model)`
- virtual bool `showContextMenuAt (QContextMenuEvent *event, Album *albumForEvent)`
- void `startDrag (Qt::DropActions supportedActions)` override

## Protected Member Functions inherited from Digikam::StateSavingObject

- QString `entryName (const QString &base)` const
- KConfigGroup `getConfigGroup ()` const

## Protected Attributes

- [AlbumFilterModel](#) \* `m_albumFilterModel` = nullptr
- [AbstractSpecificAlbumModel](#) \* `m_albumModel` = nullptr
- bool `m_checkOnMiddleClick` = false
- [AlbumModelDragDropHandler](#) \* `m_dragDropHandler` = nullptr
- Flags `m_flags` = DefaultFlags
- int `m_lastScrollBarValue` = 0
- bool `m_restoreCheckState` = false

### 6.3.1 Detailed Description

Base class for all tree views that display Album-based content provided by an [AbstractSpecificAlbumModel](#). This class enables various utility functions like selecting albums on mouse actions or providing an infrastructure for displaying a context menu for albums.

Context menu handling is implemented as template methods with hook methods that can be implemented by subclasses to provide a custom behavior. In default mode no context menu is shown at all. It must be enabled via a call to `setEnabledContextMenu`.

### 6.3.2 Member Enumeration Documentation

#### 6.3.2.1 Flag

```
enum Digikam::AbstractAlbumTreeView::Flag
```

Enumerator

CreateDefaultModel	Create a default model. Not supported by abstract classes. Not part of default flags!
CreateDefaultFilterModel	Create a default filter model.
CreateDefaultDelegate	Create a delegate which paints according to settings. If not set, the Qt default delegate of the view is used.
ShowCountAccordingToSettings	Show the count according to the settings. If not set, call <code>setShowCount()</code> on the model yourself.
AlwaysShowInclusiveCounts	Always show the inclusive counts. Not part of default flags!

### 6.3.3 Constructor & Destructor Documentation

#### 6.3.3.1 AbstractAlbumTreeView()

```
Digikam::AbstractAlbumTreeView::AbstractAlbumTreeView (
    QWidget *const parent,
    Flags flags )
```

Constructs an album tree view. If you give 0 for model, call `setAlbumModel` afterwards. If you supply 0 for filterModel, call `setAlbumFilterModel` afterwards.

### 6.3.4 Member Function Documentation

#### 6.3.4.1 adaptColumnsToContent

```
void Digikam::AbstractAlbumTreeView::adaptColumnsToContent ( ) [slot]
```

Adapt the column sizes to the contents of the tree view.

#### 6.3.4.2 addCustomContextMenuActions()

```
void Digikam::AbstractAlbumTreeView::addCustomContextMenuActions (
    ContextMenuHelper & cmh,
    Album * album ) [protected], [virtual]
```

Hook method to add custom actions to the generated context menu.

## Parameters

<i>cmh</i>	helper object to create the context menu
<i>album</i>	tag on which the context menu will be created. May be null if it is requested on no tag entry

Reimplemented in [Digikam::TagFilterView](#), [Digikam::AlbumSelectTreeView](#), [Digikam::TagCheckView](#), [Digikam::TagFolderView](#), [Digikam::EditableSearchTreeView](#), and [Digikam::NormalSearchTreeView](#).

**6.3.4.3 contextMenuIcon()**

```
QPixmap Digikam::AbstractAlbumTreeView::contextMenuIcon ( ) const [protected], [virtual]
```

Hook method that can be implemented to return a special icon used for the context menu.

## Returns

the icon for the context menu

**6.3.4.4 contextMenuTitle()**

```
QString Digikam::AbstractAlbumTreeView::contextMenuTitle ( ) const [protected], [virtual]
```

Hook method to implement that returns the title for the context menu.

## Returns

title for the context menu

Reimplemented in [Digikam::TagFolderView](#), and [Digikam::EditableSearchTreeView](#).

**6.3.4.5 currentAlbumChanged**

```
void Digikam::AbstractAlbumTreeView::currentAlbumChanged (
    Album * currentAlbum ) [signal]
```

Emitted when the currently selected album changes

**6.3.4.6 doLoadState()**

```
void Digikam::AbstractAlbumTreeView::doLoadState ( ) [override], [virtual]
```

Implements state loading for the album tree view in a somewhat clumsy procedure because the model may not be fully loaded when this method is called. Therefore the config is first parsed into `d->statesByAlbumId` which holds the state of all tree view entries indexed by album id. Afterwards an initial sync run is done restoring the state of all model entries that are already present at this time. Every processed entry is removed from `d->statesByAlbumId`. If there are still entries left in this map we assume that the model is not fully loaded at the moment. Therefore the `rowsInserted` signal is connected to a slot that restores the state of new rows based on the remaining entries in `d->statesByAlbumId`.

Implements [Digikam::StateSavingObject](#).

Reimplemented in [Digikam::AbstractCheckableAlbumTreeView](#), and [Digikam::TagCheckView](#).



### 6.3.4.7 doSaveState()

```
void Digikam::AbstractAlbumTreeView::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

Reimplemented in [Digikam::AbstractCheckableAlbumTreeView](#), and [Digikam::TagCheckView](#).

### 6.3.4.8 expandEverything

```
void Digikam::AbstractAlbumTreeView::expandEverything (
    const QModelIndex & index ) [slot]
```

Expands the complete tree under the given index.

#### Parameters

<i>index</i>	the index to start expanding everything
--------------	---

### 6.3.4.9 expandMatches()

```
bool Digikam::AbstractAlbumTreeView::expandMatches (
    const QModelIndex & index )
```

Ensures that every current match is visible by expanding all parent entries.

#### Parameters

<i>index</i>	the index to start ensuring expansion state
--------------	---

#### Returns

`true` if there was a match under `index`. This return value can normally be ignored by the caller because it is only used for an internal recursion.

### 6.3.4.10 handleCustomContextMenuAction()

```
void Digikam::AbstractAlbumTreeView::handleCustomContextMenuAction (
    QAction * action,
    const AlbumPointer< Album > & album ) [protected], [virtual]
```

Hook method to handle the custom context menu actions that were added with `addCustomContextMenuActions`.

#### Parameters

<i>action</i>	the action that was chosen by the user, may be null if none of the custom actions were selected
<i>album</i>	the tag on which the context menu was requested. May be null if there was no

Reimplemented in [Digikam::TagFilterView](#), [Digikam::AlbumSelectTreeView](#), [Digikam::TagFolderView](#), [Digikam::EditableSearchTreeView](#) and [Digikam::NormalSearchTreeView](#).

#### 6.3.4.11 indexVisuallyAt()

```
QModelIndex Digikam::AbstractAlbumTreeView::indexVisuallyAt (
    const QPoint & p )
```

This is a combination of `indexAt()` checked with `visualRect()`. `p` must be in the viewport currently. Decoration will not be included. Suitable for mouse click positions.

#### 6.3.4.12 pixmapForDrag()

```
QPixmap Digikam::AbstractAlbumTreeView::pixmapForDrag (
    const QStyleOptionViewItem & option,
    QList< QModelIndex > indexes ) [protected], [virtual]
```

TODO: Move to delegate, when we have one. Copy code from image delegate for creating icons when dragging multiple items

#### 6.3.4.13 scrollToSelectedAlbum

```
void Digikam::AbstractAlbumTreeView::scrollToSelectedAlbum ( ) [slot]
```

Scrolls to the first selected album if there is one.

#### 6.3.4.14 selectedAlbumsChanged

```
void Digikam::AbstractAlbumTreeView::selectedAlbumsChanged (
    const QList< Album * > & selectedAlbums ) [signal]
```

Emitted when the current selection changes. Use `currentChanged` unless in multi-selection mode.

#### 6.3.4.15 setAlbumManagerCurrentAlbum()

```
void Digikam::AbstractAlbumTreeView::setAlbumManagerCurrentAlbum (
    const bool setCurrentAlbum )
```

Some treeviews shall control the global current album kept by [AlbumManager](#). Other treeview are self-contained and shall not change the current album. Default: false

#### 6.3.4.16 setContextMenuIcon()

```
void Digikam::AbstractAlbumTreeView::setContextMenuIcon (
    const QPixmap & pixmap )
```

Set the context menu title and icon. This is used by the default implementation of [contextMenuIcon\(\)](#) and [contextMenuTitle\(\)](#). You can alternatively reimplement these methods.

#### 6.3.4.17 setCurrentAlbums

```
void Digikam::AbstractAlbumTreeView::setCurrentAlbums (
    const QList< Album * > & albums,
    bool selectInAlbumManager = true ) [slot]
```

Selects the given album.

## Parameters

<i>albums</i>	the albums to select
<i>selectInAlbumManager</i>	the album will be set as current album, if both this parameter is true and <a href="#">setAlbumManagerCurrentAlbum()</a> was set to true.

**6.3.4.18 setEnableContextMenu()**

```
void Digikam::AbstractAlbumTreeView::setEnableContextMenu (
    const bool enable )
```

Determines the global decision to show a popup menu or not. More detailed decision at which position a menu can be shown and where not can be made by implementing `showContextMenuAt`.

## Parameters

<i>enable</i>	if true, a context menu can be shown
---------------	--------------------------------------

**6.3.4.19 setExpandNewCurrentItem()**

```
void Digikam::AbstractAlbumTreeView::setExpandNewCurrentItem (
    const bool doThat )
```

Expand an item when making it the new current item

**6.3.4.20 setExpandOnSingleClick()**

```
void Digikam::AbstractAlbumTreeView::setExpandOnSingleClick (
    const bool doThat )
```

Enable expanding of tree items on single click on the item (default: off)

**6.3.4.21 setSelectedAlbumOnClick()**

```
void Digikam::AbstractAlbumTreeView::setSelectAlbumOnClick (
    const bool selectOnClick )
```

Sets whether to select an album on click via the album manager or not.

## Parameters

<i>selectOnClick</i>	if true, a click on an item automatically sets this item as the current album in the album manager
----------------------	--

### 6.3.4.22 setSelectOnContextMenu()

```
void Digikam::AbstractAlbumTreeView::setSelectOnContextMenu (
    const bool select )
```

Sets whether to select the album under the mouse cursor on a context menu request (so that the album is shown using the album manager) or not

Defaults to true.

#### Parameters

<i>select</i>	true if a context menu request shall select the album
---------------	---

### 6.3.4.23 showContextMenuAt()

```
bool Digikam::AbstractAlbumTreeView::showContextMenuAt (
    QContextMenuEvent * event,
    Album * albumForEvent ) [protected], [virtual]
```

Hook method to implement that determines if a context menu shall be displayed for the given event at the position coded in the event.

#### Parameters

<i>event</i>	context menu event to react on
<i>albumForEvent</i>	the album at the mouse position or null if there is no album at that position

#### Returns

true if a context menu shall be displayed at the event coordinates, else false

### 6.3.4.24 slotRootAlbumAvailable

```
void Digikam::AbstractAlbumTreeView::slotRootAlbumAvailable ( ) [protected], [virtual], [slot]
```

override if implemented behavior is not as intended

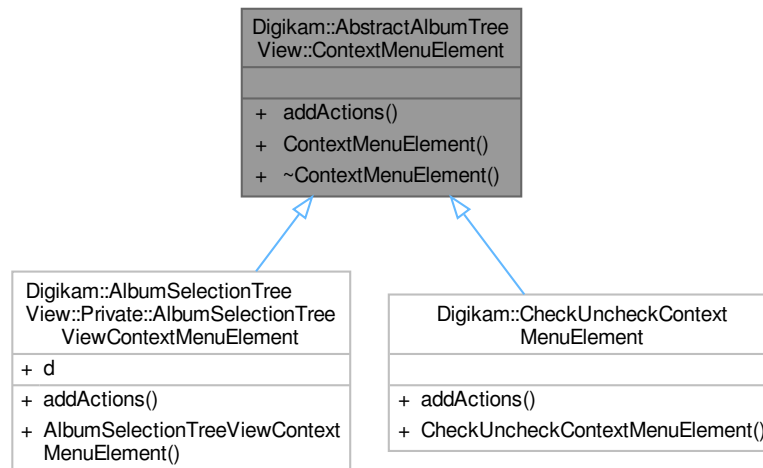
### 6.3.4.25 viewportEvent()

```
bool Digikam::AbstractAlbumTreeView::viewportEvent (
    QEvent * event ) [override]
```

For internal use only.

## 6.4 Digikam::AbstractAlbumTreeView::ContextMenuElement Class Reference

Inheritance diagram for Digikam::AbstractAlbumTreeView::ContextMenuElement:



### Public Member Functions

- virtual void `addActions` (`AbstractAlbumTreeView *view`, `ContextMenuHelper &cmh`, `Album *album`)=0

### 6.4.1 Detailed Description

Add a context menu element which can add actions to the context menu when the menu is generated. First, `addCustomContextMenuActions` is called, then all elements' `addActions` method is called in order of addition.

### 6.4.2 Member Function Documentation

#### 6.4.2.1 addActions()

```
virtual void Digikam::AbstractAlbumTreeView::ContextMenuElement::addActions (
    AbstractAlbumTreeView * view,
    ContextMenuHelper & cmh,
    Album * album ) [pure virtual]
```

Add actions to the context menu being generated

#### Parameters

<code>view</code>	The <code>AbstractAlbumTreeView</code> which generates the menu
<code>cmh</code>	helper object to create the context menu
<code>album</code>	the album on which the context menu will be created. May be null if it is requested on no tag entry

## 6.5 Digikam::AbstractAlbumTreeView::Private Class Reference

### Public Attributes

- const QString **configCurrentIndexEntry** = QLatin1String("CurrentIndex")
- const QString **configExpansionEntry** = QLatin1String("Expansion")
- const QString **configSelectionEntry** = QLatin1String("Selection")
- const QString **configSortColumnEntry** = QLatin1String("SortColumn")
- const QString **configSortOrderEntry** = QLatin1String("SortOrder")
- QList< [ContextMenuElement](#) \* > **contextMenuElements**
- QPixmap **contextMenuIcon**
- QString **contextMenuTitle**
- [AlbumTreeViewDelegate](#) \* **delegate** = nullptr
- bool **enableContextMenu** = false
- bool **expandNewCurrent** = false
- bool **expandOnSingleClick** = false
- [AlbumPointer](#)< [Album](#) > **lastSelectedAlbum**
- QTimer \* **resizeColumnsTimer** = nullptr
- QMap< int, [Digikam::State](#) > **searchBackup**
- bool **selectAlbumOnClick** = false
- bool **selectOnContextMenu** = true
- bool **setInAlbumManager** = false
- QMap< int, [Digikam::State](#) > **statesByAlbumId**

## 6.6 Digikam::AbstractAlbumTreeViewSelectComboBox Class Reference

Inheritance diagram for Digikam::AbstractAlbumTreeViewSelectComboBox:



### Public Member Functions

- [AbstractAlbumTreeViewSelectComboBox](#) (QWidget \*const parent=nullptr)
- void [addCheckUncheckContextMenuActions](#) ()
- void [setTreeView](#) (AbstractAlbumTreeView \*const treeView)

### Public Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- [AlbumSelectComboBox](#) (QWidget \*const parent=nullptr)
- QSortFilterProxyModel \* [filterModel](#) () const
- bool [isCheckedable](#) () const
- [AbstractCheckableAlbumModel](#) \* [model](#) () const
- void [setAlbumModels](#) ([AbstractCheckableAlbumModel](#) \*model, [AlbumFilterModel](#) \*filterModel=nullptr)
- void [setAllSelectedText](#) (bool all)
- void [setCheckable](#) (bool checkable)
- void [setCloseOnActivate](#) (bool close)
- void [setDefaultAlbumModel](#) ()
- void [setDefaultTagModel](#) ()
- void [setNoSelectionText](#) (const QString &text)
- void [setRecursive](#) (bool recursive)
- void [setShowCheckStateSummary](#) (bool show)

### Public Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- void [installView](#) (QAbstractItemView \*view=nullptr) override
- void [setLineEdit](#) (QLineEdit \*edit)
- void [setLineEditText](#) (const QString &text)
- [TreeViewLineEditComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::TreeViewComboBox](#)

- [TreeViewComboBox](#) (QWidget \*parent=nullptr)
- QTreeView \* [view](#) () const

### Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex [currentIndex](#) () const
- void [hidePopup](#) () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void [setCurrentIndex](#) (const QModelIndex &index)
- void [showPopup](#) () override

### Protected Member Functions

- void [installView](#) (QAbstractItemView \*view=nullptr) override
- void [sendViewportEventToView](#) (QEvent \*e) override

### Protected Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- void [installView](#) (QAbstractItemView \*view=nullptr) override



### Protected Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- virtual void [installLineEdit](#) ()

### Protected Member Functions inherited from [Digikam::TreeViewComboBox](#)

- void [sendViewportEventToView](#) (QEvent \*e) override

### Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool [eventFilter](#) (QObject \*watched, QEvent \*event) override
- void [installView](#) (QAbstractItemView \*view)

### Protected Attributes

- [AbstractAlbumTreeView](#) \* [m\\_treeView](#) = nullptr

### Protected Attributes inherited from [Digikam::TreeViewLineEditComboBox](#)

- QLineEdit \* [m\\_comboLineEdit](#) = nullptr

### Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

- QAbstractItemView \* [m\\_view](#) = nullptr

### Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex [m\\_currentIndex](#)

### Additional Inherited Members

### Public Slots inherited from [Digikam::AlbumSelectComboBox](#)

- void [hidePopup](#) () override
- virtual void [updateText](#) ()

## 6.6.1 Constructor & Destructor Documentation

### 6.6.1.1 AbstractAlbumTreeViewSelectComboBox()

```
Digikam::AbstractAlbumTreeViewSelectComboBox::AbstractAlbumTreeViewSelectComboBox (
    QWidget *const parent = nullptr) [explicit]
```

Abstract class. This is an [AlbumSelectComboBox](#) which installs an [AlbumTreeView](#), not a plain QTreeView, as view.

## 6.6.2 Member Function Documentation

### 6.6.2.1 addCheckUncheckContextMenuActions()

```
void Digikam::AbstractAlbumTreeViewSelectComboBox::addCheckUncheckContextMenuActions ( )
```

Enables a context menu which contains options to check or uncheck groups of albums, given you have a checkable model. Call this method after `setModel()`.

### 6.6.2.2 installView()

```
void Digikam::AbstractAlbumTreeViewSelectComboBox::installView (
    QAbstractItemView * view = nullptr ) [override], [protected], [virtual]
```

Replace the standard combo box list view with a `QTreeView`. Call this after installing an appropriate model.

Reimplemented from [Digikam::TreeViewComboBox](#).

### 6.6.2.3 sendViewportEventToView()

```
void Digikam::AbstractAlbumTreeViewSelectComboBox::sendViewportEventToView (
    QEvent * e ) [override], [protected], [virtual]
```

Implement in subclass: Send the given event to the `viewportEvent()` method of `m_view`. This method is protected for a usual `QAbstractItemView`. You can override, pass a view, and call parent implementation. The existing view will be used. You must then also reimplement `sendViewportEventToView`.

Implements [Digikam::StayPoppedUpComboBox](#).

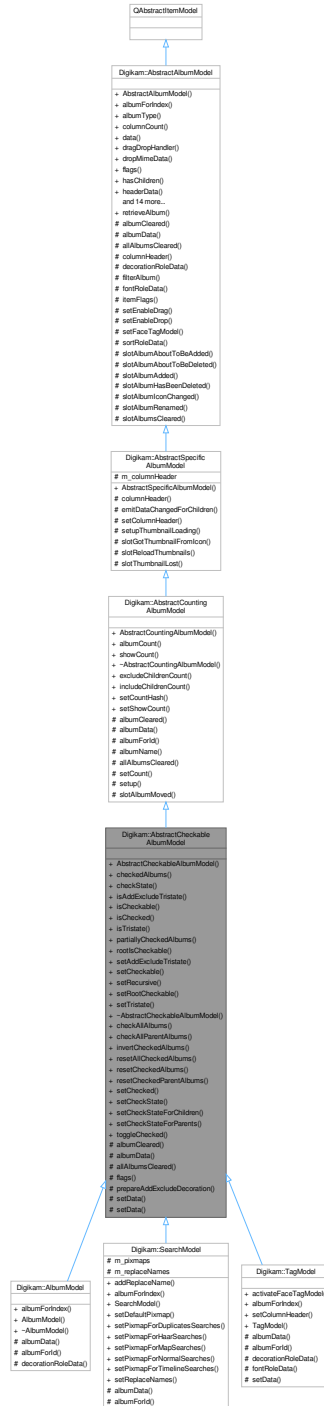
### 6.6.2.4 setTreeView()

```
void Digikam::AbstractAlbumTreeViewSelectComboBox::setTreeView (
    AbstractAlbumTreeView *const treeView )
```

Set a tree view created by you instead of creating a default view (in the subclasses). Only takes effect before calling `setModel`.

## 6.7 Digikam::AbstractCheckableAlbumModel Class Reference

Inheritance diagram for Digikam::AbstractCheckableAlbumModel:



### Public Slots

- void **checkAllAlbums** (const QModelIndex &parent=QModelIndex())  
*Checks all albums beneath the given parent.*

- void **checkAllParentAlbums** (const QModelIndex &child)  
*Checks all parent albums starting at the child, including it.*
- void **invertCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Inverts the checked state of all albums under the given parent.*
- void **resetAllCheckedAlbums** ()  
*Resets the checked state of all albums to Qt::Unchecked.*
- void **resetCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Resets the checked state of all albums under the given parent.*
- void **resetCheckedParentAlbums** (const QModelIndex &child)  
*Resets the checked state of all parents of the child including it.*
- void **setChecked** (Album \*album, bool isChecked)  
*Sets the check state of album to Checked or Unchecked.*
- void **setCheckState** (Album \*album, Qt::CheckState state)  
*Sets the check state of the album.*
- void **setCheckStateForChildren** (Album \*album, Qt::CheckState state)  
*Sets the checked state recursively for all children of but not for the given album.*
- void **setCheckStateForParents** (Album \*album, Qt::CheckState state)  
*Sets the checked state recursively for all parents of but not for the given album.*
- void **toggleChecked** (Album \*album)  
*Toggles the check state of album between Checked or Unchecked.*

### Public Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [excludeChildrenCount](#) (const QModelIndex &index)
- void [includeChildrenCount](#) (const QModelIndex &index)
- void [setCountHash](#) (const QHash< int, int > &idCountHash)
- void **setShowCount** (bool show)  
*Call to enable or disable showing the count. Default is false.*

### Signals

- void [checkStateChanged](#) (Album \*album, Qt::CheckState checkState)

### Signals inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [signalUpdateAlbumCount](#) (Album \*album)

### Signals inherited from [Digikam::AbstractAlbumModel](#)

- void [rootAlbumAvailable](#) ()

## Public Member Functions

- **AbstractCheckableAlbumModel** ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [QList](#)< [Album](#) \* > **checkedAlbums** () const  
*Returns a list of album with check state Checked.*
- [Qt::CheckState](#) **checkState** ([Album](#) \*album) const  
*Returns the check state of the album.*
- bool **isAddExcludeTristate** () const
- bool **isCheckable** () const
- bool **isChecked** ([Album](#) \*album) const  
*Returns if the given album has the check state Checked.*
- bool **isTristate** () const
- [QList](#)< [Album](#) \* > **partiallyCheckedAlbums** () const  
*Returns a list of album with partially check state Checked.*
- bool **rootsCheckable** () const
- void **setAddExcludeTristate** (bool b)
- void **setCheckable** (bool isCheckable)  
*Triggers if the albums in this model are checkable.*
- void **setRecursive** (bool recursive)
- void **setRootCheckable** (bool rootsCheckable)
- void **setTristate** (bool isTristate)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- **AbstractCountingAlbumModel** ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Supports displaying a count alongside the album name in DisplayRole.*
- virtual int **albumCount** ([Album](#) \*album) const
- bool **showCount** () const

## Public Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- **AbstractSpecificAlbumModel** ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Abstract base class, do not instantiate.*

## Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- **AbstractAlbumModel** ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [Album](#) \* **albumForIndex** (const [QModelIndex](#) &index) const
- [Album::Type](#) **albumType** () const
- int **columnCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [AlbumModelDragDropHandler](#) \* **dragDropHandler** () const
- bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role=[Qt::DisplayRole](#)) const override

- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex **indexForAlbum** (Album \*album) const
- bool **isFaceTagModel** () const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const override
- QStringList **mimeTypes** () const override
- QModelIndex **parent** (const QModelIndex &index) const override
- Album \* **rootAlbum** () const
- RootAlbumBehavior **rootAlbumBehavior** () const
- QModelIndex **rootAlbumIndex** () const
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setDragDropHandler** (AlbumModelDragDropHandler \*handler)
- void **setDropIndex** (const QModelIndex &index)
- Qt::DropActions **supportedDropActions** () const override

### Protected Member Functions

- void **albumCleared** (Album \*album) override  
*Notification when an entry is removed.*
- QVariant **albumData** (Album \*a, int role) const override  
*For subclassing convenience: A part of the implementation of data()*
- void **allAlbumsCleared** () override  
*Notification when all entries are removed.*
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- void **prepareAddExcludeDecoration** (Album \*a, QPixmap &icon) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role, bool recursive)
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- void **albumCleared** (Album \*album) override  
*Notification when an entry is removed.*
- QVariant **albumData** (Album \*a, int role) const override  
*Reimplemented from parent classes.*
- virtual Album \* **albumForId** (int id) const =0  
*need to implement in subclass*
- virtual QString **albumName** (Album \*a) const  
*Can reimplement in subclass.*
- void **allAlbumsCleared** () override  
*Notification when all entries are removed.*
- void **setCount** (Album \*album, int count)  
*If you do not use setCountHash, excludeChildrenCount and includeChildrenCount, you can set a count here.*
- void **setup** ()

### Protected Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString **columnHeader** () const override  
*For subclassing convenience: A part of the implementation of headerData()*
- void **emitDataChangedForChildren** (Album \*album)
- void **setColumnHeader** (const QString &header)
- void **setupThumbnailLoading** ()  
*You need to call this from your constructor if you intend to load the thumbnail facilities of this class.*

## Protected Member Functions inherited from Digikam::AbstractAlbumModel

- virtual QVariant [decorationRoleData](#) (Album \*a) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual bool [filterAlbum](#) (Album \*album) const
- virtual QVariant [fontRoleData](#) (Album \*a) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual Qt::ItemFlags [itemFlags](#) (Album \*album) const  
*For subclassing convenience: A part of the implementation of itemFlags()*
- void [setEnableDrag](#) (bool enable)
- void [setEnableDrop](#) (bool enable)
- void [setFaceTagModel](#) (bool enable)
- virtual QVariant [sortRoleData](#) (Album \*a) const  
*For subclassing convenience: A part of the implementation of data()*

## Additional Inherited Members

## Public Types inherited from Digikam::AbstractAlbumModel

- enum [AlbumDataRole](#) {  
[AlbumTitleRole](#) = Qt::UserRole , [AlbumTypeRole](#) = Qt::UserRole + 1 , [AlbumPointerRole](#) = Qt::UserRole + 2  
, [AlbumIdRole](#) = Qt::UserRole + 3 ,  
[AlbumGlobalIdRole](#) = Qt::UserRole + 4 , [AlbumSortRole](#) = Qt::UserRole + 5 }
- enum [RootAlbumBehavior](#) { [IncludeRootAlbum](#) , [IgnoreRootAlbum](#) }

## Static Public Member Functions inherited from Digikam::AbstractAlbumModel

- static Album \* [retrieveAlbum](#) (const QModelIndex &index)

## Protected Slots inherited from Digikam::AbstractCountingAlbumModel

- void [slotAlbumMoved](#) (Album \*album)

## Protected Slots inherited from Digikam::AbstractSpecificAlbumModel

- void [slotGotThumbnailFromIcon](#) (Album \*album, const QPixmap &thumbnail)
- void [slotReloadThumbnails](#) ()
- void [slotThumbnailLost](#) (Album \*album)

## Protected Slots inherited from Digikam::AbstractAlbumModel

- void [slotAlbumAboutToBeAdded](#) (Album \*album, Album \*parent, Album \*prev)
- void [slotAlbumAboutToBeDeleted](#) (Album \*album)
- void [slotAlbumAdded](#) (Album \*)
- void [slotAlbumHasBeenDeleted](#) (Album \*album)
- void [slotAlbumIconChanged](#) (Album \*album)
- void [slotAlbumRenamed](#) (Album \*album)
- void [slotAlbumsCleared](#) ()

## Protected Attributes inherited from [Digikam::AbstractSpecificAlbumModel](#)

- `QString m_columnHeader`

### 6.7.1 Constructor & Destructor Documentation

#### 6.7.1.1 AbstractCheckableAlbumModel()

```
Digikam::AbstractCheckableAlbumModel::AbstractCheckableAlbumModel (
    Album::Type albumType,
    Album *const rootAlbum,
    RootAlbumBehavior rootBehavior = IncludeRootAlbum,
    QObject *const parent = nullptr ) [explicit]
```

Abstract base class that manages the check state of Albums. Call `setCheckable(true)` to enable checkable albums.

### 6.7.2 Member Function Documentation

#### 6.7.2.1 albumCleared()

```
void Digikam::AbstractCheckableAlbumModel::albumCleared (
    Album * ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).

#### 6.7.2.2 albumData()

```
QVariant Digikam::AbstractCheckableAlbumModel::albumData (
    Album * a,
    int role ) const [override], [protected], [virtual]
```

#### Note

these can be reimplemented in a subclass

Reimplemented from [Digikam::AbstractAlbumModel](#).

Reimplemented in [Digikam::AlbumModel](#), [Digikam::TagModel](#), and [Digikam::SearchModel](#).

#### 6.7.2.3 allAlbumsCleared()

```
void Digikam::AbstractCheckableAlbumModel::allAlbumsCleared ( ) [override], [protected],
[virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).



#### 6.7.2.4 checkStateChanged

```
void Digikam::AbstractCheckableAlbumModel::checkStateChanged (
    Album * album,
    Qt::CheckState checkState ) [signal]
```

Emitted when the check state of an album changes. checkState contains the new Qt::CheckState of album

#### 6.7.2.5 prepareAddExcludeDecoration()

```
void Digikam::AbstractCheckableAlbumModel::prepareAddExcludeDecoration (
    Album * a,
    QPixmap & icon ) const [protected]
```

If in AddExcludeTristate mode, changes the icon as to indicate the state.

#### 6.7.2.6 setAddExcludeTristate()

```
void Digikam::AbstractCheckableAlbumModel::setAddExcludeTristate (
    bool b )
```

Sets a special tristate mode, which offers the three modes "unchecked", "added" and "excluded", where "excluded" corresponds to partially checked internally, but is reflected in the treeview through the decoration only.

#### 6.7.2.7 setData()

```
bool Digikam::AbstractCheckableAlbumModel::setData (
    const QModelIndex & index,
    const QVariant & value,
    int role = Qt::EditRole ) [override], [protected]
```

#### Note

Do not call this function directly, use the setData(..., bool recursive)

#### 6.7.2.8 setRecursive()

```
void Digikam::AbstractCheckableAlbumModel::setRecursive (
    bool recursive )
```

If an item gets checked, all childs get checked as well, If an item gets unchecked, all childs get unchecked as well

#### 6.7.2.9 setRootCheckable()

```
void Digikam::AbstractCheckableAlbumModel::setRootCheckable (
    bool rootIsCheckable )
```

Triggers if the root album is checkable. Only applicable if the root album is contained at all, and if isCheckable() is true.

### 6.7.2.10 setTristate()

```
void Digikam::AbstractCheckableAlbumModel::setTristate (
    bool isTristate )
```

Triggers if the albums in this model are tristate. Used to allow the user to actively set a third state, don't use if you only want to display a third state. Note that you want to set setCheckable(true) before.



- [AbstractCheckableAlbumModel](#) \* [albumModel](#) () const
- [CheckableAlbumFilterModel](#) \* [checkableAlbumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [checkableModel](#) () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

### Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

### Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void [addContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractSpecificAlbumModel](#) \* [albumModel](#) () const
- QList< [ContextMenuElement](#) \* > [contextMenuElements](#) () const
- template<class A >  
QList< A \* > [currentAlbums](#) ()
- bool [expandMatches](#) (const QModelIndex &index)
- QModelIndex [indexVisuallyAt](#) (const QPoint &p)
- void [removeContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > [selectedItems](#) ()  
*selectedItems()* -
- void [setAlbumManagerCurrentAlbum](#) (const bool setCurrentAlbum)
- void [setContextMenuIcon](#) (const QPixmap &pixmap)
- void [setContextMenuTitle](#) (const QString &title)
- void [setEnabledContextMenu](#) (const bool enable)
- void [setExpandNewCurrentItem](#) (const bool doThat)
- void [setExpandOnSingleClick](#) (const bool doThat)
- void [setSelectAlbumOnClick](#) (const bool selectOnClick)
- void [setSelectOnContextMenu](#) (const bool select)
- bool [viewportEvent](#) (QEvent \*event) override

### Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

**Protected Member Functions**

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void **rowsInserted** (const [QModelIndex](#) &parent, int start, int end) override
- void **setAlbumFilterModel** ([CheckableAlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AbstractCheckableAlbumModel](#) \*const model)

**Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)**

- void **rowsInserted** (const [QModelIndex](#) &parent, int start, int end) override
- void **setAlbumFilterModel** ([AlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AbstractCountingAlbumModel](#) \*const model)

**Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)**

- virtual void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album)
- virtual [QPixmap](#) [contextMenuIcon](#) () const
- virtual [QString](#) [contextMenuTitle](#) () const
- void **dragEnterEvent** ([QDragEnterEvent](#) \*e) override
- void **dragLeaveEvent** ([QDragLeaveEvent](#) \*e) override
- void **dragMoveEvent** ([QDragMoveEvent](#) \*e) override
- void **dropEvent** ([QDropEvent](#) \*e) override
- virtual void [handleCustomContextMenuAction](#) ([QAction](#) \*action, const [AlbumPointer](#)< [Album](#) > &album)
- void **mousePressEvent** ([QMouseEvent](#) \*e) override

*Other helper methods.*

- virtual [QPixmap](#) [pixmapForDrag](#) (const [QStyleOptionViewItem](#) &option, [QList](#)< [QModelIndex](#) > indexes)
- void **rowsAboutToBeRemoved** (const [QModelIndex](#) &parent, int start, int end) override
- void **rowsInserted** (const [QModelIndex](#) &index, int start, int end) override
- void **setAlbumFilterModel** ([AlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AbstractSpecificAlbumModel](#) \*const model)
- virtual bool [showContextMenuAt](#) ([QContextMenuEvent](#) \*event, [Album](#) \*albumForEvent)
- void **startDrag** ([Qt::DropActions](#) supportedActions) override

**Protected Member Functions inherited from [Digikam::StateSavingObject](#)**

- [QString](#) [entryName](#) (const [QString](#) &base) const
- [KConfigGroup](#) [getConfigGroup](#) () const

**Additional Inherited Members****Public Types inherited from [Digikam::AbstractAlbumTreeView](#)**

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

**Public Types inherited from [Digikam::StateSavingObject](#)**

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()
  - slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()
  - slotExpandNode - expands recursively selected nodes*

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) (Album \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< Album \* > &selectedAlbums)

## Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* [m\\_albumFilterModel](#) = nullptr
- [AbstractSpecificAlbumModel](#) \* [m\\_albumModel](#) = nullptr
- bool [m\\_checkOnMiddleClick](#) = false
- [AlbumModelDragDropHandler](#) \* [m\\_dragDropHandler](#) = nullptr
- Flags [m\\_flags](#) = DefaultFlags
- int [m\\_lastScrollBarValue](#) = 0
- bool [m\\_restoreCheckState](#) = false

## 6.8.1 Constructor & Destructor Documentation

### 6.8.1.1 AbstractCheckableAlbumTreeView()

```
Digikam::AbstractCheckableAlbumTreeView::AbstractCheckableAlbumTreeView (
    QWidget *const parent,
    Flags flags ) [explicit]
```

Models of these view *can* be checkable, they need *not*. You need to enable it on the model.

## 6.8.2 Member Function Documentation

### 6.8.2.1 albumModel()

```
AbstractCheckableAlbumModel * Digikam::AbstractCheckableAlbumTreeView::albumModel ( ) const
```

Manage check state through the model directly.

### 6.8.2.2 doLoadState()

```
void Digikam::AbstractCheckableAlbumTreeView::doLoadState ( ) [override], [virtual]
```

Implements state loading for the album tree view in a somewhat clumsy procedure because the model may not be fully loaded when this method is called. Therefore the config is first parsed into `d->statesByAlbumId` which holds the state of all tree view entries indexed by album id. Afterwards an initial sync run is done restoring the state of all model entries that are already present at this time. Every processed entry is removed from `d->statesByAlbumId`. If there are still entries left in this map we assume that the model is not fully loaded at the moment. Therefore the `rowsInserted` signal is connected to a slot that restores the state of new rows based on the remaining entries in `d->statesByAlbumId`.

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

Reimplemented in [Digikam::TagCheckView](#).

### 6.8.2.3 doSaveState()

```
void Digikam::AbstractCheckableAlbumTreeView::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

Reimplemented in [Digikam::TagCheckView](#).

### 6.8.2.4 isRestoreCheckState()

```
bool Digikam::AbstractCheckableAlbumTreeView::isRestoreCheckState ( ) const
```

Tells if the check state is restored while loading / saving state.

#### Returns

true if restoring check state is active

### 6.8.2.5 middleButtonPressed()

```
void Digikam::AbstractCheckableAlbumTreeView::middleButtonPressed (
    Album * a ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

### 6.8.2.6 setCheckOnMiddleClick()

```
void Digikam::AbstractCheckableAlbumTreeView::setCheckOnMiddleClick (
    bool doThat )
```

Enable checking on middle mouse button click (default: on).

### 6.8.2.7 setRestoreCheckState()

```
void Digikam::AbstractCheckableAlbumTreeView::setRestoreCheckState (
    bool restore )
```

Set whether to restore check state or not.

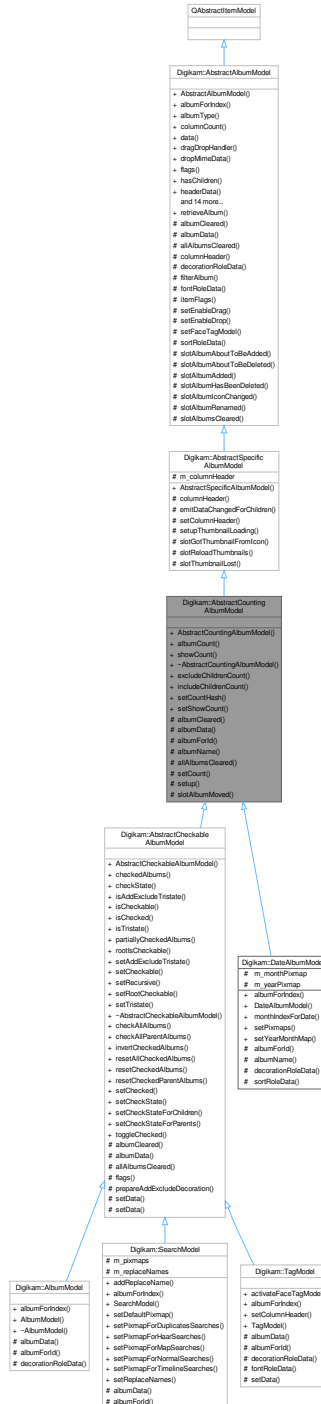
#### Parameters

<i>restore</i>	if true, restore check state
----------------	------------------------------



## 6.9 Digikam::AbstractCountingAlbumModel Class Reference

Inheritance diagram for Digikam::AbstractCountingAlbumModel:



### Public Slots

- void [excludeChildrenCount](#) (const QModelIndex &index)
- void [includeChildrenCount](#) (const QModelIndex &index)

- void [setCountHash](#) (const QHash< int, int > &idCountHash)
- void **setShowCount** (bool show)

*Call to enable or disable showing the count. Default is false.*

## Signals

- void **signalUpdateAlbumCount** ([Album](#) \*album)

## Signals inherited from [Digikam::AbstractAlbumModel](#)

- void [rootAlbumAvailable](#) ()

## Public Member Functions

- **AbstractCountingAlbumModel** ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Supports displaying a count alongside the album name in DisplayRole.*
- virtual int [albumCount](#) ([Album](#) \*album) const
- bool **showCount** () const

## Public Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- **AbstractSpecificAlbumModel** ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Abstract base class, do not instantiate.*

## Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- [AbstractAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [Album](#) \* [albumForIndex](#) (const [QModelIndex](#) &index) const
- [Album::Type](#) [albumType](#) () const
- int **columnCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [AlbumModelDragDropHandler](#) \* [dragDropHandler](#) () const
- bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role=[Qt::DisplayRole](#)) const override
- [QModelIndex](#) **index** (int row, int column, const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QModelIndex](#) [indexForAlbum](#) ([Album](#) \*album) const
- bool [isFaceTagModel](#) () const
- [QMimeData](#) \* **mimeData** (const [QModelIndexList](#) &indexes) const override
- [QStringList](#) **mimeTypes** () const override
- [QModelIndex](#) **parent** (const [QModelIndex](#) &index) const override
- [Album](#) \* **rootAlbum** () const
- [RootAlbumBehavior](#) [rootAlbumBehavior](#) () const
- [QModelIndex](#) [rootAlbumIndex](#) () const
- int **rowCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- void [setDragDropHandler](#) ([AlbumModelDragDropHandler](#) \*handler)
- void [setDropIndex](#) (const [QModelIndex](#) &index)
- [Qt::DropAction](#) **supportedDropActions** () const override

**Protected Slots**

- void **slotAlbumMoved** ([Album](#) \*album)

**Protected Slots inherited from [Digikam::AbstractSpecificAlbumModel](#)**

- void **slotGotThumbnailFromIcon** ([Album](#) \*album, const QPixmap &thumbnail)
- void **slotReloadThumbnails** ()
- void **slotThumbnailLost** ([Album](#) \*album)

**Protected Slots inherited from [Digikam::AbstractAlbumModel](#)**

- void **slotAlbumAboutToBeAdded** ([Album](#) \*album, [Album](#) \*parent, [Album](#) \*prev)
- void **slotAlbumAboutToBeDeleted** ([Album](#) \*album)
- void **slotAlbumAdded** ([Album](#) \*)
- void **slotAlbumHasBeenDeleted** ([Album](#) \*album)
- void **slotAlbumIconChanged** ([Album](#) \*album)
- void **slotAlbumRenamed** ([Album](#) \*album)
- void **slotAlbumsCleared** ()

**Protected Member Functions**

- void **albumCleared** ([Album](#) \*album) override  
*Notification when an entry is removed.*
- QVariant **albumData** ([Album](#) \*a, int role) const override  
*Reimplemented from parent classes.*
- virtual [Album](#) \* **albumForId** (int id) const =0  
*need to implement in subclass*
- virtual QString **albumName** ([Album](#) \*a) const  
*Can reimplement in subclass.*
- void **allAlbumsCleared** () override  
*Notification when all entries are removed.*
- void **setCount** ([Album](#) \*album, int count)  
*If you do not use setCountHash, excludeChildrenCount and includeChildrenCount, you can set a count here.*
- void **setup** ()

**Protected Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)**

- QString **columnHeader** () const override  
*For subclassing convenience: A part of the implementation of headerData()*
- void **emitDataChangedForChildren** ([Album](#) \*album)
- void **setColumnHeader** (const QString &header)
- void **setupThumbnailLoading** ()  
*You need to call this from your constructor if you intend to load the thumbnail facilities of this class.*

## Protected Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- virtual QVariant [decorationRoleData](#) ([Album \\*a](#)) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual bool [filterAlbum](#) ([Album \\*album](#)) const
- virtual QVariant [fontRoleData](#) ([Album \\*a](#)) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual Qt::ItemFlags [itemFlags](#) ([Album \\*album](#)) const  
*For subclassing convenience: A part of the implementation of itemFlags()*
- void [setEnableDrag](#) (bool enable)
- void [setEnableDrop](#) (bool enable)
- void [setFaceTagModel](#) (bool enable)
- virtual QVariant [sortRoleData](#) ([Album \\*a](#)) const  
*For subclassing convenience: A part of the implementation of data()*

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractAlbumModel](#)

- enum [AlbumDataRole](#) {  
[AlbumTitleRole](#) = Qt::UserRole , [AlbumTypeRole](#) = Qt::UserRole + 1 , [AlbumPointerRole](#) = Qt::UserRole + 2  
, [AlbumIdRole](#) = Qt::UserRole + 3 ,  
[AlbumGlobalIdRole](#) = Qt::UserRole + 4 , [AlbumSortRole](#) = Qt::UserRole + 5 }
- enum [RootAlbumBehavior](#) { [IncludeRootAlbum](#) , [IgnoreRootAlbum](#) }

## Static Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- static [Album \\* retrieveAlbum](#) (const [QModelIndex](#) &index)

## Protected Attributes inherited from [Digikam::AbstractSpecificAlbumModel](#)

- [QString m\\_columnHeader](#)

## 6.9.1 Member Function Documentation

### 6.9.1.1 albumCleared()

```
void Digikam::AbstractCountingAlbumModel::albumCleared (
    Album * ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).

### 6.9.1.2 albumCount()

```
int Digikam::AbstractCountingAlbumModel::albumCount (
    Album * album ) const [virtual]
```

Returns the number of included items for this album.

#### Returns

positive value or -1 if unknown

### 6.9.1.3 albumData()

```
QVariant Digikam::AbstractCountingAlbumModel::albumData (
    Album * a,
    int role ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).

Reimplemented in [Digikam::AlbumModel](#), [Digikam::TagModel](#), and [Digikam::SearchModel](#).

### 6.9.1.4 albumForId()

```
virtual Album * Digikam::AbstractCountingAlbumModel::albumForId (
    int id ) const [protected], [pure virtual]
```

Implemented in [Digikam::AlbumModel](#), [Digikam::TagModel](#), [Digikam::SearchModel](#), and [Digikam::DateAlbumModel](#).

### 6.9.1.5 albumName()

```
QString Digikam::AbstractCountingAlbumModel::albumName (
    Album * a ) const [protected], [virtual]
```

Reimplemented in [Digikam::DateAlbumModel](#).

### 6.9.1.6 allAlbumsCleared()

```
void Digikam::AbstractCountingAlbumModel::allAlbumsCleared ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).

### 6.9.1.7 excludeChildrenCount

```
void Digikam::AbstractCountingAlbumModel::excludeChildrenCount (
    const QModelIndex & index ) [slot]
```

Displays only the count of the album, without adding child albums' counts. This is the default. Can connect to QTreeView's expanded() signal.

### 6.9.1.8 includeChildrenCount

```
void Digikam::AbstractCountingAlbumModel::includeChildrenCount (
    const QModelIndex & index ) [slot]
```

Displays sum of the count of the album and child albums' counts. Can connect to QTreeView's collapsed() signal.

### 6.9.1.9 setCountHash

```
void Digikam::AbstractCountingAlbumModel::setCountHash (
    const QHash< int, int > & idCountHash ) [slot]
```

Enable displaying the count. Set a map of album id -> count (excluding children). If an album is not contained, no count is displayed. To display a count of 0, there must be an entry album id -> 0.

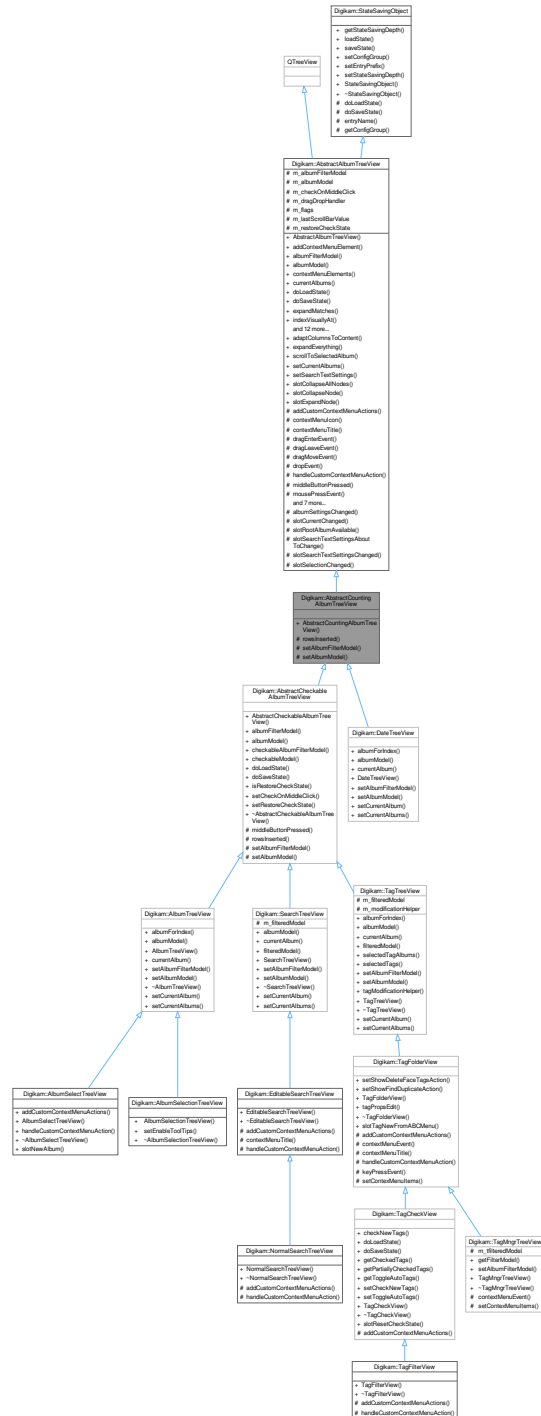
### 6.9.1.10 setup()

```
void Digikam::AbstractCountingAlbumModel::setup ( ) [protected]
```

Call this method in children class constructors to init signal/slots connections.

## 6.10 Digikam::AbstractCountingAlbumTreeView Class Reference

Inheritance diagram for Digikam::AbstractCountingAlbumTreeView:



### Public Member Functions

- `AbstractCountingAlbumTreeView` (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- void **doLoadState** () override
- void **doSaveState** () override
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Protected Member Functions

- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **setAlbumFilterModel** ([AlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AbstractCountingAlbumModel](#) \*const model)



## Protected Member Functions inherited from Digikam::AbstractAlbumTreeView

- virtual void [addCustomContextMenuActions](#) (ContextMenuHelper &cmh, Album \*album)
- virtual QPixmap [contextMenuIcon](#) () const
- virtual QString [contextMenuTitle](#) () const
- void **dragEnterEvent** (QDragEnterEvent \*e) override
- void **dragLeaveEvent** (QDragLeaveEvent \*e) override
- void **dragMoveEvent** (QDragMoveEvent \*e) override
- void **dropEvent** (QDropEvent \*e) override
- virtual void [handleCustomContextMenuAction](#) (QAction \*action, const AlbumPointer< Album > &album)
- virtual void **middleButtonPressed** (Album \*a)
- void **mousePressEvent** (QMouseEvent \*e) override

*Other helper methods.*

- virtual QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &index, int start, int end) override
- void **setAlbumFilterModel** (AlbumFilterModel \*const filterModel)
- void **setAlbumModel** (AbstractSpecificAlbumModel \*const model)
- virtual bool [showContextMenuAt](#) (QContextMenuEvent \*event, Album \*albumForEvent)
- void **startDrag** (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from Digikam::StateSavingObject

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## Additional Inherited Members

## Public Types inherited from Digikam::AbstractAlbumTreeView

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from Digikam::StateSavingObject

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots inherited from Digikam::AbstractAlbumTreeView

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)
- void **slotCollapseAllNodes** ()  
*slotCollapseAllNodes - collapse all nodes without root node*
- void **slotCollapseNode** ()  
*slotCollapseNode - collapse recursively selected nodes*
- void **slotExpandNode** ()  
*slotExpandNode - expands recursively selected nodes*

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< [Album](#) \* > &selectedAlbums)

## Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

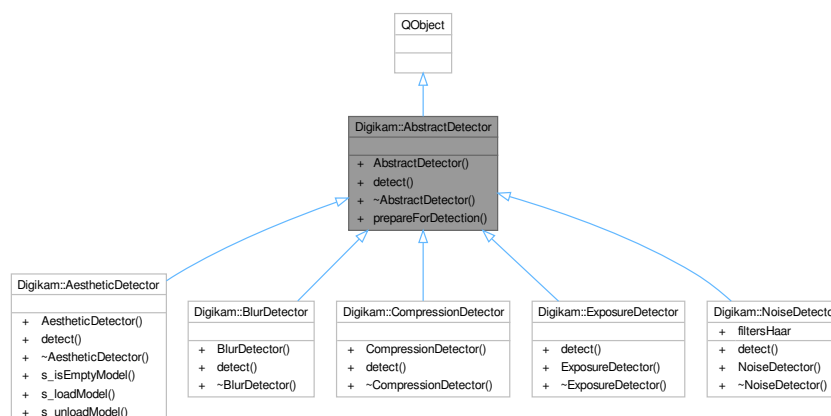
- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* [m\\_albumFilterModel](#) = nullptr
- [AbstractSpecificAlbumModel](#) \* [m\\_albumModel](#) = nullptr
- bool [m\\_checkOnMiddleClick](#) = false
- [AlbumModelDragDropHandler](#) \* [m\\_dragDropHandler](#) = nullptr
- Flags [m\\_flags](#) = DefaultFlags
- int [m\\_lastScrollBarValue](#) = 0
- bool [m\\_restoreCheckState](#) = false

## 6.11 Digikam::AbstractDetector Class Reference

Inheritance diagram for Digikam::AbstractDetector:



### Public Member Functions

- **AbstractDetector** (QObject \*const parent=nullptr)
- virtual float **detect** (const cv::Mat &image) const =0

## Static Public Member Functions

- static cv::Mat [prepareForDetection](#) (const [DImg](#) &inputImage)

### 6.11.1 Member Function Documentation

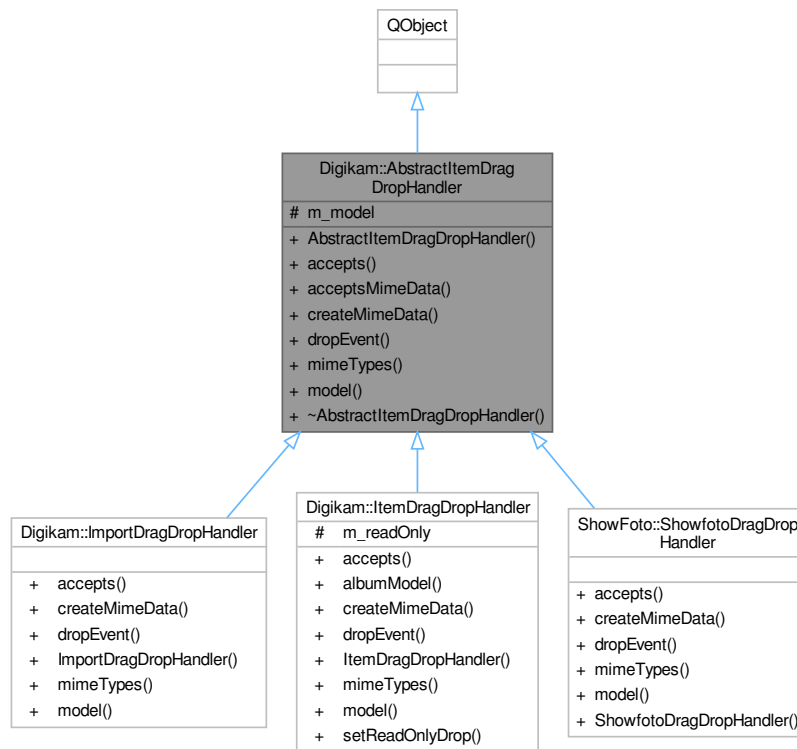
#### 6.11.1.1 prepareForDetection()

```
cv::Mat Digikam::AbstractDetector::prepareForDetection (
    const DImg & inputImage ) [static]
```

NOTE: Maybe this function will move to `read_image()` of `imagequalityparser` in case all detectors of IQS use `cv::Mat`

## 6.12 Digikam::AbstractItemDragDropHandler Class Reference

Inheritance diagram for Digikam::AbstractItemDragDropHandler:



## Public Member Functions

- **AbstractItemDragDropHandler** (`QAbstractItemModel *const model`)
- virtual `Qt::DropAction` [accepts](#) (`const QDropEvent *e`, `const QModelIndex &dropIndex`)
- virtual `bool` [acceptsMimeData](#) (`const QMimeData *data`)
- virtual `QMimeData *` [createMimeData](#) (`const QList< QModelIndex > &`)
- virtual `bool` [dropEvent](#) (`QAbstractItemView *view`, `const QDropEvent *e`, `const QModelIndex &droppedOn`)
- virtual `QStringList` [mimeTypes](#) () const
- `QAbstractItemModel *` [model](#) () const

## Protected Attributes

- `QAbstractItemModel * m_model = nullptr`

## 6.12.1 Member Function Documentation

### 6.12.1.1 `accepts()`

```
Qt::DropAction Digikam::AbstractItemDragDropHandler::accepts (
    const QDropEvent * e,
    const QModelIndex & dropIndex ) [virtual]
```

Returns if the given mime data is accepted for drop on dropIndex. Returns the proposed action, or `Qt::IgnoreAction` if not accepted.

Reimplemented in [Digikam::ImportDragDropHandler](#), [Digikam::ItemDragDropHandler](#), and [ShowFoto::ShowfotoDragDropHandler](#).

### 6.12.1.2 `acceptsMimeData()`

```
bool Digikam::AbstractItemDragDropHandler::acceptsMimeData (
    const QMimeData * data ) [virtual]
```

Returns if the given mime data can be handled. `acceptsMimeData` shall return true if a drop of the given mime data will be accepted on any index or place at all. If this returns false, the more specific method [accepts\(\)](#) will not be called for this drag. The default implementation uses [mimeTypes\(\)](#) to check for supported mime types. There is usually no need to reimplement this.

### 6.12.1.3 `createMimeData()`

```
QMimeData * Digikam::AbstractItemDragDropHandler::createMimeData (
    const QList< QModelIndex > & ) [virtual]
```

Create a mime data object for starting a drag from the given Albums

Reimplemented in [Digikam::ImportDragDropHandler](#), [Digikam::ItemDragDropHandler](#), and [ShowFoto::ShowfotoDragDropHandler](#).

### 6.12.1.4 `dropEvent()`

```
bool Digikam::AbstractItemDragDropHandler::dropEvent (
    QAbstractItemView * view,
    const QDropEvent * e,
    const QModelIndex & droppedOn ) [virtual]
```

Gives the view and the occurring drop event. The index is the index where the drop was dropped on. It may be invalid (dropped on decoration, viewport) Returns true if the event is to be accepted.

Reimplemented in [Digikam::ImportDragDropHandler](#), [Digikam::ItemDragDropHandler](#), and [ShowFoto::ShowfotoDragDropHandler](#).

### 6.12.1.5 mimeTypees()

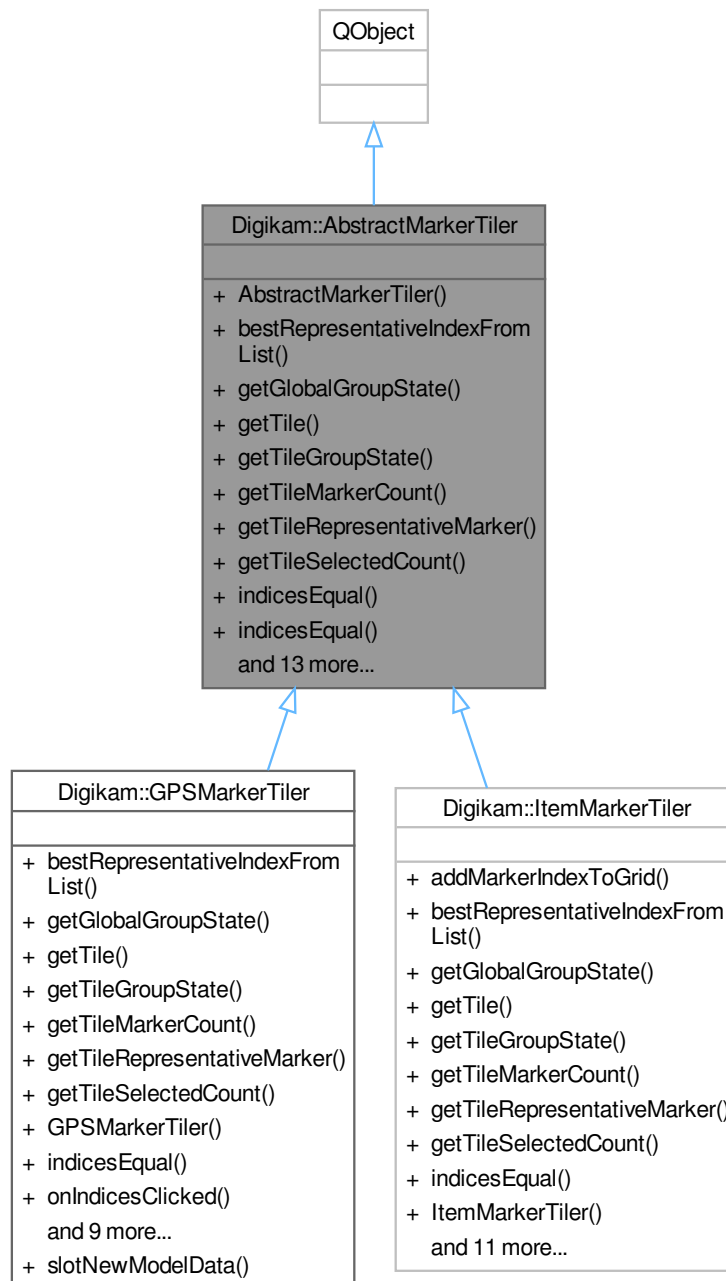
```
QStringList Digikam::AbstractItemDragDropHandler::mimeTypees ( ) const [virtual]
```

Returns the supported mime types. Called by the default implementation of model's [mimeTypees\(\)](#).

Reimplemented in [Digikam::ImportDragDropHandler](#), [Digikam::ItemDragDropHandler](#), and [ShowFoto::ShowfotoDragDropHandler](#).

## 6.13 Digikam::AbstractMarkerTiler Class Reference

Inheritance diagram for Digikam::AbstractMarkerTiler:



### Classes

- class [ClickInfo](#)
- class [NonEmptyIterator](#)
- class [Tile](#)

## Public Types

- enum **TilerFlag** { **FlagNull** = 0 , **FlagMovable** = 1 }

## Signals

- void **signalThumbnailAvailableForIndex** (const QVariant &index, const QPixmap &pixmap)
- void **signalTilesOrSelectionChanged** ()

## Public Member Functions

- **AbstractMarkerTiler** (QObject \*const parent=nullptr)
- virtual QVariant **bestRepresentativeIndexFromList** (const QList< QVariant > &indices, const int sortKey)=0
- virtual GeoGroupState **getGlobalGroupState** ()=0
- virtual **Tile** \* **getTile** (const **TileIndex** &tileIndex, const bool stopIfEmpty)=0
- virtual GeoGroupState **getTileGroupState** (const **TileIndex** &tileIndex)=0
- virtual int **getTileMarkerCount** (const **TileIndex** &tileIndex)=0
- virtual QVariant **getTileRepresentativeMarker** (const **TileIndex** &tileIndex, const int sortKey)=0  
*These should be implemented for thumbnail handling.*
- virtual int **getTileSelectedCount** (const **TileIndex** &tileIndex)=0
- bool **indicesEqual** (const QList &a, const QList &b, const int upToLevel) const
- virtual bool **indicesEqual** (const QVariant &a, const QVariant &b) const =0
- bool **isDirty** () const
- virtual void **onIndicesClicked** (const **ClickInfo** &clickInfo)  
*These can be implemented if you want to react to actions in geolocation interface.*
- virtual void **onIndicesMoved** (const **TileIndex::List** &tileIndicesList, const **GeoCoordinates** &target←Coordinates, const QPersistentModelIndex &targetSnapIndex)
- virtual QPixmap **pixmapFromRepresentativeIndex** (const QVariant &index, const QSize &size)=0
- virtual void **prepareTiles** (const **GeoCoordinates** &upperLeft, const **GeoCoordinates** &lowerRight, int level)=0
- virtual void **regenerateTiles** ()=0
- void **resetRootTile** ()
- **Tile** \* **rootTile** ()
- virtual void **setActive** (const bool state)=0
- void **setDirty** (const bool state=true)
- virtual **Tile** \* **tileNew** ()=0
- virtual TilerFlags **tilerFlags** () const  
*These have to be implemented.*

### 6.13.1 Member Function Documentation

#### 6.13.1.1 bestRepresentativeIndexFromList()

```
virtual QVariant Digikam::AbstractMarkerTiler::bestRepresentativeIndexFromList (
    const QList< QVariant > & indices,
    const int sortKey ) [pure virtual]
```

Implemented in [Digikam::GPSMarkerTiler](#).

### 6.13.1.2 `getTile()`

```
virtual Tile * Digikam::AbstractMarkerTiler::getTile (
    const TileIndex & tileIndex,
    const bool stopIfEmpty ) [pure virtual]
```

Implemented in [Digikam::GPSMarkerTiler](#).

### 6.13.1.3 `getTileGroupState()`

```
virtual GeoGroupState Digikam::AbstractMarkerTiler::getTileGroupState (
    const TileIndex & tileIndex ) [pure virtual]
```

Implemented in [Digikam::GPSMarkerTiler](#).

### 6.13.1.4 `getTileRepresentativeMarker()`

```
virtual QVariant Digikam::AbstractMarkerTiler::getTileRepresentativeMarker (
    const TileIndex & tileIndex,
    const int sortKey ) [pure virtual]
```

Implemented in [Digikam::ItemMarkerTiler](#), and [Digikam::GPSMarkerTiler](#).

### 6.13.1.5 `indicesEqual()`

```
virtual bool Digikam::AbstractMarkerTiler::indicesEqual (
    const QVariant & a,
    const QVariant & b ) const [pure virtual]
```

Implemented in [Digikam::GPSMarkerTiler](#).

### 6.13.1.6 `onIndicesClicked()`

```
void Digikam::AbstractMarkerTiler::onIndicesClicked (
    const ClickInfo & clickInfo ) [virtual]
```

Reimplemented in [Digikam::ItemMarkerTiler](#), and [Digikam::GPSMarkerTiler](#).

### 6.13.1.7 `QPixmapFromRepresentativeIndex()`

```
virtual QPixmap Digikam::AbstractMarkerTiler::PixmapFromRepresentativeIndex (
    const QVariant & index,
    const QSize & size ) [pure virtual]
```

Implemented in [Digikam::GPSMarkerTiler](#).



### 6.13.1.8 prepareTiles()

```
virtual void Digikam::AbstractMarkerTiler::prepareTiles (
    const GeoCoordinates & upperLeft,
    const GeoCoordinates & lowerRight,
    int level ) [pure virtual]
```

Implemented in [Digikam::GPSMarkerTiler](#).

### 6.13.1.9 setActive()

```
virtual void Digikam::AbstractMarkerTiler::setActive (
    const bool state ) [pure virtual]
```

Implemented in [Digikam::GPSMarkerTiler](#).

### 6.13.1.10 tilerFlags()

```
AbstractMarkerTiler::TilerFlags Digikam::AbstractMarkerTiler::tilerFlags ( ) const [virtual]
```

Reimplemented in [Digikam::ItemMarkerTiler](#).

## 6.14 Digikam::AbstractMarkerTiler::ClickInfo Class Reference

### Public Attributes

- GeoMouseModes **currentMouseMode**
- GeoGroupState **groupSelectionMode**
- QVariant **representativeIndex**
- TileIndex::List **tileIndicesList**

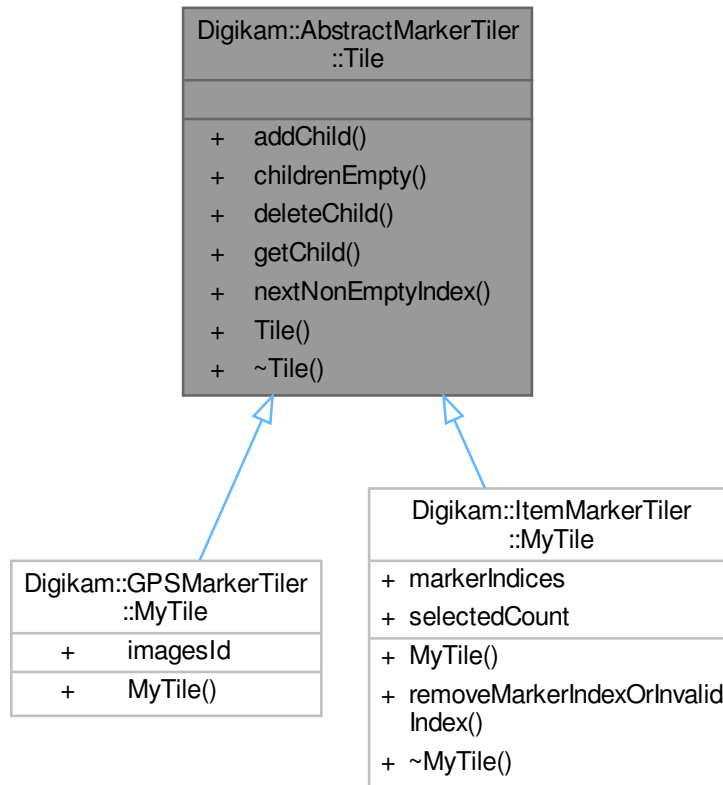
## 6.15 Digikam::AbstractMarkerTiler::NonEmptyIterator Class Reference

### Public Member Functions

- bool **atEnd** () const
- [TileIndex](#) **currentIndex** () const
- [AbstractMarkerTiler](#) \* **model** () const
- [TileIndex](#) **nextIndex** ()
- **NonEmptyIterator** ([AbstractMarkerTiler](#) \*const model, const int level)
- **NonEmptyIterator** ([AbstractMarkerTiler](#) \*const model, const int level, const [GeoCoordinates::PairList](#) &normalizedMapBounds)
- **NonEmptyIterator** ([AbstractMarkerTiler](#) \*const model, const int level, const [TileIndex](#) &startIndex, const [TileIndex](#) &endIndex)

## 6.16 Digikam::AbstractMarkerTiler::Tile Class Reference

Inheritance diagram for Digikam::AbstractMarkerTiler::Tile:

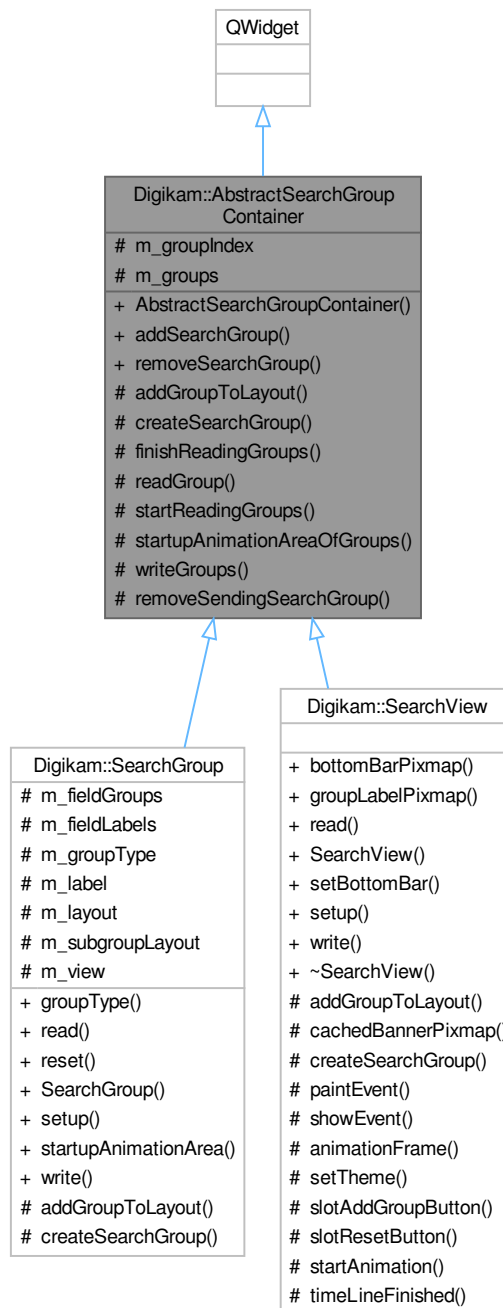


### Public Member Functions

- `Tile * addChild` (const int linearIndex, `Tile *tilePointer`)
- bool `childrenEmpty` () const
- void `deleteChild` (`Tile *const childTile`, const int knownLinearIndex=-1)  
*Sets the pointer to a child tile to zero and deletes the child.*
- `Tile * getChild` (const int linearIndex)
- int `nextNonEmptyIndex` (int linearIndex) const  
*returns the next non empty child index or -1.*

## 6.17 Digikam::AbstractSearchGroupContainer Class Reference

Inheritance diagram for Digikam::AbstractSearchGroupContainer:



### Public Slots

- `SearchGroup * addSearchGroup ()`
- `void removeSearchGroup (SearchGroup *group)`

## Public Member Functions

- [AbstractSearchGroupContainer](#) (QWidget \*const parent=nullptr)

## Protected Slots

- void **removeSendingSearchGroup** ()

## Protected Member Functions

- virtual void **addGroupToLayout** ([SearchGroup](#) \*group)=0  
*Re-implement: Adds a newly created group to the layout structures.*
- virtual [SearchGroup](#) \* **createSearchGroup** ()=0  
*Re-implement: create and setup a search group.*
- void **finishReadingGroups** ()  
*Call when the XML part is finished.*
- void **readGroup** ([SearchXmlCachingReader](#) &reader)  
*Call when a group element is the current element.*
- void **startReadingGroups** ([SearchXmlCachingReader](#) &reader)  
*Call before reading the XML part that could contain group elements.*
- QList< QRect > **startupAnimationAreaOfGroups** () const  
*Collects the data from the same method of all contained groups (position relative to this widget)*
- void **writeGroups** ([SearchXmlWriter](#) &writer) const  
*Write contained groups to writer.*

## Protected Attributes

- int **m\_groupIndex** = 0
- QList< [SearchGroup](#) \* > **m\_groups**

## 6.17.1 Constructor & Destructor Documentation

### 6.17.1.1 AbstractSearchGroupContainer()

```
Digikam::AbstractSearchGroupContainer::AbstractSearchGroupContainer (
    QWidget *const parent = nullptr ) [explicit]
```

Abstract base class for classes that contain SearchGroups To contain common code of [SearchView](#) and [SearchGroup](#), as SearchGroups can have subgroups.

## 6.17.2 Member Function Documentation

### 6.17.2.1 addGroupToLayout()

```
virtual void Digikam::AbstractSearchGroupContainer::addGroupToLayout (
    SearchGroup * group ) [protected], [pure virtual]
```

Implemented in [Digikam::SearchGroup](#), and [Digikam::SearchView](#).

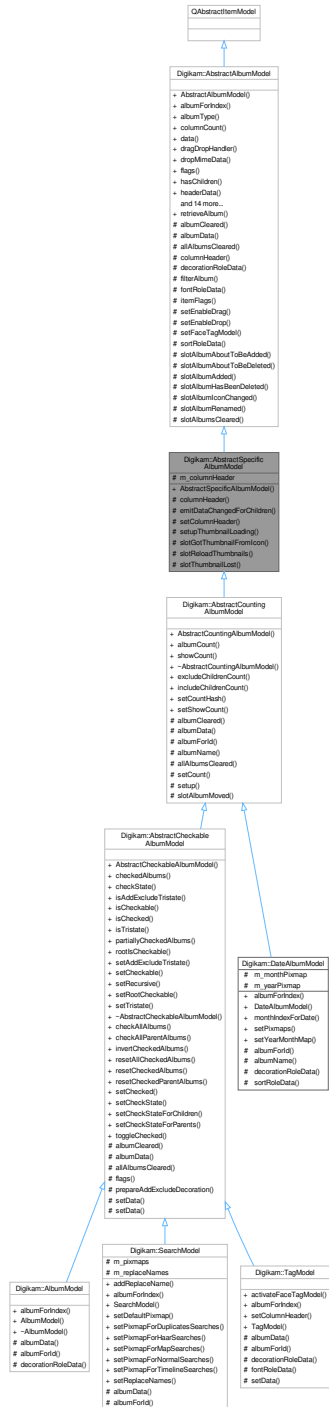
## 6.17.2.2 createSearchGroup()

```
virtual SearchGroup * Digikam::AbstractSearchGroupContainer::createSearchGroup ( ) [protected],
[pure virtual]
```

Implemented in [Digikam::SearchGroup](#), and [Digikam::SearchView](#).

## 6.18 Digikam::AbstractSpecificAlbumModel Class Reference

Inheritance diagram for Digikam::AbstractSpecificAlbumModel:



## Public Member Functions

- **AbstractSpecificAlbumModel** ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)

*Abstract base class, do not instantiate.*

## Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- [AbstractAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [Album](#) \* [albumForIndex](#) (const [QModelIndex](#) &index) const
- [Album::Type](#) [albumType](#) () const
- int **columnCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [AlbumModelDragDropHandler](#) \* [dragDropHandler](#) () const
- bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role=[Qt::DisplayRole](#)) const override
- [QModelIndex](#) **index** (int row, int column, const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QModelIndex](#) [indexForAlbum](#) ([Album](#) \*album) const
- bool **isFaceTagModel** () const
- [QMimeData](#) \* **mimeData** (const [QModelIndexList](#) &indexes) const override
- [QStringList](#) **mimeTypes** () const override
- [QModelIndex](#) **parent** (const [QModelIndex](#) &index) const override
- [Album](#) \* **rootAlbum** () const
- [RootAlbumBehavior](#) [rootAlbumBehavior](#) () const
- [QModelIndex](#) [rootAlbumIndex](#) () const
- int **rowCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- void [setDragDropHandler](#) ([AlbumModelDragDropHandler](#) \*handler)
- void [setDropIndex](#) (const [QModelIndex](#) &index)
- [Qt::DropActions](#) **supportedDropActions** () const override

## Protected Slots

- void **slotGotThumbnailFromIcon** ([Album](#) \*album, const [QPixmap](#) &thumbnail)
- void **slotReloadThumbnails** ()
- void **slotThumbnailLost** ([Album](#) \*album)

## Protected Slots inherited from [Digikam::AbstractAlbumModel](#)

- void **slotAlbumAboutToBeAdded** ([Album](#) \*album, [Album](#) \*parent, [Album](#) \*prev)
- void **slotAlbumAboutToBeDeleted** ([Album](#) \*album)
- void **slotAlbumAdded** ([Album](#) \*)
- void **slotAlbumHasBeenDeleted** ([Album](#) \*album)
- void **slotAlbumIconChanged** ([Album](#) \*album)
- void **slotAlbumRenamed** ([Album](#) \*album)
- void **slotAlbumsCleared** ()

### Protected Member Functions

- QString `columnHeader` () const override  
*For subclassing convenience: A part of the implementation of `headerData()`*
- void `emitDataChangedForChildren` (Album \*album)
- void `setColumnHeader` (const QString &header)
- void `setupThumbnailLoading` ()  
*You need to call this from your constructor if you intend to load the thumbnail facilities of this class.*

### Protected Member Functions inherited from Digikam::AbstractAlbumModel

- virtual void `albumCleared` (Album \*)  
*Notification when an entry is removed.*
- virtual QVariant `albumData` (Album \*a, int role) const  
*For subclassing convenience: A part of the implementation of `data()`*
- virtual void `allAlbumsCleared` ()  
*Notification when all entries are removed.*
- virtual QVariant `decorationRoleData` (Album \*a) const  
*For subclassing convenience: A part of the implementation of `data()`*
- virtual bool `filterAlbum` (Album \*album) const
- virtual QVariant `fontRoleData` (Album \*a) const  
*For subclassing convenience: A part of the implementation of `data()`*
- virtual Qt::ItemFlags `itemFlags` (Album \*album) const  
*For subclassing convenience: A part of the implementation of `itemFlags()`*
- void `setEnabledDrag` (bool enable)
- void `setEnabledDrop` (bool enable)
- void `setFaceTagModel` (bool enable)
- virtual QVariant `sortRoleData` (Album \*a) const  
*For subclassing convenience: A part of the implementation of `data()`*

### Protected Attributes

- QString `m_columnHeader`

### Additional Inherited Members

### Public Types inherited from Digikam::AbstractAlbumModel

- enum `AlbumDataRole` {  
  `AlbumTitleRole` = Qt::UserRole , `AlbumTypeRole` = Qt::UserRole + 1 , `AlbumPointerRole` = Qt::UserRole + 2  
  , `AlbumIdRole` = Qt::UserRole + 3 ,  
  `AlbumGlobalIdRole` = Qt::UserRole + 4 , `AlbumSortRole` = Qt::UserRole + 5 }
- enum `RootAlbumBehavior` { `IncludeRootAlbum` , `IgnoreRootAlbum` }

### Signals inherited from Digikam::AbstractAlbumModel

- void `rootAlbumAvailable` ()





**Protected Slots**

- virtual void [slotEntered](#) (const QModelIndex &index)
- virtual void [slotLayoutChanged](#) ()
- virtual void [slotReset](#) ()
- virtual void [slotRowsRemoved](#) (const QModelIndex &parent, int start, int end)
- virtual void [slotViewportEntered](#) ()

**Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)**

- virtual void [visualChange](#) ()

**Protected Member Functions**

- virtual bool [checkIndex](#) (const QModelIndex &index) const
- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- virtual QWidget \* [createWidget](#) ()=0
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- virtual void [widgetEnterEvent](#) ()
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- virtual void [widgetLeaveEvent](#) ()
- void [widgetLeaveNotifyMultiple](#) ()

**Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)**

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

**Protected Attributes**

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

**Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)**

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

**Additional Inherited Members****Signals inherited from [Digikam::ItemDelegateOverlay](#)**

- void [hideNotification](#) ()
- void [requestNotification](#) (const QModelIndex &index, const QString &message)
- void [update](#) (const QModelIndex &index)

## 6.19.1 Constructor & Destructor Documentation

### 6.19.1.1 AbstractWidgetDelegateOverlay()

```
Digikam::AbstractWidgetDelegateOverlay::AbstractWidgetDelegateOverlay (
    QObject *const parent ) [explicit]
```

This class provides functionality for using a widget in an overlay. You must reimplement at least `createWidget` to return your widget. Per default it will be shown when the cursor enters an index and hidden when left. Reimplement `slotEntered()` and `mouseMove()` for more fine grained control.

## 6.19.2 Member Function Documentation

### 6.19.2.1 checkIndex()

```
bool Digikam::AbstractWidgetDelegateOverlay::checkIndex (
    const QModelIndex & index ) const [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented in [Digikam::AssignNameOverlay](#), [Digikam::FaceRejectionOverlay](#), [Digikam::GroupIndicatorOverlay](#), [Digikam::ItemCoordinatesOverlay](#), [Digikam::ItemFullScreenOverlay](#), [Digikam::ItemRotateOverlay](#), [Digikam::ShowHideVersionsOverlay](#), [Digikam::ActionVersionsOverlay](#), [ShowFoto::ShowfotoCoordinatesOverlay](#), [Digikam::ImportCoordinatesOverlay](#), [Digikam::ImportLockOverlay](#), [Digikam::ImportDownloadOverlay](#), and [Digikam::ImportRotateOverlay](#).

### 6.19.2.2 checkIndexOnEnter()

```
bool Digikam::AbstractWidgetDelegateOverlay::checkIndexOnEnter (
    const QModelIndex & index ) const [protected]
```

Utility method called from `slotEntered`

### 6.19.2.3 createWidget()

```
virtual QWidget * Digikam::AbstractWidgetDelegateOverlay::createWidget ( ) [protected], [pure
virtual]
```

Create your widget here. When creating the object, pass `parentWidget()` as parent widget. Ownership of the object is passed. It will be deleted in `setActive(false)`.

Implemented in [Digikam::AssignNameOverlay](#), [Digikam::GroupIndicatorOverlay](#), [Digikam::ItemCoordinatesOverlay](#), [Digikam::ItemRatingOverlay](#), [Digikam::TagsLineEditOverlay](#), [Digikam::HoverButtonDelegateOverlay](#), [ShowFoto::ShowfotoCoordinatesOverlay](#), [Digikam::ImportCoordinatesOverlay](#), [Digikam::ImportLockOverlay](#), [Digikam::ImportDownloadOverlay](#), and [Digikam::ImportRatingOverlay](#).

### 6.19.2.4 hide()

```
void Digikam::AbstractWidgetDelegateOverlay::hide ( ) [protected], [virtual]
```

Called when the widget shall be hidden (mouse cursor left index, viewport, uninstalled etc.). Default implementation `hide()`s `m_widget`.

Reimplemented in [Digikam::ItemRatingOverlay](#), [Digikam::TagsLineEditOverlay](#), [Digikam::PersistentWidgetDelegateOverlay](#), and [Digikam::ImportRatingOverlay](#).

### 6.19.2.5 parentWidget()

```
QWidget * Digikam::AbstractWidgetDelegateOverlay::parentWidget ( ) const [protected]
```

Returns the widget to be used as parent for your widget created in [createWidget\(\)](#)

### 6.19.2.6 setActive()

```
void Digikam::AbstractWidgetDelegateOverlay::setActive (
    bool active ) [override], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::ItemDelegateOverlay](#).

Reimplemented in [Digikam::FaceRejectionOverlay](#), [Digikam::ItemCoordinatesOverlay](#), [Digikam::ItemFullScreenOverlay](#), [Digikam::ItemRotateOverlay](#), [Digikam::ItemSelectionOverlay](#), [Digikam::ShowHideVersionsOverlay](#), [Digikam::ActionVersionsOverlay](#), [Digikam::HoverButtonDelegateOverlay](#), [Digikam::PersistentWidgetDelegateOverlay](#), [ShowFoto::ShowfotoCoordinatesOverlay](#), [Digikam::ImportCoordinatesOverlay](#), [Digikam::ImportLockOverlay](#), [Digikam::ImportDownloadOverlay](#), [Digikam::ImportRotateOverlay](#), [Digikam::AssignNameOverlay](#), [Digikam::GroupIndicatorOverlay](#), [Digikam::ItemRatingOverlay](#), [Digikam::TagsLineEditOverlay](#), and [Digikam::ImportRatingOverlay](#).

### 6.19.2.7 slotEntered

```
void Digikam::AbstractWidgetDelegateOverlay::slotEntered (
    const QModelIndex & index ) [protected], [virtual], [slot]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented in [Digikam::GroupIndicatorOverlay](#), [Digikam::ItemCoordinatesOverlay](#), [Digikam::ItemRatingOverlay](#), [Digikam::TagsLineEditOverlay](#), [Digikam::PersistentWidgetDelegateOverlay](#), [ShowFoto::ShowfotoCoordinatesOverlay](#), [Digikam::ImportCoordinatesOverlay](#), [Digikam::ImportLockOverlay](#), [Digikam::ImportDownloadOverlay](#), and [Digikam::ImportRatingOverlay](#).

### 6.19.2.8 slotReset

```
void Digikam::AbstractWidgetDelegateOverlay::slotReset ( ) [protected], [virtual], [slot]
```

Default implementations of these three slots call [hide\(\)](#)

Reimplemented in [Digikam::PersistentWidgetDelegateOverlay](#).

### 6.19.2.9 viewportLeaveEvent()

```
void Digikam::AbstractWidgetDelegateOverlay::viewportLeaveEvent (
    QObject * obj,
    QEvent * event ) [protected], [virtual]
```

Called when a `QEvent::Leave` of the viewport is received. The default implementation [hide\(\)](#)s.

Reimplemented in [Digikam::AssignNameOverlay](#), and [Digikam::PersistentWidgetDelegateOverlay](#).

### 6.19.2.10 widgetEnterEvent()

```
void Digikam::AbstractWidgetDelegateOverlay::widgetEnterEvent ( ) [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented in [Digikam::AssignNameOverlay](#), [Digikam::FaceRejectionOverlay](#), [Digikam::ItemFullScreenOverlay](#), [Digikam::ItemRatingOverlay](#), [Digikam::ItemRotateOverlay](#), [Digikam::ImportRatingOverlay](#), and [Digikam::ImportRotateOverlay](#).

### 6.19.2.11 widgetEnterNotifyMultiple()

```
void Digikam::AbstractWidgetDelegateOverlay::widgetEnterNotifyMultiple (
    const QModelIndex & index ) [protected]
```

A sample implementation for above methods

## 6.20 Digikam::ActionCategorizedView Class Reference

Inheritance diagram for Digikam::ActionCategorizedView:



### Public Member Functions

- **ActionCategorizedView** (`QWidget *const parent=nullptr, bool autoScroll=false`)
- void **adjustGridSize** ()
- void **setupIconMode** ()

## Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual QModelIndexList [categorizedIndexesIn](#) (const QRect &rect) const
- virtual QModelIndex [categoryAt](#) (const QPoint &point) const
- [DCategoryDrawer](#) \* [categoryDrawer](#) () const
- virtual QItemSelectionRange [categoryRange](#) (const QModelIndex &index) const
- virtual QRect [categoryVisualRect](#) (const QModelIndex &index) const
- [DCategorizedView](#) (QWidget \*const parent=nullptr)
- QModelIndex [indexAt](#) (const QPoint &point) const override
- void [setCategoryDrawer](#) ([DCategoryDrawer](#) \*categoryDrawer)
- void [setDrawDraggedItems](#) (bool drawDraggedItems)
- void [setGridSize](#) (const QSize &size)
- void [setModel](#) (QAbstractItemModel \*model) override
- QRect [visualRect](#) (const QModelIndex &index) const override

## Protected Member Functions

- void [autoScroll](#) (float relativePos, QScrollBar \*scrollBar, QPropertyAnimation \*animation)
- int [autoScrollDuration](#) (float relativeDifference, QPropertyAnimation \*animation)
- void [leaveEvent](#) (QEvent \*e) override
- void [mouseMoveEvent](#) (QMouseEvent \*e) override

## Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void [dragLeaveEvent](#) (QDragLeaveEvent \*event) override
- void [dragMoveEvent](#) (QDragMoveEvent \*event) override
- void [dropEvent](#) (QDropEvent \*event) override
- void [leaveEvent](#) (QEvent \*event) override
- void [mouseMoveEvent](#) (QMouseEvent \*event) override
- void [mousePressEvent](#) (QMouseEvent \*event) override
- void [mouseReleaseEvent](#) (QMouseEvent \*event) override
- QModelIndex [moveCursor](#) (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void [paintEvent](#) (QPaintEvent \*event) override
- void [resizeEvent](#) (QResizeEvent \*event) override
- void [setSelection](#) (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void [startDrag](#) (Qt::DropActions supportedActions) override

## Protected Attributes

- bool [m\\_autoScroll](#) = false
- QPropertyAnimation \* [m\\_horizontalScrollAnimation](#) = nullptr
- QPropertyAnimation \* [m\\_verticalScrollAnimation](#) = nullptr

## Additional Inherited Members

## Public Slots inherited from [Digikam::DCategorizedView](#)

- void [reset](#) () override

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **rowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## 6.21 Digikam::ActionData Class Reference

### Public Types

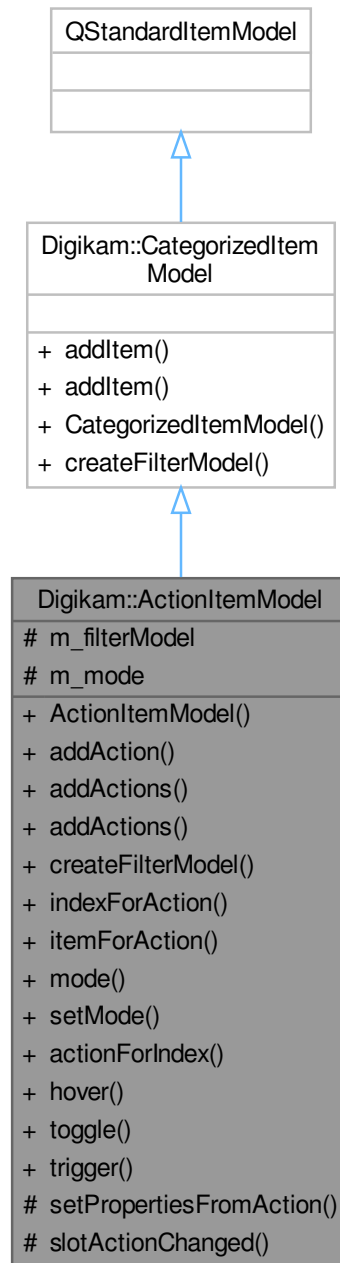
- enum **ActionStatus** {  
**None** = 0 , **BatchStarted** , **BatchDone** , **BatchFailed** ,  
**BatchSkipped** , **BatchCanceled** , **TaskDone** , **TaskFailed** ,  
**TaskCanceled** }

### Public Attributes

- QUrl **destUrl**
- QUrl **fileUrl**
- QString **message**
- bool **noWrite** = false
- ActionStatus **status** = None

## 6.22 Digikam::ActionItemModel Class Reference

Inheritance diagram for Digikam::ActionItemModel:



### Public Types

- enum **ExtraRoles** { **ItemActionRole** = Qt::UserRole + 10 }
- enum **MenuCategoryFlag** { **ToplevelMenuCategory** = 1 << 0 , **ParentMenuCategory** = 1 << 1 , **SortCategoriesAlphabetically** = 1 << 10 , **SortCategoriesByInsertionOrder** = 1 << 11 }



## Public Types inherited from [Digikam::CategorizedItemModel](#)

- enum [ExtraRoles](#) { [ItemOrderRole](#) = Qt::UserRole + 1 }

### Public Slots

- void [hover](#) (const QModelIndex &index)
- void [toggle](#) (const QModelIndex &index)
- void [trigger](#) (const QModelIndex &index)

### Public Member Functions

- [ActionItemModel](#) (QObject \*const parent=nullptr)
- QStandardItem \* [addAction](#) (QAction \*action, const QString &category, const QVariant &category←  
Sorting=QVariant())
- void [addActions](#) (QWidget \*widget)
- void [addActions](#) (QWidget \*widget, const QList< QAction \* > &actionWhiteList)
- [DCategorizedSortFilterProxyModel](#) \* [createFilterModel](#) () override
- QModelIndex [indexForAction](#) (QAction \*action) const
- QStandardItem \* [itemForAction](#) (QAction \*action) const
- MenuCategoryMode [mode](#) () const
- void [setMode](#) (MenuCategoryMode mode)

## Public Member Functions inherited from [Digikam::CategorizedItemModel](#)

- QStandardItem \* [addItem](#) (const QString &text, const QIcon &decoration, const QVariant &category, const  
QVariant &categorySorting=QVariant())
- QStandardItem \* [addItem](#) (const QString &text, const QVariant &category, const QVariant &category←  
Sorting=QVariant())
- [CategorizedItemModel](#) (QObject \*const parent=nullptr)

### Static Public Member Functions

- static QAction \* [actionForIndex](#) (const QModelIndex &index)

### Protected Slots

- void [slotActionChanged](#) ()

### Protected Member Functions

- void [setPropertiesFromAction](#) (QStandardItem \*item, QAction \*action)

### Protected Attributes

- [DCategorizedSortFilterProxyModel](#) \* [m\\_filterModel](#) = nullptr
- MenuCategoryMode [m\\_mode](#) = MenuCategoryMode([ToplevelMenuCategory](#) | [SortCategoriesAlphabetically](#))

## 6.22.1 Member Enumeration Documentation

### 6.22.1.1 MenuCategoryFlag

```
enum Digikam::ActionItemModel::MenuCategoryFlag
```

## Enumerator

ToplevelMenuCategory	The toplevel menu's text is used as category.
ParentMenuCategory	If the action is in a submenu, this menu's text is taken as category.
SortCategoriesAlphabetically	Sort categories alphabetically by category name.
SortCategoriesByInsertionOrder	Sort categories by the order they are added (found in the scanned menu)

## 6.22.2 Constructor & Destructor Documentation

### 6.22.2.1 ActionItemModel()

```
Digikam::ActionItemModel::ActionItemModel (
    QObject *const parent = nullptr ) [explicit]
```

This class is a [CategorizedItemModel](#) based on QActions, taking an action's text and icon for display and decoration. It is possible to retrieve an action for an index, and to call the action's slots from a given index.

## 6.22.3 Member Function Documentation

### 6.22.3.1 actionForIndex()

```
QAction * Digikam::ActionItemModel::actionForIndex (
    const QModelIndex & index ) [static]
```

Returns the action for the given index. The method can also be used for indices from proxy models.

### 6.22.3.2 createFilterModel()

```
DCategorizedSortFilterProxyModel * Digikam::ActionItemModel::createFilterModel ( ) [override],
[virtual]
```

Reimplemented from [Digikam::CategorizedItemModel](#).

### 6.22.3.3 hover

```
void Digikam::ActionItemModel::hover (
    const QModelIndex & index ) [slot]
```

These three slots will cause the slots of the referred action to be called. Connect here for example a view's signals. Note that you can also pass indices from proxy models.

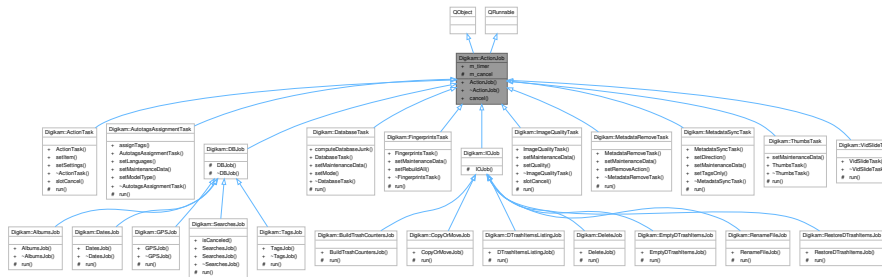
### 6.22.3.4 itemForAction()

```
QStandardItem * Digikam::ActionItemModel::itemForAction (
    QAction * action ) const
```

Returns the action for the given index. Note: these methods perform O(n).

## 6.23 Digikam::ActionJob Class Reference

Inheritance diagram for Digikam::ActionJob:



### Public Slots

- void [cancel](#) ()

### Signals

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Member Functions

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Public Attributes

- QElapsedTimer [m\\_timer](#)

### Protected Attributes

- bool [m\\_cancel](#) = false

## 6.23.1 Constructor & Destructor Documentation

### 6.23.1.1 ActionJob()

```
Digikam::ActionJob::ActionJob (
    QObject *const parent = nullptr ) [explicit]
```

Constructor which delegate deletion of QRunnable instance to [ActionThreadBase](#), not QThreadPool.

### 6.23.1.2 ~ActionJob()

```
Digikam::ActionJob::~~ActionJob ( ) [override]
```

Re-implement destructor in you implementation. Don't forget to cancel job.

## 6.23.2 Member Function Documentation

### 6.23.2.1 cancel

```
void Digikam::ActionJob::cancel ( ) [slot]
```

Call this method to cancel job.

### 6.23.2.2 signalDone

```
void Digikam::ActionJob::signalDone ( ) [signal]
```

Use this signal in your implementation to inform [ActionThreadBase](#) manager the job is done.

### 6.23.2.3 signalProgress

```
void Digikam::ActionJob::signalProgress (
    int ) [signal]
```

Use this signal in your implementation to inform [ActionThreadBase](#) manager the job progress

### 6.23.2.4 signalStarted

```
void Digikam::ActionJob::signalStarted ( ) [signal]
```

Use this signal in your implementation to inform [ActionThreadBase](#) manager that job is started

## 6.23.3 Member Data Documentation

### 6.23.3.1 m\_cancel

```
bool Digikam::ActionJob::m_cancel = false [protected]
```

You can use this boolean in your implementation to know if job must be canceled.

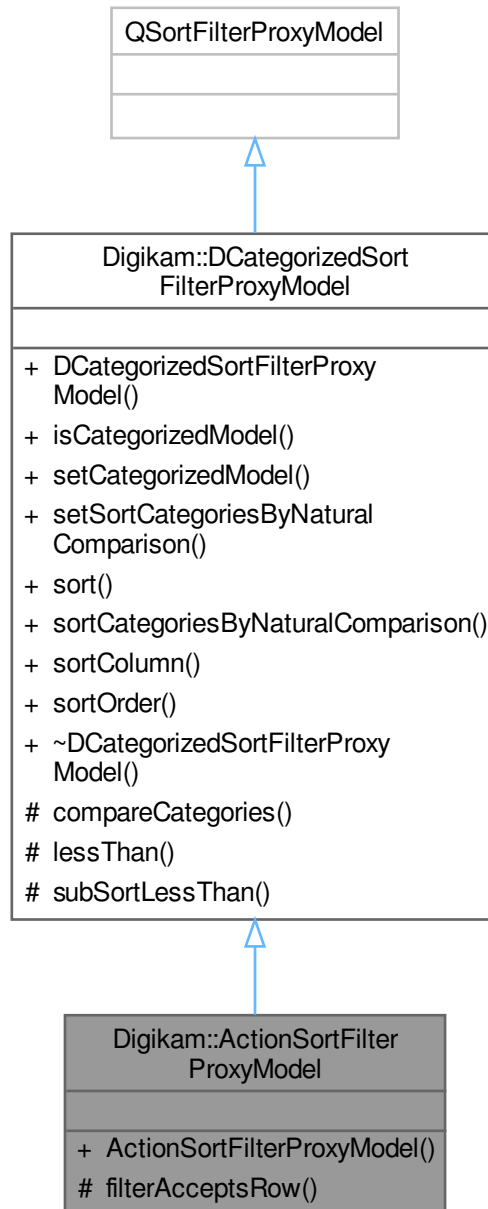
### 6.23.3.2 m\_timer

```
QElapsedTimer Digikam::ActionJob::m_timer
```

Timer to determine the running time of the job.

## 6.24 Digikam::ActionSortFilterProxyModel Class Reference

Inheritance diagram for Digikam::ActionSortFilterProxyModel:



### Public Member Functions

- **ActionSortFilterProxyModel** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- **DCategorizedSortFilterProxyModel** (QObject \*const parent=nullptr)
- bool [isCategorizedModel](#) () const
- void [setCategorizedModel](#) (bool categorizedModel)
- void [setSortCategoriesByNaturalComparison](#) (bool [sortCategoriesByNaturalComparison](#))
- void [sort](#) (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool [sortCategoriesByNaturalComparison](#) () const
- int [sortColumn](#) () const
- Qt::SortOrder [sortOrder](#) () const

## Protected Member Functions

- bool [filterAcceptsRow](#) (int source\_row, const QModelIndex &source\_parent) const override

## Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- virtual int [compareCategories](#) (const QModelIndex &left, const QModelIndex &right) const
- bool [lessThan](#) (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool [subSortLessThan](#) (const QModelIndex &left, const QModelIndex &right) const

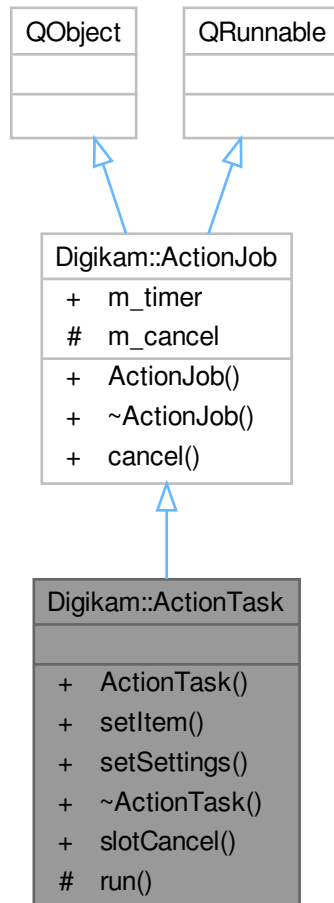
## Additional Inherited Members

## Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## 6.25 Digikam::ActionTask Class Reference

Inheritance diagram for Digikam::ActionTask:



### Public Slots

- void **slotCancel** ()

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Signals

- void **signalFinished** (const [Digikam::ActionData](#) &ad)
- void **signalStarting** (const [Digikam::ActionData](#) &ad)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Member Functions

- void **setItem** (const [AssignedBatchTools](#) &tools)
- void **setSettings** (const [QueueSettings](#) &settings)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void **run** () override

## Additional Inherited Members

## Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

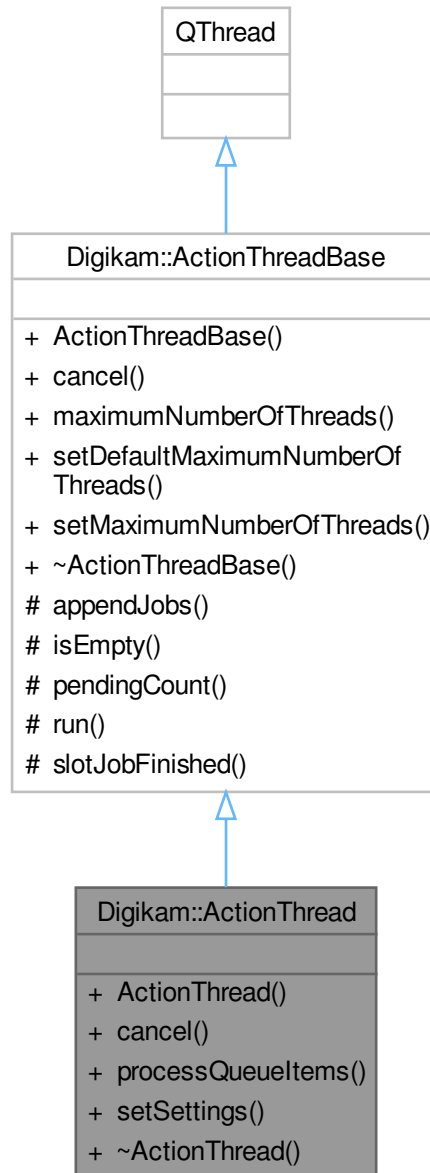
## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false



## 6.26 Digikam::ActionThread Class Reference

Inheritance diagram for Digikam::ActionThread:



### Signals

- void [signalCancelActionTask](#) ()
- void [signalFinished](#) (const [Digikam::ActionData](#) &ad)
- void [signalQueueProcessed](#) ()
- void [signalStarting](#) (const [Digikam::ActionData](#) &ad)

## Public Member Functions

- **ActionThread** (QObject \*const parent)
- void **cancel** ()
- void **processQueueItems** (const QList< [AssignedBatchTools](#) > &items)
- void **setSettings** (const [QueueSettings](#) &settings)

## Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void **slotJobFinished** ()

## Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void **appendJobs** (const [ActionJobCollection](#) &jobs)
- bool **isEmpty** () const
- int **pendingCount** () const
- void **run** () override

## 6.26.1 Member Function Documentation

### 6.26.1.1 signalCancelActionTask

```
void Digikam::ActionThread::signalCancelActionTask ( ) [signal]
```

Signal to emit to sub-tasks to cancel processing.

### 6.26.1.2 signalFinished

```
void Digikam::ActionThread::signalFinished (
    const Digikam::ActionData & ad ) [signal]
```

Emit when an item from a queue have been processed.

### 6.26.1.3 signalQueueProcessed

```
void Digikam::ActionThread::signalQueueProcessed ( ) [signal]
```

Emit when a queue have been fully processed (all items from queue are finished).

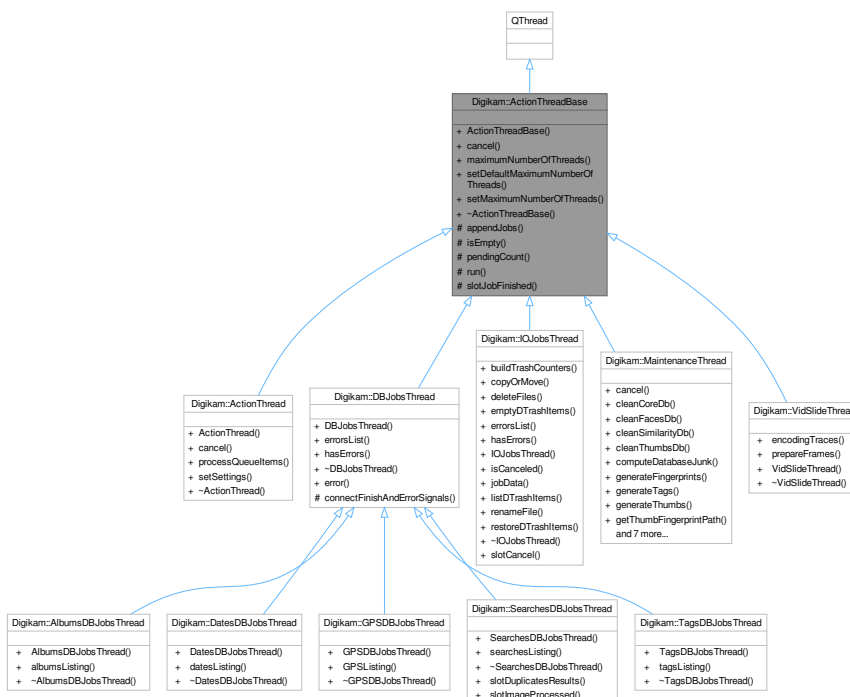
## 6.26.1.4 signalStarting

```
void Digikam::ActionThread::signalStarting (
    const Digikam::ActionData & ad ) [signal]
```

Emit when an item from a queue start to be processed.

## 6.27 Digikam::ActionThreadBase Class Reference

Inheritance diagram for Digikam::ActionThreadBase:



## Public Member Functions

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

## Protected Slots

- void **slotJobFinished** ()

## Protected Member Functions

- void [appendJobs](#) (const [ActionJobCollection](#) &jobs)
- bool [isEmpty](#) () const
- int [pendingCount](#) () const
- void [run](#) () override

## 6.27.1 Member Function Documentation

### 6.27.1.1 [appendJobs\(\)](#)

```
void Digikam::ActionThreadBase::appendJobs (
    const ActionJobCollection & jobs ) [protected]
```

Append a collection of jobs to process into QThreadPool. Jobs are add to pending lists and will be deleted by [ActionThreadBase](#), not QThreadPool.

### 6.27.1.2 [cancel\(\)](#)

```
void Digikam::ActionThreadBase::cancel (
    bool isCancel = true )
```

Cancel processing of current jobs under progress.

### 6.27.1.3 [isEmpty\(\)](#)

```
bool Digikam::ActionThreadBase::isEmpty ( ) const [protected]
```

Return true if list of pending jobs to process is empty.

### 6.27.1.4 [maximumNumberOfThreads\(\)](#)

```
int Digikam::ActionThreadBase::maximumNumberOfThreads ( ) const
```

Return the maximum number of threads used to parallelize collection of job processing.

### 6.27.1.5 [pendingCount\(\)](#)

```
int Digikam::ActionThreadBase::pendingCount ( ) const [protected]
```

Return the number of pending jobs to process.

### 6.27.1.6 [run\(\)](#)

```
void Digikam::ActionThreadBase::run ( ) [override], [protected]
```

Main thread loop used to process jobs in todo list.

### 6.27.1.7 setDefaultMaximumNumberOfThreads()

```
void Digikam::ActionThreadBase::setDefaultMaximumNumberOfThreads ( )
```

Reset maximum number of threads used to parallelize collection of job processing to max core detected on computer. This method is called in constructor.

### 6.27.1.8 setMaximumNumberOfThreads()

```
void Digikam::ActionThreadBase::setMaximumNumberOfThreads (
    int n )
```

Adjust maximum number of threads used to parallelize collection of job processing.

## 6.28 Digikam::ActionVersionsOverlay Class Reference

Inheritance diagram for Digikam::ActionVersionsOverlay:



### Signals

- void **activated** (const [ItemInfo](#) &info)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- **ActionVersionsOverlay** (QObject \*const parent, const QIcon &icon, const QString &text, const QString &tip=QString())
- void **setActive** (bool active) override
- void **setReferenceModel** (const [ItemModel](#) \*model)

## Public Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- [ItemViewHoverButton](#) \* **button** () const
- **HoverButtonDelegateOverlay** (QObject \*const parent)
- void **setActive** (bool active) override

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotClicked** (bool checked)

## Protected Slots inherited from [Digikam::HoverButtonDelegateOverlay](#)

- void **slotEntered** (const QModelIndex &index) override
- void **slotReset** () override

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotEntered** (const QModelIndex &index)
- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

### Protected Member Functions

- Button \* **button** () const
- bool **checkIndex** (const QModelIndex &index) const override
- [ItemViewHoverButton](#) \* **createButton** () override
- void **updateButton** (const QModelIndex &index) override

## Protected Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- QWidget \* **createWidget** () override
- void **visualChange** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **checkIndexOnEnter** (const QModelIndex &index) const
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual void **hide** ()
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* **parentWidget** () const
- virtual void **viewportLeaveEvent** (QObject \*obj, QEvent \*event)
- virtual void **widgetEnterEvent** ()
- void **widgetEnterNotifyMultiple** (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool **affectsMultiple** (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool **viewHasMultiSelection** () const

### Protected Attributes

- QIcon **m\_icon**
- const [ItemModel](#) \* **m\_referenceModel** = nullptr
- QString **m\_text**
- QString **m\_tip**

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **m\_mouseButtonPressedOnWidget** = false
- QWidget \* **m\_widget** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* **m\_delegate** = nullptr
- QAbstractItemView \* **m\_view** = nullptr



## 6.28.1 Member Function Documentation

### 6.28.1.1 checkIndex()

```
bool Digikam::ActionVersionsOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.28.1.2 createButton()

```
ItemViewHoverButton * Digikam::ActionVersionsOverlay::createButton ( ) [override], [protected],
[virtual]
```

Create your widget here. Pass view() as parent.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.28.1.3 setActive()

```
void Digikam::ActionVersionsOverlay::setActive (
    bool active ) [override], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.28.1.4 updateButton()

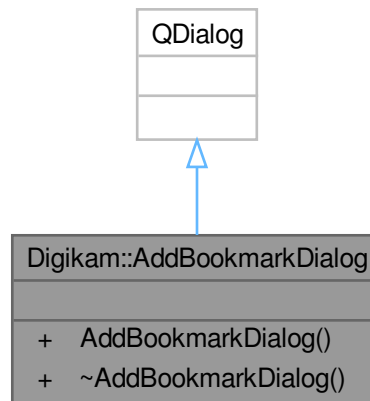
```
void Digikam::ActionVersionsOverlay::updateButton (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implements [Digikam::HoverButtonDelegateOverlay](#).

## 6.29 Digikam::AddBookmarkDialog Class Reference

Inheritance diagram for Digikam::AddBookmarkDialog:

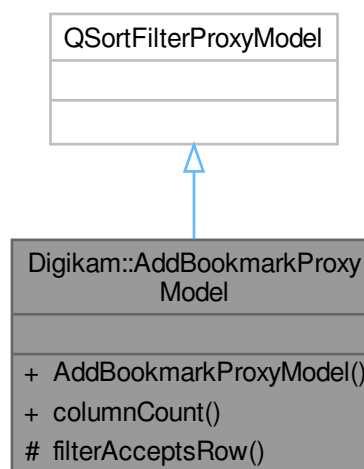


### Public Member Functions

- **AddBookmarkDialog** (const QString &url, const QString &title, QWidget \*const parent=nullptr, [BookmarksManager](#) \*const mngr=nullptr)

## 6.30 Digikam::AddBookmarkProxyModel Class Reference

Inheritance diagram for Digikam::AddBookmarkProxyModel:



### Public Member Functions

- **AddBookmarkProxyModel** (QObject \*const parent=nullptr)
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override

### Protected Member Functions

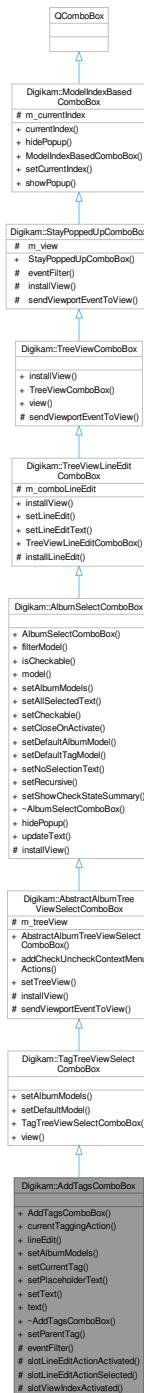
- bool **filterAcceptsRow** (int srow, const QModelIndex &sparent) const override

## 6.30.1 Detailed Description

Proxy model that filters out the bookmarks so only the folders are left behind. Used in the add bookmark dialog combobox.

## 6.31 Digikam::AddTagsComboBox Class Reference

Inheritance diagram for Digikam::AddTagsComboBox:



### Public Slots

- void `setParentTag` (TAlbum \*const album)

## Public Slots inherited from [Digikam::AlbumSelectComboBox](#)

- void **hidePopup** () override
- virtual void **updateText** ()

## Signals

- void **taggingActionActivated** (const [TaggingAction](#) &action)
- void **taggingActionSelected** (const [TaggingAction](#) &action)

## Public Member Functions

- **AddTagsComboBox** (QWidget \*const parent=nullptr)
- [TaggingAction](#) **currentTaggingAction** ()
- [AddTagsLineEdit](#) \* **lineEdit** () const
- void **setAlbumModels** ([TagModel](#) \*const **model**, [TagPropertiesFilterModel](#) \*const **filteredModel**=nullptr, [CheckableAlbumFilterModel](#) \*const **filterModel**=nullptr)
- void **setCurrentTag** ([TAlbum](#) \*const album)
- void **setPlaceholderText** (const QString &message)
- void **setText** (const QString &text)
- QString **text** () const

## Public Member Functions inherited from [Digikam::TagTreeViewSelectComboBox](#)

- void **setAlbumModels** ([TagModel](#) \***model**, [TagPropertiesFilterModel](#) \***filteredModel**=nullptr, [CheckableAlbumFilterModel](#) \***filterModel**=nullptr)
- void **setDefaultModel** ()
- [TagTreeViewSelectComboBox](#) (QWidget \*const parent=nullptr)
- [TagTreeView](#) \* **view** () const

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- [AbstractAlbumTreeViewSelectComboBox](#) (QWidget \*const parent=nullptr)
- void **addCheckUncheckContextMenuActions** ()
- void **setTreeView** ([AbstractAlbumTreeView](#) \*const treeView)

## Public Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- [AlbumSelectComboBox](#) (QWidget \*const parent=nullptr)
- QSortFilterProxyModel \* **filterModel** () const
- bool **isCheckable** () const
- [AbstractCheckableAlbumModel](#) \* **model** () const
- void **setAlbumModels** ([AbstractCheckableAlbumModel](#) \***model**, [AlbumFilterModel](#) \***filterModel**=nullptr)
- void **setAllSelectedText** (bool all)
- void **setCheckable** (bool checkable)
- void **setCloseOnActivate** (bool close)
- void **setDefaultAlbumModel** ()
- void **setDefaultTagModel** ()
- void **setNoSelectionText** (const QString &text)
- void **setRecursive** (bool recursive)
- void **setShowCheckStateSummary** (bool show)

### Public Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- void [installView](#) (QAbstractItemView \*view=nullptr) override
- void [setLineEdit](#) (QLineEdit \*edit)
- void [setLineEditText](#) (const QString &text)
- [TreeViewLineEditComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::TreeViewComboBox](#)

- [TreeViewComboBox](#) (QWidget \*parent=nullptr)
- QTreeView \* [view](#) () const

### Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex [currentIndex](#) () const
- void [hidePopup](#) () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void [setCurrentIndex](#) (const QModelIndex &index)
- void [showPopup](#) () override

### Protected Slots

- void [slotLineEditActionActivated](#) (const [TaggingAction](#) &action)
- void [slotLineEditActionSelected](#) (const [TaggingAction](#) &action)
- void [slotViewIndexActivated](#) (const QModelIndex &)

### Protected Member Functions

- bool [eventFilter](#) (QObject \*object, QEvent \*event) override

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- void [installView](#) (QAbstractItemView \*view=nullptr) override
- void [sendViewportEventToView](#) (QEvent \*e) override

### Protected Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- void [installView](#) (QAbstractItemView \*view=nullptr) override

### Protected Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- virtual void [installLineEdit](#) ()

**Protected Member Functions inherited from [Digikam::TreeViewComboBox](#)**

- void [sendViewportEventToView](#) (QEvent \*e) override

**Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)**

- bool [eventFilter](#) (QObject \*watched, QEvent \*event) override
- void [installView](#) (QAbstractItemView \*view)

**Additional Inherited Members****Protected Attributes inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)**

- [AbstractAlbumTreeView](#) \* [m\\_treeView](#) = nullptr

**Protected Attributes inherited from [Digikam::TreeViewLineEditComboBox](#)**

- QLineEdit \* [m\\_comboLineEdit](#) = nullptr

**Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)**

- QAbstractItemView \* [m\\_view](#) = nullptr

**Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)**

- QPersistentModelIndex [m\\_currentIndex](#)

**6.31.1 Member Function Documentation****6.31.1.1 [currentTaggingAction\(\)](#)**

[TaggingAction](#) Digikam::AddTagsComboBox::currentTaggingAction ( )

Returns the currently set tagging action. This is the last action emitted by either [taggingActionActivated\(\)](#) or [taggingActionSelected\(\)](#)

**6.31.1.2 [setAlbumModels\(\)](#)**

```
void Digikam::AddTagsComboBox::setAlbumModels (
    TagModel *const model,
    TagPropertiesFilterModel *const filteredModel = nullptr,
    CheckableAlbumFilterModel *const filterModel = nullptr )
```

You must call this after construction. If filtered/filterModel is 0, a default one is constructed

### 6.31.1.3 setCurrentTag()

```
void Digikam::AddTagsComboBox::setCurrentTag (
    TAlbum *const album )
```

Sets the currently selected tag

### 6.31.1.4 setParentTag

```
void Digikam::AddTagsComboBox::setParentTag (
    TAlbum *const album ) [slot]
```

Set a parent tag for suggesting a parent tag for a new tag, and a default action.

### 6.31.1.5 taggingActionActivated

```
void Digikam::AddTagsComboBox::taggingActionActivated (
    const TaggingAction & action ) [signal]
```

Emitted when the user activates an action (typically, by pressing return)

### 6.31.1.6 taggingActionSelected

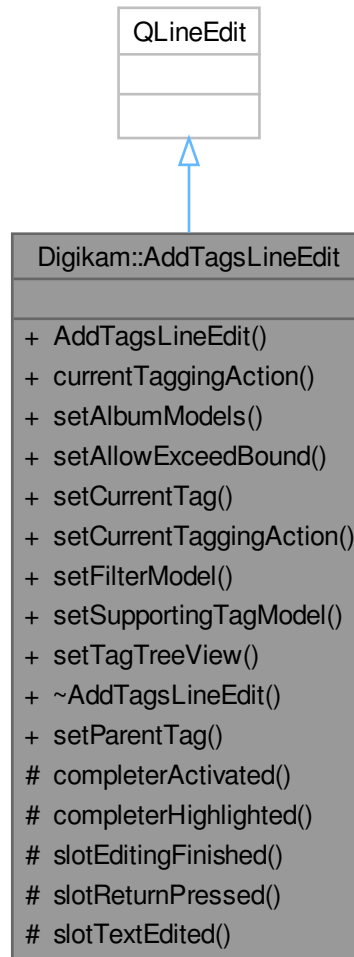
```
void Digikam::AddTagsComboBox::taggingActionSelected (
    const TaggingAction & action ) [signal]
```

Emitted when an action is selected, but not explicitly activated. (typically by selecting an item in the tree view)



## 6.32 Digikam::AddTagsLineEdit Class Reference

Inheritance diagram for Digikam::AddTagsLineEdit:



### Public Slots

- void `setParentTag` (`Album *const album`)

### Signals

- void `taggingActionActivated` (`const TaggingAction &action`)
- void `taggingActionFinished` ()
- void `taggingActionSelected` (`const TaggingAction &action`)

## Public Member Functions

- **AddTagsLineEdit** (QWidget \*const parent=nullptr)
- **TaggingAction currentTaggingAction** () const
- void **setAlbumModels** (TagModel \*const model, TagPropertiesFilterModel \*const filteredModel, AlbumFilterModel \*const filterModel)
- void **setAllowExceedBound** (bool value)
- void **setCurrentTag** (TAlbum \*const tag)
- void **setCurrentTaggingAction** (const TaggingAction &action)
- void **setFilterModel** (AlbumFilterModel \*const model)
- void **setSupportingTagModel** (TagModel \*const model)
- void **setTagTreeView** (TagTreeView \*const treeView)

## Protected Slots

- void **completerActivated** (const TaggingAction &action)
- void **completerHighlighted** (const TaggingAction &action)
- void **slotEditingFinished** ()
- void **slotReturnPressed** ()
- void **slotTextEdited** (const QString &text)

## 6.32.1 Member Function Documentation

### 6.32.1.1 setAlbumModels()

```
void Digikam::AddTagsLineEdit::setAlbumModels (
    TagModel *const model,
    TagPropertiesFilterModel *const filteredModel,
    AlbumFilterModel *const filterModel )
```

Convenience: Will call [setSupportingTagModel\(\)](#) and [setFilterModel\(\)](#)

### 6.32.1.2 setCurrentTag()

```
void Digikam::AddTagsLineEdit::setCurrentTag (
    TAlbum *const tag )
```

Adjusts the current default tagging action to assign the given tag

### 6.32.1.3 setFilterModel()

```
void Digikam::AddTagsLineEdit::setFilterModel (
    AlbumFilterModel *const model )
```

Set a tag filter model. Completion suggestions will be limited to tags contained in the filter model.

#### 6.32.1.4 setParentTag

```
void Digikam::AddTagsLineEdit::setParentTag (
    Album *const album ) [slot]
```

Set a parent tag for suggesting a parent tag for a new tag, and a default action. If you set a tag tree view, this is taken care for automatically.

#### 6.32.1.5 setSupportingTagModel()

```
void Digikam::AddTagsLineEdit::setSupportingTagModel (
    TagModel *const model )
```

Optional: set a model for additional information, like tag icons

#### 6.32.1.6 setTagTreeView()

```
void Digikam::AddTagsLineEdit::setTagTreeView (
    TagTreeView *const treeView )
```

Reads a tag treeview and takes the currently selected tag into account when suggesting a parent tag for a new tag, and a default action.

#### 6.32.1.7 slotReturnPressed

```
void Digikam::AddTagsLineEdit::slotReturnPressed ( ) [protected], [slot]
```

Tagging action is used by facemanagement and assignwidget

#### 6.32.1.8 taggingActionActivated

```
void Digikam::AddTagsLineEdit::taggingActionActivated (
    const TaggingAction & action ) [signal]
```

Emitted when the user activates an action (typically, by pressing return)

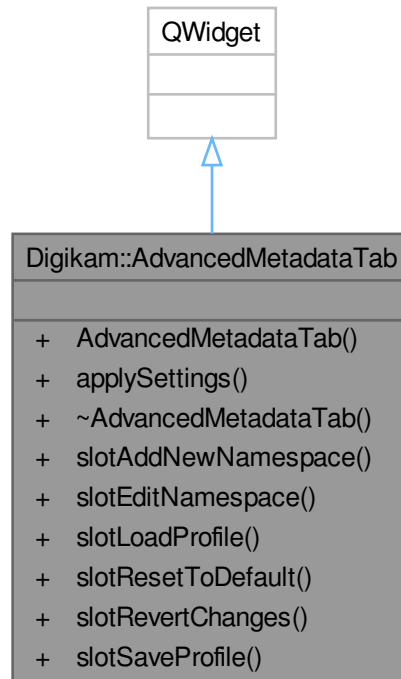
#### 6.32.1.9 taggingActionSelected

```
void Digikam::AddTagsLineEdit::taggingActionSelected (
    const TaggingAction & action ) [signal]
```

Emitted when an action is selected. This already happens if anything is typed.

## 6.33 Digikam::AdvancedMetadataTab Class Reference

Inheritance diagram for Digikam::AdvancedMetadataTab:



### Public Slots

- void **slotAddNewNamespace** ()
- void **slotEditNamespace** ()
- void **slotLoadProfile** ()
- void **slotResetToDefault** ()
- void **slotRevertChanges** ()
- void **slotSaveProfile** ()

### Public Member Functions

- [AdvancedMetadataTab](#) (QWidget \*const parent=nullptr) [explicit]
- void **applySettings** ()

### 6.33.1 Constructor & Destructor Documentation

#### 6.33.1.1 AdvancedMetadataTab()

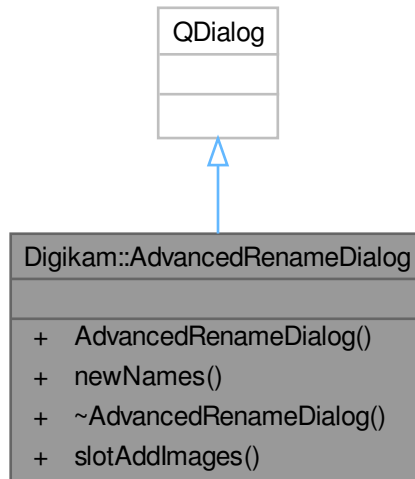
```

Digikam::AdvancedMetadataTab::AdvancedMetadataTab (
    QWidget *const parent = nullptr ) [explicit]
  
```

Connect all actions to `slotRevertAvailable`, which will enable revert to original if an add, edit, delete, or reorder was made

## 6.34 Digikam::AdvancedRenameDialog Class Reference

Inheritance diagram for Digikam::AdvancedRenameDialog:



### Public Slots

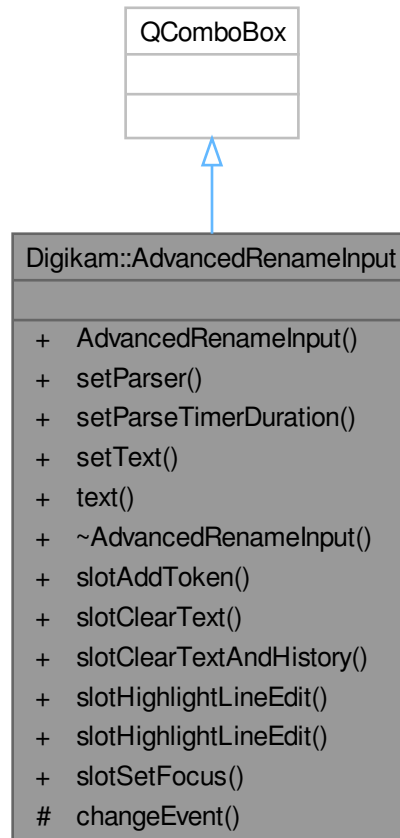
- void **slotAddImages** (const QList< QUrl > &urls)

### Public Member Functions

- **AdvancedRenameDialog** (QWidget \*const parent=nullptr)
- NewNamesList **newNames** () const

## 6.35 Digikam::AdvancedRenameInput Class Reference

Inheritance diagram for Digikam::AdvancedRenameInput:



### Public Slots

- void **slotAddToken** (const QString &)
- void **slotClearText** ()
- void **slotClearTextAndHistory** ()
- void **slotHighlightLineEdit** ()
- void **slotHighlightLineEdit** (const QString &word)
- void **slotSetFocus** ()

### Signals

- void **signalReturnPressed** ()
- void **signalTextChanged** (const QString &)
- void **signalTokenMarked** (bool)

**Public Member Functions**

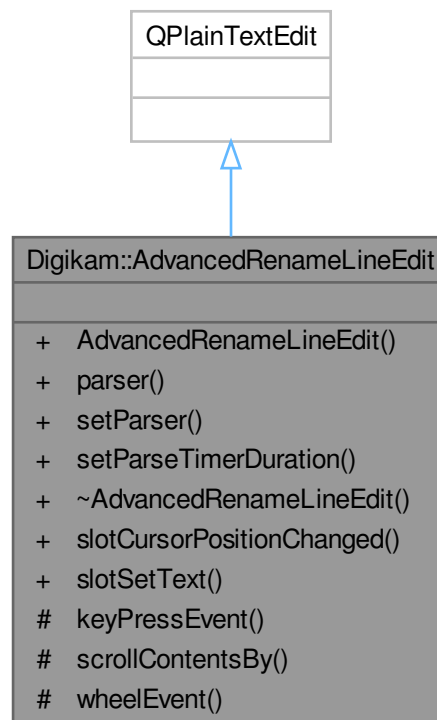
- **AdvancedRenameInput** (QWidget \*const parent=nullptr)
- void **setParser** (Parser \*parser)
- void **setParseTimerDuration** (int milliseconds)
- void **setText** (const QString &text)
- QString **text** () const

**Protected Member Functions**

- void **changeEvent** (QEvent \*e) override

## 6.36 Digikam::AdvancedRenameLineEdit Class Reference

Inheritance diagram for Digikam::AdvancedRenameLineEdit:

**Public Slots**

- void **slotCursorPositionChanged** ()
- void **slotSetText** (const QString &)

## Signals

- void **signalReturnPressed** ()
- void **signalTextChanged** (const QString &)
- void **signalTokenMarked** (bool)

## Public Member Functions

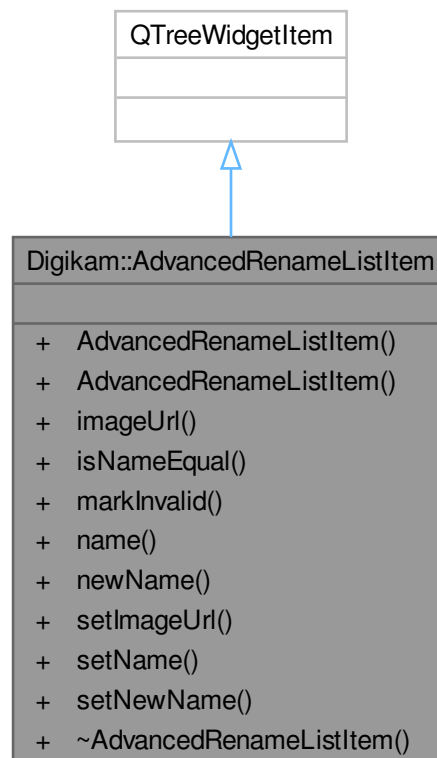
- **AdvancedRenameLineEdit** (QWidget \*const parent=nullptr)
- **Parser** \* **parser** () const
- void **setParser** (**Parser** \*parser)
- void **setParseTimerDuration** (int milliseconds)

## Protected Member Functions

- void **keyPressEvent** (QKeyEvent \*e) override
- void **scrollContentsBy** (int dx, int dy) override
- void **wheelEvent** (QWheelEvent \*e) override

## 6.37 Digikam::AdvancedRenameListItem Class Reference

Inheritance diagram for Digikam::AdvancedRenameListItem:





### Public Types

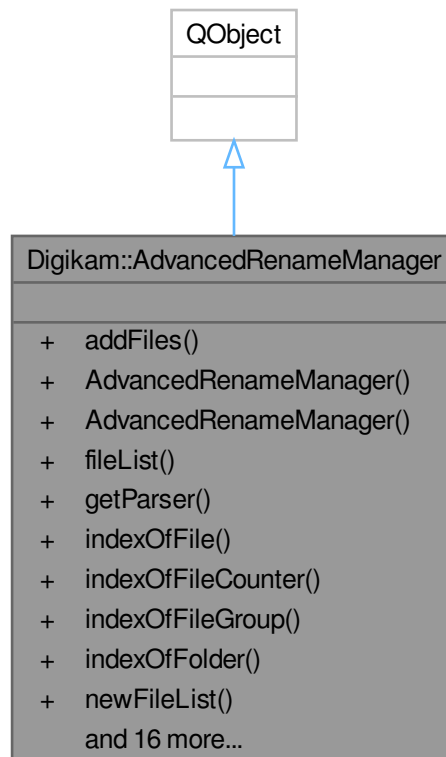
- enum **Column** { **OldName** = 0 , **NewName** }

### Public Member Functions

- **AdvancedRenameListItem** (QTreeWidgetItem \*const view)
- **AdvancedRenameListItem** (QTreeWidgetItem \*const view, const QUrl &info)
- QUrl **imageUrl** () const
- bool **isNameEqual** () const
- void **markInvalid** (bool invalid)
- QString **name** () const
- QString **newName** () const
- void **setImageUrl** (const QUrl &url)
- void **setName** (const QString &name)
- void **setNewName** (const QString &name)

## 6.38 Digikam::AdvancedRenameManager Class Reference

Inheritance diagram for Digikam::AdvancedRenameManager:



## Public Types

- enum **ParserType** { **DefaultParser** = 0 , **ImportParser** }
- enum **SortAction** { **SortName** = 0 , **SortDate** , **SortSize** , **SortCustom** }
- enum **SortDirection** { **SortAscending** = 0 , **SortDescending** }

## Signals

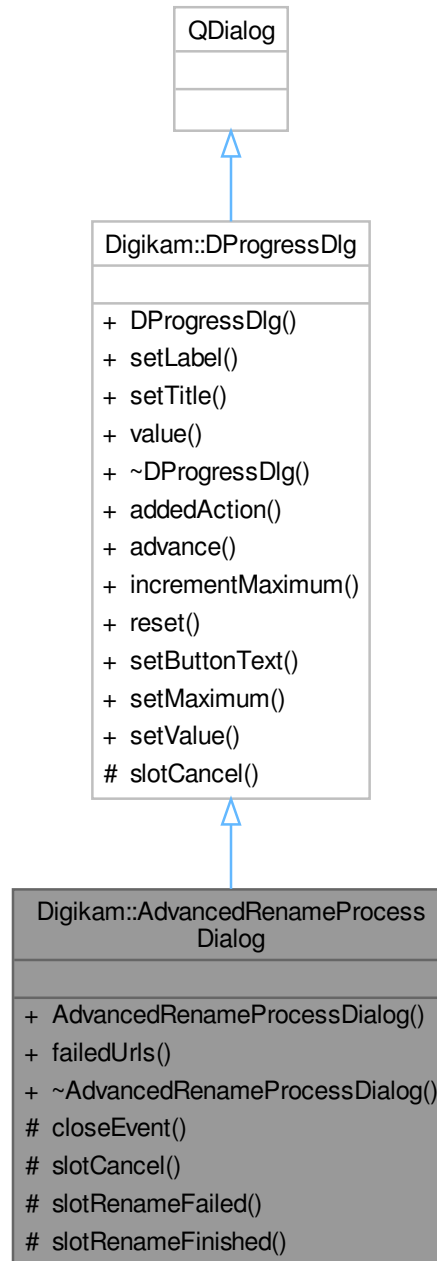
- void **signalSortingChanged** (QList< QUrl >)

## Public Member Functions

- void **addFiles** (const QList< [ParseSettings](#) > &files)
- **AdvancedRenameManager** (const QList< [ParseSettings](#) > &files)
- QStringList **fileList** () const
- [Parser](#) \* **getParser** () const
- int **indexOfFile** (const QString &filename)
- int **indexOfFileCounter** (const QString &filename)
- int **indexOfFileGroup** (const QString &filename)
- int **indexOfFolder** (const QString &filename)
- QMap< QString, QString > **newFileList** (bool checkFileSystem=false) const
- QString **newName** (const QString &filename) const
- void **parseFiles** ()
- void **parseFiles** (const [ParseSettings](#) &settings)
- void **parseFiles** (const QString &parseString)
- void **parseFiles** (const QString &parseString, const [ParseSettings](#) &settings)
- QString **randomStringOfIndex** (int index)
- void **reset** ()
- void **setCutFileName** (int index)
- void **setParserType** (ParserType type)
- void **setSortAction** (SortAction action)
- void **setSortDirection** (SortDirection direction)
- void **setStartIndex** (int index)
- void **setWidget** ([AdvancedRenameWidget](#) \*widget)
- SortAction **sortAction** () const
- SortDirection **sortDirection** () const

## 6.39 Digikam::AdvancedRenameProcessDialog Class Reference

Inheritance diagram for Digikam::AdvancedRenameProcessDialog:



### Public Member Functions

- **AdvancedRenameProcessDialog** (const NewNamesList &list, QWidget \*const parent=nullptr)
- `QList< QUrl > failedUrls ()` const

## Public Member Functions inherited from [Digikam::DProgressDlg](#)

- **DProgressDlg** (QWidget \*const parent=nullptr, const QString &caption=QString())
- void **setLabel** (const QString &text)
- void **setTitle** (const QString &text)
- int **value** () const

## Protected Slots

- void **slotCancel** ()
- void **slotRenameFailed** (const QUrl &url)
- void **slotRenameFinished** ()

## Protected Slots inherited from [Digikam::DProgressDlg](#)

- void **slotCancel** ()

## Protected Member Functions

- void **closeEvent** (QCloseEvent \*e) override

## Additional Inherited Members

## Public Slots inherited from [Digikam::DProgressDlg](#)

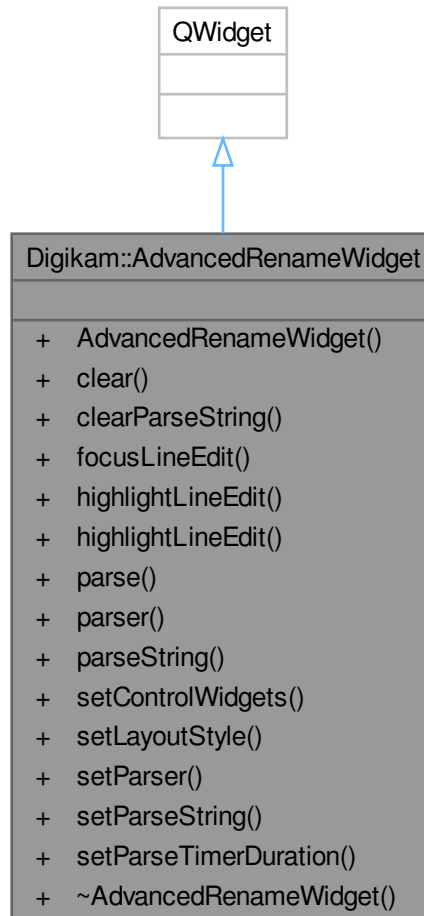
- void **addedAction** (const QPixmap &icon, const QString &text)
- void **advance** (int offset)
- void **incrementMaximum** (int added)
- void **reset** ()
- void **setButtonText** (const QString &text)
- void **setMaximum** (int max)
- void **setValue** (int value)

## Signals inherited from [Digikam::DProgressDlg](#)

- void **signalCancelPressed** ()

## 6.40 Digikam::AdvancedRenameWidget Class Reference

Inheritance diagram for Digikam::AdvancedRenameWidget:



### Public Types

- enum `ControlWidget` { `None` = 0x0 , `ToolTipButton` = 0x1 , `TokenButtons` = 0x2 , `ModifierToolButton` = 0x4 , `DefaultControls` = `TokenButtons` | `ToolTipButton` | `ModifierToolButton` }
- enum `LayoutStyle` { `LayoutNormal` , `LayoutCompact` }

### Signals

- void `signalReturnPressed` ()
- void `signalTextChanged` (const `QString` &)

## Public Member Functions

- **AdvancedRenameWidget** (QWidget \*const parent=nullptr)
- void **clear** ()
- void **clearParseString** ()
- void **focusLineEdit** ()
- void **highlightLineEdit** ()
- void **highlightLineEdit** (const QString &word)
- QString **parse** (ParseSettings &settings) const
- Parser \* **parser** () const
- QString **parseString** () const
- void **setControlWidgets** (ControlWidgets mask)
- void **setLayoutStyle** (LayoutStyle style)
- void **setParser** (Parser \*parser)
- void **setParseString** (const QString &text)
- void **setParseTimerDuration** (int milliseconds)

### 6.40.1 Member Function Documentation

#### 6.40.1.1 clear()

```
void Digikam::AdvancedRenameWidget::clear ( )
```

clears the parse string as well as the history

#### 6.40.1.2 clearParseString()

```
void Digikam::AdvancedRenameWidget::clearParseString ( )
```

resets the current parse string, the LineEdit widget will be empty

#### 6.40.1.3 focusLineEdit()

```
void Digikam::AdvancedRenameWidget::focusLineEdit ( )
```

set focus for the LineEdit widget

#### 6.40.1.4 highlightLineEdit() [1/2]

```
void Digikam::AdvancedRenameWidget::highlightLineEdit ( )
```

highlight the LineEdit widgets text

#### 6.40.1.5 highlightLineEdit() [2/2]

```
void Digikam::AdvancedRenameWidget::highlightLineEdit (
    const QString & word )
```

highlight a word in the LineEdit widgets text

#### 6.40.1.6 parse()

```
QString Digikam::AdvancedRenameWidget::parse (
    ParseSettings & settings ) const
```

evaluates the parse string and executes the parser

**Parameters**

<i>settings</i>	information about the file to be renamed
-----------------	--

**Returns**

the new name of the file

**6.40.1.7 parser()**

```
Parser * Digikam::AdvancedRenameWidget::parser ( ) const
```

returns a pointer to the currently assigned parser

**6.40.1.8 parseString()**

```
QString Digikam::AdvancedRenameWidget::parseString ( ) const
```

returns the current parse string

**6.40.1.9 setControlWidgets()**

```
void Digikam::AdvancedRenameWidget::setControlWidgets (
    ControlWidgets mask )
```

sets the layout of the control widgets

**See also**

ControlWidget

**Parameters**

<i>mask</i>	a bitmask for setting the control widgets
-------------	---

**6.40.1.10 setLayoutStyle()**

```
void Digikam::AdvancedRenameWidget::setLayoutStyle (
    LayoutStyle style )
```

set the layout style of the widget

**Parameters**

<i>style</i>	the style of the layout
--------------	-------------------------

See also

LayoutStyle

#### 6.40.1.11 setParser()

```
void Digikam::AdvancedRenameWidget::setParser (
    Parser * parser )
```

sets the current parser. If a parser has already been assigned, it will be deleted first.

Parameters

<i>parser</i>	a pointer to the new parser instance
---------------	--------------------------------------

#### 6.40.1.12 setParseString()

```
void Digikam::AdvancedRenameWidget::setParseString (
    const QString & text )
```

sets the current parse string

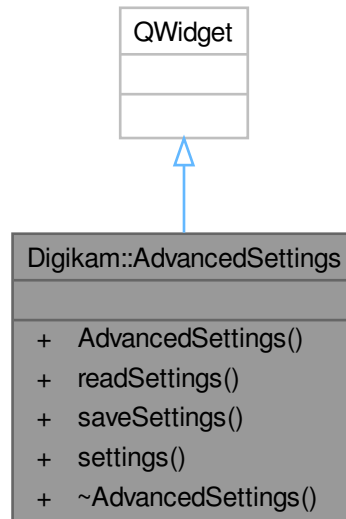
Parameters

<i>text</i>	the new parse string
-------------	----------------------



## 6.41 Digikam::AdvancedSettings Class Reference

Inheritance diagram for Digikam::AdvancedSettings:



### Signals

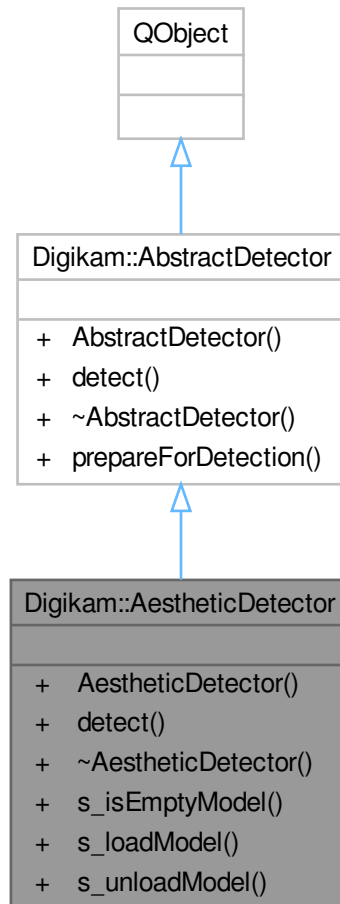
- void **signalDownloadNameChanged** ()

### Public Member Functions

- **AdvancedSettings** (`QWidget *const parent=nullptr`)
- void **readSettings** (`const KConfigGroup &group`)
- void **saveSettings** (`KConfigGroup &group`)
- [DownloadSettings](#) **settings** () const

## 6.42 Digikam::AestheticDetector Class Reference

Inheritance diagram for Digikam::AestheticDetector:



### Public Member Functions

- float `detect` (const cv::Mat &image) const override

### Public Member Functions inherited from [Digikam::AbstractDetector](#)

- `AbstractDetector` (QObject \*const parent=nullptr)

### Static Public Member Functions

- static bool `s_isEmptyModel` ()
- static bool `s_loadModel` ()
- static void `s_unloadModel` ()

## Static Public Member Functions inherited from [Digikam::AbstractDetector](#)

- static cv::Mat [prepareForDetection](#) (const [DImg](#) &inputImage)

### 6.42.1 Member Function Documentation

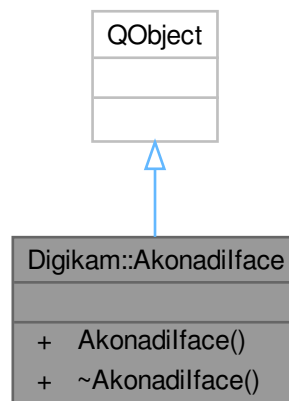
#### 6.42.1.1 detect()

```
float Digikam::AestheticDetector::detect (
    const cv::Mat & image ) const [override], [virtual]
```

Implements [Digikam::AbstractDetector](#).

## 6.43 Digikam::Akonadilface Class Reference

Inheritance diagram for Digikam::Akonadilface:



### Signals

- void **signalContactTriggered** (const `QString` &)

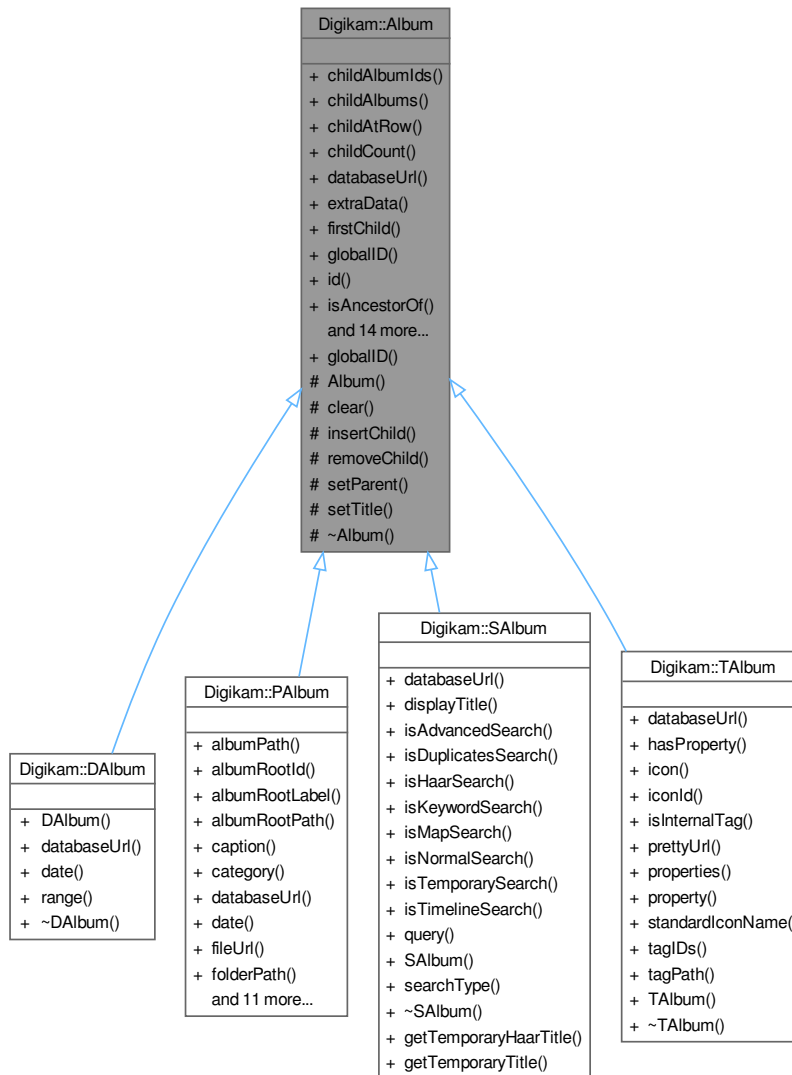
### Public Member Functions

- **Akonadilface** (`QMenu *const parent`)

## 6.44 Digikam::Album Class Reference

Abstract base class for all album types.

Inheritance diagram for Digikam::Album:



### Public Types

- enum [Type](#) {  
[PHYSICAL](#) = 0 , [TAG](#) , [DATE](#) , [SEARCH](#) ,  
[FACE](#) }

### Public Member Functions

- QList< int > [childAlbumIds](#) (bool recursive=false)

- AlbumList [childAlbums](#) (bool recursive=false)
- [Album](#) \* [childAtRow](#) (int row) const
- int [childCount](#) () const
- virtual [CoreDbUrl](#) [databaseUrl](#) () const =0
- void \* [extraData](#) (const void \*const key) const
- [Album](#) \* [firstChild](#) () const
- int [globalID](#) () const
- int [id](#) () const
- bool [isAncestorOf](#) ([Album](#) \*const album) const
- bool [isRoot](#) () const
- bool [isTrashAlbum](#) () const
- bool [isUsedByLabelsTree](#) () const
- [Album](#) \* [lastChild](#) () const
- [Album](#) \* [next](#) () const
- [Album](#) \* [parent](#) () const
- void [prepareForDeletion](#) ()
- [Album](#) \* [prev](#) () const
- void [removeExtraData](#) (const void \*const key)
- int [rowFromAlbum](#) () const
- void [setExtraData](#) (const void \*const key, void \*const value)
- void [setUsedByLabelsTree](#) (bool isUsed)
- QString [title](#) () const
- [Type](#) [type](#) () const

### Static Public Member Functions

- static int [globalID](#) ([Type](#) [type](#), int [id](#))  
*Produces the global id.*

### Protected Member Functions

- [Album](#) ([Album::Type](#) [type](#), int [id](#), bool root)
- void [clear](#) ()
- void [insertChild](#) ([Album](#) \*const child)
- void [removeChild](#) ([Album](#) \*const child)
- void [setParent](#) ([Album](#) \*const [parent](#))
- void [setTitle](#) (const QString &[title](#))
- virtual [~Album](#) ()

### Friends

- class [AlbumManager](#)

## 6.44.1 Detailed Description

A class which provides an abstraction for a type [Album](#). This class is meant to be derived and every time a new [Album](#) Type is defined add a enum corresponding to that to [Album::Type](#)

This class provides a means of building a tree representation for Albums

#### See also

[Album::setParent\(\)](#).

## 6.44.2 Member Enumeration Documentation

### 6.44.2.1 Type

enum `Digikam::Album::Type`

## Enumerator

PHYSICAL	A physical album type.  See also <a href="#">PAlbum</a>
TAG	A tag album type.  See also <a href="#">TAlbum</a>
DATE	A date album type.  See also <a href="#">DAlbum</a>
SEARCH	A search album type.  See also <a href="#">SAlbum</a>
FACE	A faces album type.  See also FAlbum

### 6.44.3 Constructor & Destructor Documentation

#### 6.44.3.1 Album()

```
Digikam::Album::Album (
    Album::Type type,
    int id,
    bool root ) [protected]
```

Constructor

#### 6.44.3.2 ~Album()

```
Digikam::Album::~Album ( ) [protected], [virtual]
```

Destructor

this will also recursively delete all child Albums

### 6.44.4 Member Function Documentation

#### 6.44.4.1 childAlbumIds()

```
QList< int > Digikam::Album::childAlbumIds (
    bool recursive = false )
```

Returns

a list of all child Albums

#### 6.44.4.2 childAlbums()

```
AlbumList Digikam::Album::childAlbums (
    bool recursive = false )
```

##### Returns

a list of all child Albums

#### 6.44.4.3 childAtRow()

```
Album * Digikam::Album::childAtRow (
    int row ) const
```

##### Returns

the child of this album at row

#### 6.44.4.4 childCount()

```
int Digikam::Album::childCount ( ) const
```

##### Returns

the childCount of the album

#### 6.44.4.5 clear()

```
void Digikam::Album::clear ( ) [protected]
```

Delete all child albums and also remove any associated extra data

#### 6.44.4.6 databaseUrl()

```
virtual CoreDbUrl Digikam::Album::databaseUrl ( ) const [pure virtual]
```

##### Returns

the kde url of the album

Implemented in [Digikam::PAlbum](#), [Digikam::TAlbum](#), [Digikam::DAlbum](#), and [Digikam::SAlbum](#).

#### 6.44.4.7 extraData()

```
void * Digikam::Album::extraData (
    const void *const key ) const
```

Retrieve the associated extra data associated with key



## Parameters

key	the key of the extra data
-----	---------------------------

## See also

[setExtraData](#)[extraData](#)**6.44.4.8 firstChild()**

```
Album * Digikam::Album::firstChild ( ) const
```

## Returns

the first child of this album or 0 if no children

**6.44.4.9 globalID() [1/2]**

```
int Digikam::Album::globalID ( ) const
```

An album ID is only unique among the set of all Albums of its Type. This is a global Identifier which will uniquely identifying the [Album](#) among all Albums

## Note

If you are adding a new [Album](#) Type make sure to update this implementation.

You can always get the ID of the album using something like

```
int albumID = rootAlbum->globalID() - album->globalID();
```

## Returns

the globalID of the album

## See also

[id\(\)](#)**6.44.4.10 globalID() [2/2]**

```
int Digikam::Album::globalID (
    Type type,
    int id ) [static]
```

**Parameters**

<i>type</i>	The type of the album
<i>id</i>	the (type-specific) id of the album

**Returns**

the global id

**6.44.4.11 id()**

```
int Digikam::Album::id ( ) const
```

Each album has a ID uniquely identifying it in the set of Albums of a Type

**Note**

The ID for a root [Album](#) is always 0

**Returns**

the ID of the album

**See also**

[globalID\(\)](#)

**6.44.4.12 isAncestorOf()**

```
bool Digikam::Album::isAncestorOf (
    Album *const album ) const
```

**Returns**

true if the `album` is in the parent hierarchy

**Parameters**

<i>album</i>	the album to check whether it belongs in the child hierarchy
--------------	--

**6.44.4.13 isRoot()**

```
bool Digikam::Album::isRoot ( ) const
```

**Returns**

true if the album is a Root [Album](#)

**6.44.4.14 isTrashAlbum()**

```
bool Digikam::Album::isTrashAlbum ( ) const
```

**Returns**

true if the album was created to be a trash virtual album

**6.44.4.15 isUsedByLabelsTree()**

```
bool Digikam::Album::isUsedByLabelsTree ( ) const
```

**Returns**

true if the [Album](#) was created by Labels Tree

**6.44.4.16 lastChild()**

```
Album * Digikam::Album::lastChild ( ) const
```

**Returns**

the last child of this album or 0 if no children

**6.44.4.17 next()**

```
Album * Digikam::Album::next ( ) const
```

**Returns**

the next sibling of this album of this album or 0 if no next sibling

**See also**

[AlbumIterator](#)

**6.44.4.18 parent()**

```
Album * Digikam::Album::parent ( ) const
```

**Returns**

the parent album for this album

#### 6.44.4.19 prepareForDeletion()

```
void Digikam::Album::prepareForDeletion ( )
```

For secure deletion in an album model, call this function beforehand

#### 6.44.4.20 prev()

```
Album * Digikam::Album::prev ( ) const
```

##### Returns

the previous sibling of this album of this album or 0 if no previous sibling

##### See also

[AlbumIterator](#)

#### 6.44.4.21 removeExtraData()

```
void Digikam::Album::removeExtraData (
    const void *const key )
```

Remove the associated extra data associated with `key`

##### Parameters

<code>key</code>	the key of the extra data
------------------	---------------------------

##### See also

[setExtraData](#)

[extraData](#)

#### 6.44.4.22 rowFromAlbum()

```
int Digikam::Album::rowFromAlbum ( ) const
```

##### Returns

the `rowFromAlbum` of the album

**6.44.4.23 setExtraData()**

```
void Digikam::Album::setExtraData (
    const void *const key,
    void *const value )
```

This allows to associate some "extra" data to a [Album](#). As one [Album](#) can be used by several objects (often views) which all need to add some data, you have to use a key to reference your extra data within the [Album](#).

That way a [Album](#) can hold and provide access to all those views separately.

for eg,

```
album->setExtraData( this, searchFolderItem );
```

and can later access the searchFolderItem by doing

```
SearchFolderItem *item = static_cast<SearchFolderItem*>(album->extraData(this));
```

Note: you have to remove and destroy the data you associated yourself when you don't need it anymore!

**Parameters**

<i>key</i>	the key of the extra data
<i>value</i>	the value of the extra data

**See also**

[extraData](#)

[removeExtraData](#)

**6.44.4.24 setUsedByLabelsTree()**

```
void Digikam::Album::setUsedByLabelsTree (
    bool isUsed )
```

Sets the property `m_usedByLabelsTree` to true if the search album was created using the Colors and labels tree view

**Parameters**

<i>isUsed</i>	=> the status of the usage
---------------	----------------------------

**6.44.4.25 title()**

```
QString Digikam::Album::title ( ) const
```

**Returns**

the `title` aka name of the album

#### 6.44.4.26 type()

`Album::Type Digikam::Album::type ( ) const`

##### Returns

the type of album

##### See also

[Type](#)

## 6.45 Digikam::AlbumChangeset Class Reference

### Public Types

- enum **Operation** { **Unknown** , **Added** , **Deleted** , **Renamed** , **PropertiesChanged** }

### Public Member Functions

- **AlbumChangeset** (int albumId, Operation operation)
- int **albumId** () const
- Operation **operation** () const

## 6.46 Digikam::AlbumCopyMoveHint Class Reference

### Public Member Functions

- [AlbumCopyMoveHint](#) ()=default
- **AlbumCopyMoveHint** (int srcAlbumRootId, int srcAlbum, int dstAlbumRootId, const QString &dstRelativePath)
- int **albumIdSrc** () const
- int **albumRootIdDst** () const
- int **albumRootIdSrc** () const
- CollectionScannerHints::DstPath **dst** () const
- bool **isDstAlbum** (int albumRootId, const QString &relativePath) const
- bool **isSrcAlbum** (int albumRootId, int albumId) const
- **operator const CollectionScannerHints::Album &** () const
- **operator const CollectionScannerHints::DstPath &** () const
- bool **operator==** (const CollectionScannerHints::Album &src) const
- bool **operator==** (const CollectionScannerHints::DstPath &dst) const
- QT\_HASH\_TYPE **qHash** () const
- QString **relativePathDst** () const
- CollectionScannerHints::Album **src** () const

### Protected Attributes

- CollectionScannerHints::DstPath **m\_dst**
- CollectionScannerHints::Album **m\_src**

## 6.46.1 Constructor & Destructor Documentation

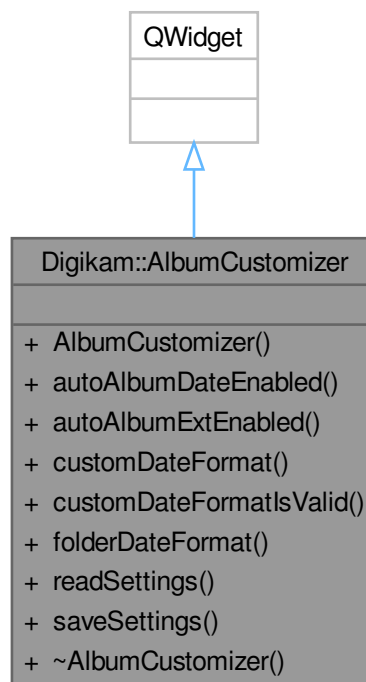
### 6.46.1.1 AlbumCopyMoveHint()

```
Digikam::AlbumCopyMoveHint::AlbumCopyMoveHint ( ) [default]
```

An [AlbumCopyMoveHint](#) describes an existing album and a destination to which this album is expected to be copied, moved or renamed.

## 6.47 Digikam::AlbumCustomizer Class Reference

Inheritance diagram for Digikam::AlbumCustomizer:



### Public Types

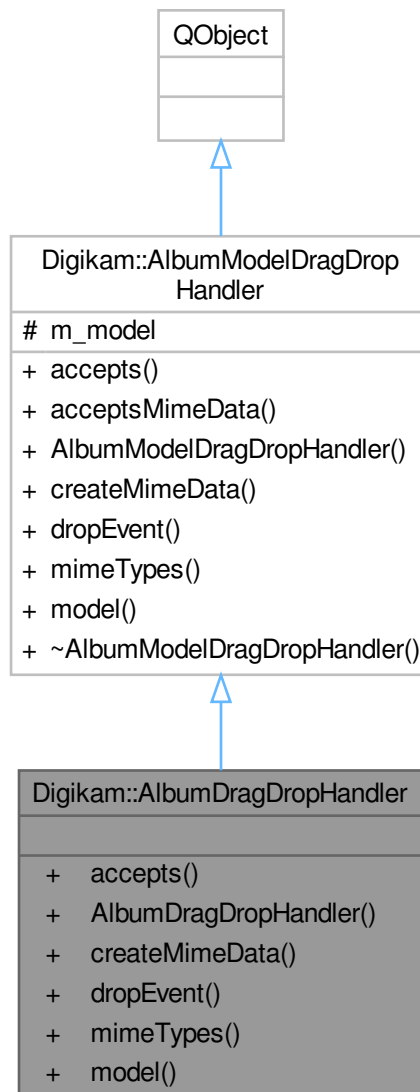
- enum **DateFormatOptions** { **IsoDateFormat** = 0 , **TextDateFormat** , **LocalDateFormat** , **CustomDate↵  
Format** }

### Public Member Functions

- **AlbumCustomizer** (QWidget \*const parent=nullptr)
- bool **autoAlbumDateEnabled** () const
- bool **autoAlbumExtEnabled** () const
- QString **customDateFormat** () const
- bool **customDateFormatsValid** () const
- int **folderDateFormat** () const
- void **readSettings** (const KConfigGroup &group)
- void **saveSettings** (KConfigGroup &group)

## 6.48 Digikam::AlbumDragDropHandler Class Reference

Inheritance diagram for Digikam::AlbumDragDropHandler:





## Public Member Functions

- Qt::DropAction [accepts](#) (const QDropEvent \*e, const QModelIndex &dropIndex) override
- **AlbumDragDropHandler** (AlbumModel \*const model)
- QMimeData \* [createMimeData](#) (const QList< Album \* > &) override
- bool [dropEvent](#) (QAbstractItemView \*view, const QDropEvent \*e, const QModelIndex &droppedOn) override
- QStringList [mimeTypes](#) () const override
- AlbumModel \* [model](#) () const

## Public Member Functions inherited from [Digikam::AlbumModelDragDropHandler](#)

- virtual bool [acceptsMimeData](#) (const QMimeData \*data)
- **AlbumModelDragDropHandler** (AbstractAlbumModel \*model)
- [AbstractAlbumModel](#) \* [model](#) () const

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::AlbumModelDragDropHandler](#)

- [AbstractAlbumModel](#) \* [m\\_model](#) = nullptr

## 6.48.1 Member Function Documentation

### 6.48.1.1 accepts()

```
Qt::DropAction Digikam::AlbumDragDropHandler::accepts (
    const QDropEvent * e,
    const QModelIndex & dropIndex ) [override], [virtual]
```

Returns if the given mime data is accepted for drop on dropIndex. Returns the proposed action, or Qt::IgnoreAction if not accepted.

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

### 6.48.1.2 createMimeData()

```
QMimeData * Digikam::AlbumDragDropHandler::createMimeData (
    const QList< Album * > & ) [override], [virtual]
```

Create a mime data object for starting a drag from the given Albums

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

### 6.48.1.3 dropEvent()

```
bool Digikam::AlbumDragDropHandler::dropEvent (
    QAbstractItemView * view,
    const QDropEvent * e,
    const QModelIndex & droppedOn ) [override], [virtual]
```

Gives the view and the occurring drop event. The index is the index where the drop was dropped on. It may be invalid (dropped on decoration, viewport) Returns true if the event is to be accepted.

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

#### 6.48.1.4 mimeTypees()

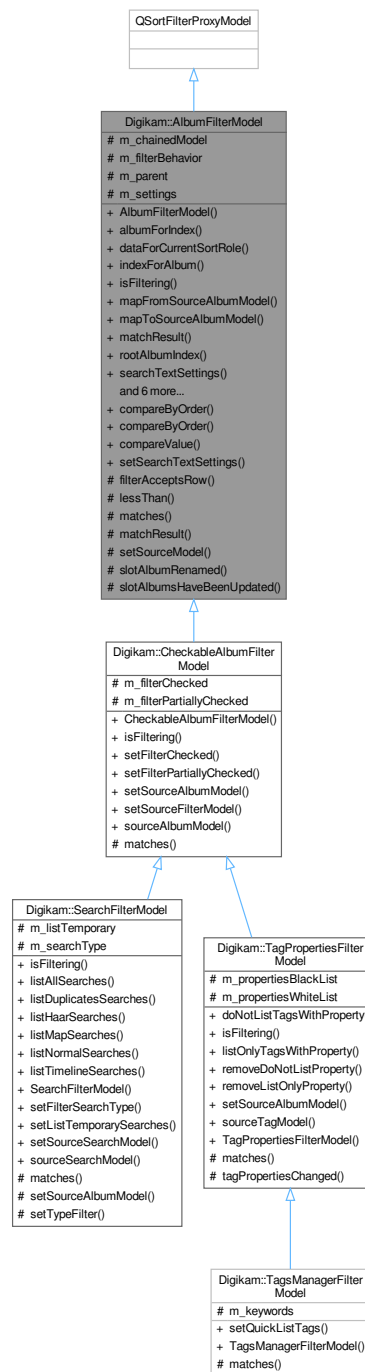
```
QStringList Digikam::AlbumDragDropHandler::mimeTypees ( ) const [override], [virtual]
```

Returns the supported mime types. Called by the default implementation of model's [mimeTypees\(\)](#).

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

## 6.49 Digikam::AlbumFilterModel Class Reference

Inheritance diagram for Digikam::AlbumFilterModel:



### Public Types

- enum [FilterBehavior](#) { [SimpleFiltering](#) , [FullFiltering](#) , [StrictFiltering](#) }
- enum [MatchResult](#) { [NoMatch](#) = 0 , [DirectMatch](#) , [ParentMatch](#) , [ChildMatch](#) , [SpecialMatch](#) }

## Public Slots

- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)

## Signals

- void [hasSearchResult](#) (bool hasResult)
- void [searchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [searchTextSettingsChanged](#) (bool wasSearching, bool searched)
- void [signalFilterChanged](#) ()

## Public Member Functions

- **AlbumFilterModel** (QObject \*const parent=nullptr)
- **Album \* albumForIndex** (const QModelIndex &index) const  
*Convenience methods.*
- QVariant **dataForCurrentSortRole** ([Album](#) \*album) const
- QModelIndex **indexForAlbum** ([Album](#) \*album) const
- virtual bool **isFiltering** () const
- QModelIndex **mapFromSourceAlbumModel** (const QModelIndex &index) const
- QModelIndex **mapToSourceAlbumModel** (const QModelIndex &index) const
- [MatchResult](#) **matchResult** (const QModelIndex &index) const
- QModelIndex **rootAlbumIndex** () const
- [SearchTextSettings](#) **searchTextSettings** () const
- void **setFilterBehavior** ([FilterBehavior](#) behavior)
- void **setSourceAlbumModel** ([AbstractAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([AlbumFilterModel](#) \*const source)
- [AbstractAlbumModel](#) \* **sourceAlbumModel** () const
- [AlbumFilterModel](#) \* **sourceFilterModel** () const
- void **updateFilter** ()

## Static Public Member Functions

- template<typename T >  
static int **compareByOrder** (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int **compareByOrder** (int compareResult, Qt::SortOrder sortOrder)
- template<typename T >  
static int **compareValue** (const T &a, const T &b)

## Protected Slots

- void **slotAlbumRenamed** ([Album](#) \*album)
- void **slotAlbumsHaveBeenUpdated** (int type)

## Protected Member Functions

- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool **matches** ([Album](#) \*album) const
- [MatchResult](#) **matchResult** ([Album](#) \*album) const
- void **setSourceModel** (QAbstractItemModel \*const model) override

## Protected Attributes

- `QPointer< AlbumFilterModel > m_chainedModel = nullptr`
- `FilterBehavior m_filterBehavior = FullFiltering`
- `QObject * m_parent = nullptr`
- `SearchTextSettings m_settings`

## 6.49.1 Member Enumeration Documentation

### 6.49.1.1 FilterBehavior

```
enum Digikam::AlbumFilterModel::FilterBehavior
```

#### Enumerator

SimpleFiltering	If an index does not match, the index and all its children are filtered out. This is the Qt default behavior, but undesirable for album trees.
FullFiltering	Default behavior. If an index matches, it is shown, which directly means all its parents are shown as well. In addition, all its children are shown as well.
StrictFiltering	If an index matches, it is shown, which directly means all its parents are shown as well. Its children are not shown unless they also match.

### 6.49.1.2 MatchResult

```
enum Digikam::AlbumFilterModel::MatchResult
```

#### Enumerator

NoMatch	This enum can be used as a boolean value if match/no match only is needed.
DirectMatch	The index itself is matched.
ParentMatch	A parent if the index is matched.
ChildMatch	A child of the index is matched.
SpecialMatch	The index is matched not because of search settings, but because it has a special type.

## 6.49.2 Member Function Documentation

### 6.49.2.1 compareByOrder()

```
static int Digikam::AlbumFilterModel::compareByOrder (
    int compareResult,
    Qt::SortOrder sortOrder ) [inline], [static]
```

Takes a typical result from a compare method (0 is equal, -1 is less than, 1 is greater than) and applies the given sort order to it.

### 6.49.2.2 compareValue()

```
template<typename T >
static int Digikam::AlbumFilterModel::compareValue (
    const T & a,
    const T & b ) [inline], [static]
```

Returns the usual compare result of -1, 0, or 1 for lessThan, equals and greaterThan.

### 6.49.2.3 hasSearchResult

```
void Digikam::AlbumFilterModel::hasSearchResult (
    bool hasResult ) [signal]
```

Indicates whether the newly applied filter results in a search result or not.

#### Parameters

<i>hasResult</i>	true if the new filter matches any album, else false
------------------	--

### 6.49.2.4 isFiltering()

```
bool Digikam::AlbumFilterModel::isFiltering ( ) const [virtual]
```

Returns if the currently applied filters will result in any filtering.

#### Returns

true if the current selected filter could result in any filtering without checking if this really happens.

Reimplemented in [Digikam::CheckableAlbumFilterModel](#), [Digikam::SearchFilterModel](#), and [Digikam::TagPropertiesFilterModel](#).

### 6.49.2.5 lessThan()

```
bool Digikam::AlbumFilterModel::lessThan (
    const QModelIndex & left,
    const QModelIndex & right ) const [override], [protected]
```

Implementation to sort Tags that contain Unconfirmed Faces, according to the Unconfirmed Face Count.

### 6.49.2.6 matches()

```
bool Digikam::AlbumFilterModel::matches (
    Album * album ) const [protected], [virtual]
```

This method provides the basic match checking algorithm. Return true if this single album matches the current criteria. This method can be overridden to provide custom filtering.

## Parameters

<i>album</i>	the album to tell if it matches the filter criteria or not.
--------------	---

Reimplemented in [Digikam::CheckableAlbumFilterModel](#), [Digikam::SearchFilterModel](#), [Digikam::TagPropertiesFilterModel](#), and [Digikam::TagsManagerFilterModel](#).

**6.49.2.7 matchResult()** [1/2]

```
AlbumFilterModel::MatchResult Digikam::AlbumFilterModel::matchResult (
    Album * album ) const [protected]
```

Returns if the filter matches this album (same logic as `filterAcceptsRow`). An album matches if the search text settings are found in a parent album's title, in the album's title or in a child album's title, or if it is a special album (root) that is never filtered out.

**6.49.2.8 matchResult()** [2/2]

```
AlbumFilterModel::MatchResult Digikam::AlbumFilterModel::matchResult (
    const QModelIndex & index ) const
```

Returns the `MatchResult` of an index of this model. Never returns `NoMatch` for a valid index, because in this case, the index would rather be filtered out.

**6.49.2.9 searchTextSettings()**

```
SearchTextSettings Digikam::AlbumFilterModel::searchTextSettings ( ) const
```

Returns the settings currently used for filtering.

## Returns

current settings for filtering.

**6.49.2.10 searchTextSettingsAboutToChange**

```
void Digikam::AlbumFilterModel::searchTextSettingsAboutToChange (
    bool searched,
    bool willSearch ) [signal]
```

This signal indicates that a new [SearchTextSettings](#) arrived and is about to be applied to the model.

## Parameters

<i>searched</i>	true if filtering by text was enabled before applying the new settings
<i>willSearch</i>	true if the new settings can result in any filtering by text, else false.

**6.49.2.11 searchTextSettingsChanged**

```
void Digikam::AlbumFilterModel::searchTextSettingsChanged (
    bool wasSearching,
    bool searched ) [signal]
```

Indicates that new search text settings were applied.

**Parameters**

<i>wasSearching</i>	true if this is not a new search that
<i>searched</i>	true if the new settings result in any filtering

**6.49.2.12 setFilterBehavior()**

```
void Digikam::AlbumFilterModel::setFilterBehavior (
    FilterBehavior behavior )
```

Sets the filter behavior. Default is FullFiltering.

**6.49.2.13 setSearchTextSettings**

```
void Digikam::AlbumFilterModel::setSearchTextSettings (
    const SearchTextSettings & settings ) [slot]
```

Accepts new settings used for filtering and applies them to the model.

**Parameters**

<i>settings</i>	new settings to apply. An empty text will be interpreted as no filtering
-----------------	--

**6.49.2.14 setSourceAlbumModel()**

```
void Digikam::AlbumFilterModel::setSourceAlbumModel (
    AbstractAlbumModel *const source )
```

Sets the source model. Note: If a chained filter model is set, it will not be reset, but the source album model will be made source of the chained filter model.

**6.49.2.15 setSourceFilterModel()**

```
void Digikam::AlbumFilterModel::setSourceFilterModel (
    AlbumFilterModel *const source )
```

Sets a chained filter model. Note: If a direct source album model is set as current source, it will be set as source AlbumModel of the new source filter model.



### 6.49.2.16 setSourceModel()

```
void Digikam::AlbumFilterModel::setSourceModel (
    QAbstractItemModel *const model ) [override], [protected]
```

Use setSourceAlbumModel.

See also

[setSourceAlbumModel](#)

Parameters

<i>model</i>	source model
--------------	--------------

### 6.49.2.17 signalFilterChanged

```
void Digikam::AlbumFilterModel::signalFilterChanged ( ) [signal]
```

Indicates that a new filter was applied to the model.

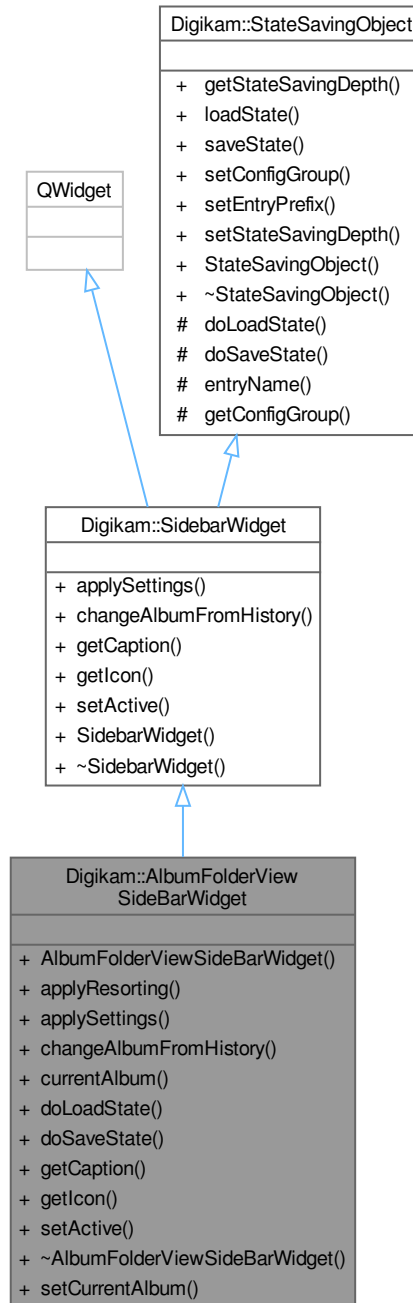
### 6.49.2.18 updateFilter()

```
void Digikam::AlbumFilterModel::updateFilter ( )
```

Force invalidateFilter() externally.

## 6.50 Digikam::AlbumFolderViewSideBarWidget Class Reference

Inheritance diagram for Digikam::AlbumFolderViewSideBarWidget:



### Public Slots

- void **setCurrentAlbum** ([PAlbum](#) \*album)

## Signals

- void **signalFindDuplicates** (const QList< [PAlbum](#) \* > &albums)

## Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

## Public Member Functions

- **AlbumFolderViewSideBarWidget** (QWidget \*const parent, [AlbumModel](#) \*const model, [AlbumModificationHelper](#) \*const albumModificationHelper)
- void **applyResorting** ()
- void [applySettings](#) () override
- void [changeAlbumFromHistory](#) (const QList< [Album](#) \* > &album) override
- [AlbumPointer](#)< [PAlbum](#) > **currentAlbum** () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- void [setActive](#) (bool active) override

## Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.50.1 Member Function Documentation

### 6.50.1.1 applySettings()

```
void Digikam::AlbumFolderViewSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).

### 6.50.1.2 changeAlbumFromHistory()

```
void Digikam::AlbumFolderViewSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.50.1.3 doLoadState()

```
void Digikam::AlbumFolderViewSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.50.1.4 doSaveState()

```
void Digikam::AlbumFolderViewSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.50.1.5 getCaption()

```
const QString Digikam::AlbumFolderViewSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.50.1.6 `getIcon()`

```
const QIcon Digikam::AlbumFolderViewSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.50.1.7 `setActive()`

```
void Digikam::AlbumFolderViewSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.51 Digikam::AlbumHistory Class Reference

Inheritance diagram for Digikam::AlbumHistory:



## Public Slots

- void **slotAlbumCurrentChanged** ()
- void **slotAlbumDeleted** ([Album](#) \*album)
- void **slotAlbumsCleared** ()
- void **slotAlbumSelected** ()
- void **slotClearSelectPAlbum** (const [ItemInfo](#) &imageInfo)
- void **slotClearSelectTAlbum** (int id)
- void **slotCurrentChange** (const [ItemInfo](#) &info)
- void **slotImageSelected** (const [ItemInfoList](#) &selectedImage)

## Signals

- void **signalSetCurrent** (qulonglong imageId)
- void **signalSetSelectedInfos** (const [QList](#)< [ItemInfo](#) > &)

## Public Member Functions

- void **addAlbums** (const [QList](#)< [Album](#) \* > &albums, [QWidget](#) \*const widget, const [QHash](#)< [LabelsTreeView::Labels](#), [QList](#)< int > > &selectedLabels)
 

*AlbumHistory::addAlbums A special overloaded function for handling [AlbumHistory](#) for the Labels tree-view.*
- void **addAlbums** (const [QList](#)< [Album](#) \* > &albums, [QWidget](#) \*const widget=nullptr)
- **AlbumHistory** ([QObject](#) \*const parent=nullptr)
- void **back** ([QList](#)< [Album](#) \* > &album, [QWidget](#) \*\*const widget, unsigned int steps=1)
- void **clearHistory** ()
- void **deleteAlbum** ([Album](#) \*const album)
- void **forward** ([QList](#)< [Album](#) \* > &album, [QWidget](#) \*\*const widget, unsigned int steps=1)
- void **getBackwardHistory** ([QStringList](#) &list) const
- void **getCurrentAlbum** ([Album](#) \*\*const album, [QWidget](#) \*\*const widget) const
- void **getForwardHistory** ([QStringList](#) &list) const
- bool **isBackwardEmpty** () const
- bool **isForwardEmpty** () const
- [QHash](#)< [LabelsTreeView::Labels](#), [QList](#)< int > > **neededLabels** ()

### 6.51.1 Detailed Description

Manages the history of the last visited albums.

The user is able to navigate through the albums, he has opened during a session.

### 6.51.2 Member Function Documentation

#### 6.51.2.1 addAlbums()

```
void Digikam::AlbumHistory::addAlbums (
    const QList< Album * > & albums,
    QWidget *const widget,
    const QHash< LabelsTreeView::Labels, QList< int > > & selectedLabels )
```

#### Author

Mohamed\_Anwer

## 6.52 Digikam::AlbumInfo Class Reference

### Public Types

- typedef QList< [AlbumInfo](#) > **List**

### Public Member Functions

- bool **isNull** () const
- bool **operator**< (const [AlbumInfo](#) &info) const

### Public Attributes

- int **albumRootId** = 0
- QString **caption**
- QString **category**
- QDate **date**
- qlonglong **iconId** = 0
- int **id** = 0
- QString **relativePath**

### 6.52.1 Detailed Description

A container class for transporting album information from the database to [AlbumManager](#)

### 6.52.2 Member Function Documentation

#### 6.52.2.1 operator<()

```
bool Digikam::AlbumInfo::operator< (  
    const AlbumInfo & info ) const [inline]
```

needed for sorting

## 6.53 Digikam::AlbumIterator Class Reference

### Public Member Functions

- **AlbumIterator** ([Album](#) \*const album)
- [Album](#) \* **current** () const
- [Album](#) \* **operator**\* ()
- [AlbumIterator](#) & **operator**++ ()



### 6.53.1 Detailed Description

Iterate over all children of this [Album](#).

#### Note

It will not include the specified album

#### Example usage:

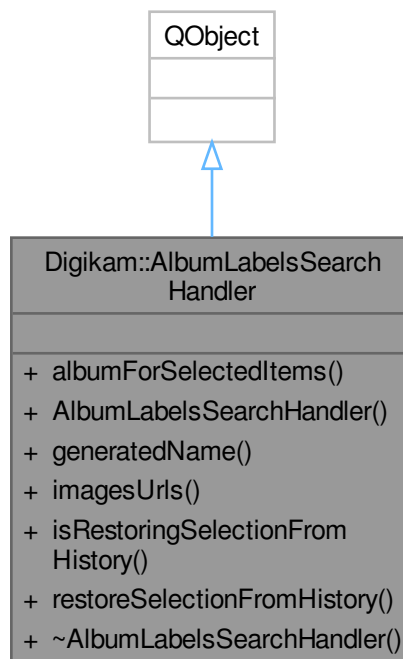
```
AlbumIterator it(album);
while ( it.current() )
{
    qDebug(DIGIKAM_GENERAL_LOG) << "Album: " << it.current()->title();
    ++it;
}
```

#### Warning

Do not delete albums using this iterator.

## 6.54 Digikam::AlbumLabelsSearchHandler Class Reference

Inheritance diagram for Digikam::AlbumLabelsSearchHandler:



#### Signals

- void **checkStateChanged** ([Album](#) \*album, Qt::CheckState checkState)

## Public Member Functions

- `Album * albumForSelectedItems () const`
- `AlbumLabelsSearchHandler (LabelsTreeView *const treeWidget)`
- `QString generatedName () const`
- `QList< QUrl > imagesUrls () const`  
*Gets the list of images generated, for exporting.*
- `bool isRestoringSelectionFromHistory () const`
- `void restoreSelectionFromHistory (const QHash< LabelsTreeView::Labels, QList< int > > &neededLabels)`  
*Restores the selection of the tree-view from history.*

## 6.54.1 Member Function Documentation

### 6.54.1.1 albumForSelectedItems()

```
Album * Digikam::AlbumLabelsSearchHandler::albumForSelectedItems ( ) const
```

#### Returns

`Album` pointer of the currently selected labels

### 6.54.1.2 generatedName()

```
QString Digikam::AlbumLabelsSearchHandler::generatedName ( ) const
```

#### Returns

A string for a name generated by

#### See also

`generateAlbumNameForExporting()`

### 6.54.1.3 imagesUrls()

```
QList< QUrl > Digikam::AlbumLabelsSearchHandler::imagesUrls ( ) const
```

#### Returns

`QUrl` List of images Urls

### 6.54.1.4 isRestoringSelectionFromHistory()

```
bool Digikam::AlbumLabelsSearchHandler::isRestoringSelectionFromHistory ( ) const
```

#### Returns

true if the tree-view is restoring the selection state from history to block searching until the restoring is done

### 6.54.1.5 restoreSelectionFromHistory()

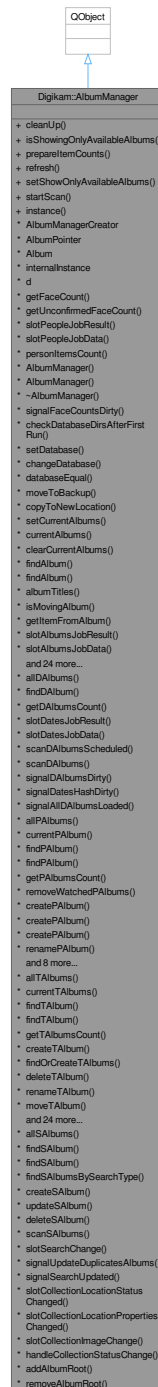
```
void Digikam::AlbumLabelsSearchHandler::restoreSelectionFromHistory (
    const QHash< LabelsTreeView::Labels, QList< int > > & neededLabels )
```

## Parameters

<i>neededLabels</i>	a hash to restore selection from it
---------------------	-------------------------------------

## 6.55 Digikam::AlbumManager Class Reference

Inheritance diagram for Digikam::AlbumManager:



## Classes

- class [Private](#)

## Public Member Functions

- void [cleanUp](#) ()
- bool [isShowingOnlyAvailableAlbums](#) () const
- void [prepareItemCounts](#) ()
- void [refresh](#) ()
- void [setShowOnlyAvailableAlbums](#) (bool onlyAvailable)
- void [startScan](#) ()

## Static Public Member Functions

- static [AlbumManager](#) \* [instance](#) ()

## Operations on Face Album

- class [AlbumManagerCreator](#)
- template<class T >  
class [AlbumPointer](#)
- class [Album](#)
- QHash< int, int > [getFaceCount](#) () const
- QHash< int, int > [getUnconfirmedFaceCount](#) () const
- void [signalFaceCountsDirty](#) (const QHash< int, int > &faceCount, const QHash< int, int > &uFaceCount, const QList< int > &toUpdatedFaces)

## Operations with database

- static void [checkDatabaseDirsAfterFirstRun](#) (const QString &dbPath, const QString &albumPath)
- bool [setDatabase](#) (const [DbEngineParameters](#) &params, bool priority, const QString &suggestedAlbumRoot=QString(), bool ignoreDisappearedLocations=false)
- void [changeDatabase](#) (const [DbEngineParameters](#) &params)
- bool [databaseEqual](#) (const [DbEngineParameters](#) &parameters) const

## Operations on generic Album

- void [setCurrentAlbums](#) (const QList< [Album](#) \* > &albums)
- AlbumList [currentAlbums](#) () const
- void [clearCurrentAlbums](#) ()
- [Album](#) \* [findAlbum](#) (int gid) const
- [Album](#) \* [findAlbum](#) ([Album](#)::Type type, int id) const
- QHash< int, QString > [albumTitles](#) () const
- bool [isMovingAlbum](#) ([Album](#) \*album) const
- qlonglong [getItemFromAlbum](#) ([Album](#) \*const album, const QString &fileName)
- void [signalAlbumAboutToBeAdded](#) ([Album](#) \*album, [Album](#) \*parent, [Album](#) \*prev)
- void [signalAlbumAdded](#) ([Album](#) \*album)
- void [signalAlbumAboutToBeDeleted](#) ([Album](#) \*album)
- void [signalAlbumDeleted](#) ([Album](#) \*album)
- void [signalAlbumHasBeenDeleted](#) ([Album](#) \*album)

- void **signalAlbumsCleared** ()
- void **signalAlbumCurrentChanged** (const QList< Album \* > &albums)
- void **signalAllAlbumsLoaded** ()
- void **signalAlbumIconChanged** (Album \*album)
- void **signalAlbumRenamed** (Album \*album)
- void **signalAlbumNewPath** (Album \*album)
- void **signalAlbumAboutToBeMoved** (Album \*album)
- void **signalAlbumMoved** (Album \*album)
- void **signalAlbumsUpdated** (int type)
- void **signalShowOnlyAvailableAlbumsChanged** (bool showsOnlyAvailableAlbums)

### Operations on Date Album

- AlbumList **allDAAlbums** () const
- DAAlbum \* **findDAAlbum** (int id) const
- QMap< YearMonth, int > **getDAAlbumsCount** () const
- void **signalDAAlbumsDirty** (const QMap< YearMonth, int > &)
- void **signalDatesHashDirty** (const QHash< QDateTime, int > &)
- void **signalAllDAAlbumsLoaded** ()

### Operations on Physical Album

- AlbumList **allPAAlbums** () const
- PAAlbum \* **currentPAAlbum** () const
- PAAlbum \* **findPAAlbum** (const QUrl &url) const
- PAAlbum \* **findPAAlbum** (int id) const
- QHash< int, int > **getPAAlbumsCount** () const
- void **removeWatchedPAAlbums** (const PAAlbum \*const album)
- PAAlbum \* **createPAAlbum** (PAAlbum \*parent, const QString &name, const QString &caption, const QDate &date, const QString &category, QString &errMsg)
- PAAlbum \* **createPAAlbum** (const QString &albumRootPath, const QString &name, const QString &caption, const QDate &date, const QString &category, QString &errMsg)
- PAAlbum \* **createPAAlbum** (const CollectionLocation &location, const QString &name, const QString &caption, const QDate &date, const QString &category, QString &errMsg)
- bool **renamePAAlbum** (PAAlbum \*album, const QString &newName, QString &errMsg)
- bool **updatePAAlbumIcon** (PAAlbum \*album, qlonglong iconID, QString &errMsg)
- void **signalPAAlbumsDirty** (const QHash< int, int > &)
- void **signalEmptyTrash** ()

### Operations on Tag Album

- AlbumList **allTAAlbums** () const
- QList< TAAlbum \* > **currentTAAlbums** () const
- TAAlbum \* **findTAAlbum** (int id) const
- TAAlbum \* **findTAAlbum** (const QString &tagPath) const
- QHash< int, int > **getTAAlbumsCount** () const
- TAAlbum \* **createTAAlbum** (TAAlbum \*parent, const QString &name, const QString &iconkde, QString &errMsg)
- AlbumList **findOrCreateTAAlbums** (const QStringList &tagPaths)
- bool **deleteTAAlbum** (TAAlbum \*album, QString &errMsg, QList< qlonglong > \*imageIds=nullptr)
- bool **renameTAAlbum** (TAAlbum \*album, const QString &name, QString &errMsg)
- bool **moveTAAlbum** (TAAlbum \*album, TAAlbum \*newParent, QString &errMsg)
- bool **mergeTAAlbum** (TAAlbum \*album, TAAlbum \*destAlbum, bool dialog, QString &errMsg)

- bool [updateAlbumIcon](#) ([TAlbum](#) \*album, const QString &iconKDE, qlonglong iconID, QString &errMsg)
- AlbumList [getRecentlyAssignedTags](#) (bool includeInternal=false) const
- QStringList [tagPaths](#) (const QList< int > &tagIDs, bool leadingSlash=true, bool includeInternal=false) const
- QStringList [tagNames](#) (const QList< int > &tagIDs, bool includeInternal=false) const
- QHash< int, QString > [tagPaths](#) (bool leadingSlash=true, bool includeInternal=false) const
- QHash< int, QString > [tagNames](#) (bool includeInternal=false) const
- AlbumList [findTagsWithProperty](#) (const QString &property)
- AlbumList [findTagsWithProperty](#) (const QString &property, const QString &value)
- QList< int > [subTags](#) (int tagId, bool recursive=false) const
- int [findTopId](#) (int tagId) const
- void [askUserForWriteChangedAlbumToFiles](#) ([TAlbum](#) \*const album)
- void [askUserForWriteChangedAlbumToFiles](#) (const QList< qlonglong > &imagelds)
- void [signalAlbumsDirty](#) (const QHash< int, int > &)
- void [signalTagPropertiesChanged](#) ([TAlbum](#) \*album)

### Operations on Search Album

- AlbumList [allSAlbums](#) () const
- [SAlbum](#) \* [findSAlbum](#) (int id) const
- [SAlbum](#) \* [findSAlbum](#) (const QString &name) const
- QList< [SAlbum](#) \* > [findSAlbumsBySearchType](#) (int searchType) const
- [SAlbum](#) \* [createSAlbum](#) (const QString &name, DatabaseSearch::Type type, const QString &query)
- bool [updateSAlbum](#) ([SAlbum](#) \*album, const QString &changedQuery, const QString &changed←Name=QString(), DatabaseSearch::Type type=DatabaseSearch::UndefinedType)
- bool [deleteSAlbum](#) ([SAlbum](#) \*album)
- void [signalUpdateDuplicatesAlbums](#) (const QList< [SAlbum](#) \* > &modifiedAlbums, const QList< qlonglong > &deletedImages)
- void [signalSearchUpdated](#) ([SAlbum](#) \*album)

## 6.55.1 Detailed Description

[AlbumManager](#) manages albums: does listing of albums and controls the lifetime of it. For [PAlbums](#) and [TAlbums](#), the listing is done by reading the db directly and building the hierarchy of the albums. For [DAlbums](#), since the listing takes time, the work is delegated to a dbjob. Interested frontend entities can connect to the albummanager to receive notifications of new Albums, when Albums are deleted and when the current album is changed.

Additional operations are provided for: creating/deleting/rename Albums, updating icons and moving Albums.

## 6.55.2 Member Function Documentation

### 6.55.2.1 albumTitles()

```
QHash< int, QString > Digikam::AlbumManager::albumTitles ( ) const
```

#### Returns

A hash with the titles for all album IDs.

### 6.55.2.2 allDAAlbums()

```
AlbumList Digikam::AlbumManager::allDAAlbums ( ) const
```

#### Returns

a list of all DAAlbums

### 6.55.2.3 allPAAlbums()

```
AlbumList Digikam::AlbumManager::allPAAlbums ( ) const
```

#### Returns

a list of all PAAlbums

### 6.55.2.4 allSAAlbums()

```
AlbumList Digikam::AlbumManager::allSAAlbums ( ) const
```

#### Returns

a list of all SAAlbums

### 6.55.2.5 allTAAlbums()

```
AlbumList Digikam::AlbumManager::allTAAlbums ( ) const
```

#### Returns

a list of all TAAlbums

### 6.55.2.6 changeDatabase()

```
void Digikam::AlbumManager::changeDatabase (
    const DbEngineParameters & params )
```

Sets new database when chosen by the user in setup. Handles user notification about problems. Call this instead of setDatabase when digiKam is up and running.

### 6.55.2.7 checkDatabaseDirsAfterFirstRun()

```
void Digikam::AlbumManager::checkDatabaseDirsAfterFirstRun (
    const QString & dbPath,
    const QString & albumPath ) [static]
```

Some checks for settings done in first run wizard in case of QSQLite Database.

**6.55.2.8 cleanUp()**

```
void Digikam::AlbumManager::cleanUp ( )
```

Stop ongoing operations, prepare for application shutdown

**6.55.2.9 clearCurrentAlbums()**

```
void Digikam::AlbumManager::clearCurrentAlbums ( )
```

clear current albums.

**6.55.2.10 createPAlbum() [1/3]**

```
PAlbum * Digikam::AlbumManager::createPAlbum (
    const CollectionLocation & location,
    const QString & name,
    const QString & caption,
    const QDate & date,
    const QString & category,
    QString & errMsg )
```

Overloaded method. Here you can supply a collection location (which must be available).

**Parameters**

<i>location</i>	the collection for the new album
<i>name</i>	the name of the new album
<i>caption</i>	the caption for the new album
<i>date</i>	the date for the new album
<i>category</i>	the category for the new album
<i>errMsg</i>	this will contain the error message describing why the operation failed

**6.55.2.11 createPAlbum() [2/3]**

```
PAlbum * Digikam::AlbumManager::createPAlbum (
    const QString & albumRootPath,
    const QString & name,
    const QString & caption,
    const QDate & date,
    const QString & category,
    QString & errMsg )
```

Overloaded method. Here you can supply an albumRootPath which must correspond to an available collection location.

**6.55.2.12 createPAlbum() [3/3]**

```
PAlbum * Digikam::AlbumManager::createPAlbum (
    PAlbum * parent,
```



```

    const QString & name,
    const QString & caption,
    const QDate & date,
    const QString & category,
    QString & errMsg )

```

Create a new [PAAlbum](#) with supplied properties as a child of the parent This is equivalent to creating a new folder on the disk with supplied name in the parent's folder path. Also the supplied attributes are written out to the database

#### Note

the signalAlbumAdded will be fired before this function returns. Its recommended to connect to that signal to get notification of new album added

#### Returns

the newly created [PAAlbum](#) or 0 if it fails

#### Parameters

<i>parent</i>	the parent album under which to create the new <a href="#">Album</a> . Parent must not be root. Otherwise, use the other variants of this method. If parent is root, the albumRootPath must be supplied.
<i>name</i>	the name of the new album
<i>caption</i>	the caption for the new album
<i>date</i>	the date for the new album
<i>category</i>	the category for the new album
<i>errMsg</i>	this will contain the error message describing why the operation failed

#### 6.55.2.13 createSAAlbum()

```

SAAlbum * Digikam::AlbumManager::createSAAlbum (
    const QString & name,
    DatabaseSearch::Type type,
    const QString & query )

```

Create a new [SAAlbum](#) with supplied url. If an existing [SAAlbum](#) with same name exists this function will return a pointer to that album, instead of creating a new one. A newly created search album is added to the database. For an existing [SAAlbum](#), the url is updated and written out to the database

#### Note

the signalAlbumAdded will be fired before this function returns. Its recommended to connect to that signal to get notification of new album added

#### Returns

the newly created [SAAlbum](#) or an existing [SAAlbum](#) with same name

#### Parameters

<i>name</i>	name for the new search
<i>type</i>	the type of the search
<i>query</i>	search query to use

### 6.55.2.14 createTAlbum()

```
TAlbum * Digikam::AlbumManager::createTAlbum (
    TAlbum * parent,
    const QString & name,
    const QString & iconkde,
    QString & errMsg )
```

Create a new [TAlbum](#) with supplied properties as a child of the parent The tag is added to the database

#### Note

the signalAlbumAdded will be fired before this function returns. Its recommended to connect to that signal to get notification of new album added

#### Returns

the newly created [TAlbum](#) or 0 if it fails

#### Parameters

<i>parent</i>	the parent album under which to create the new <a href="#">Album</a>
<i>name</i>	the name of the new album
<i>iconkde</i>	the iconkde for the new album (this is a filename which kde iconloader can load up
<i>errMsg</i>	this will contain the error message describing why the operation failed

### 6.55.2.15 currentAlbums()

```
AlbumList Digikam::AlbumManager::currentAlbums ( ) const
```

#### Returns

current albums, previously set up by setCurrentAlbums

### 6.55.2.16 currentPAlbum()

```
PAlbum * Digikam::AlbumManager::currentPAlbum ( ) const
```

#### Returns

the current [PAlbum](#) or null if no one is selected

Temporary fix, to return multiple items, iterate and cast each element

### 6.55.2.17 currentTAlbums()

```
QList< TAlbum * > Digikam::AlbumManager::currentTAlbums ( ) const
```

#### Returns

the current [TAlbum](#) or null if no one is selected

This method is not yet used

### 6.55.2.18 databaseEqual()

```
bool Digikam::AlbumManager::databaseEqual (
    const DbEngineParameters & parameters ) const
```

Checks if the given database path is equal to the current one

### 6.55.2.19 deleteSAlbum()

```
bool Digikam::AlbumManager::deleteSAlbum (
    SAlbum * album )
```

Delete a [SAlbum](#) from the database

#### Note

the signalAlbumDeleted will be fired before this function returns. Its recommended to connect to that signal to get notification of album deletes

#### Returns

true if the operation succeeds, false otherwise

#### Parameters

<i>album</i>	the album to delete
--------------	---------------------

### 6.55.2.20 deleteTAlbum()

```
bool Digikam::AlbumManager::deleteTAlbum (
    TAlbum * album,
    QString & errMsg,
    QList< qlonglong > * imageIds = nullptr )
```

Delete a [TAlbum](#). The tag is removed from the database

**Note**

the signal `AlbumDeleted` will be fired before this function returns. Its recommended to connect to that signal to get notification of album deletes

**Returns**

true if the operation succeeds or false otherwise

**Parameters**

<i>album</i>	the <code>TAlbum</code> to delete
<i>errMsg</i>	this will contain the error message describing why the
<i>imageIds</i>	list of image ID from the database where tag is removed

**6.55.2.21 findAlbum() [1/2]**

```
Album * Digikam::AlbumManager::findAlbum (
    Album::Type type,
    int id ) const
```

**Returns**

a `Album` with the given type and id

**Parameters**

<i>type</i>	the type of album
<i>id</i>	the id for the album (not the global id)

**6.55.2.22 findAlbum() [2/2]**

```
Album * Digikam::AlbumManager::findAlbum (
    int gid ) const
```

**Returns**

a `Album` with the given globalID

**Parameters**

<i>gid</i>	the global id for the album
------------	-----------------------------

**6.55.2.23 findDAAlbum()**

```
DAlbum * Digikam::AlbumManager::findDAAlbum (
```

```
int id ) const
```

**Returns**

a [DAAlbum](#) with given ID

**Parameters**

<i>id</i>	the id for the <a href="#">DAAlbum</a>
-----------	--

**6.55.2.24 findOrCreateTAlbums()**

```
AlbumList Digikam::AlbumManager::findOrCreateTAlbums (  
    const QStringList & tagPaths )
```

A list of tag paths is supplied. If no corresponding [TAlbum](#) exists, a new one will be created.

**Parameters**

<i>tagPaths</i>	A list of tag paths
-----------------	---------------------

**Returns**

A list of all [TAlbums](#) for the list (already existing or newly created)

**6.55.2.25 findPAlbum() [1/2]**

```
PAlbum * Digikam::AlbumManager::findPAlbum (  
    const QUrl & url ) const
```

Given a complete file url (kde url with file protocol), it will try to find a [PAlbum](#) corresponding to it.

**Warning**

This should not be used, unless really necessary

**Returns**

[PAlbum](#) corresponding to supplied url

**Parameters**

<i>url</i>	the url we need to check
------------	--------------------------

### 6.55.2.26 findPAlbum() [2/2]

```
PAlbum * Digikam::AlbumManager::findPAlbum (
    int id ) const
```

#### Returns

a [PAlbum](#) with given ID

#### Parameters

<i>id</i>	the id for the <a href="#">PAlbum</a>
-----------	---------------------------------------

### 6.55.2.27 findSAlbum() [1/2]

```
SAlbum * Digikam::AlbumManager::findSAlbum (
    const QString & name ) const
```

#### Returns

a [SAlbum](#) with given name, or 0 if not found

#### Parameters

<i>name</i>	the name of the search
-------------	------------------------

### 6.55.2.28 findSAlbum() [2/2]

```
SAlbum * Digikam::AlbumManager::findSAlbum (
    int id ) const
```

#### Returns

a [SAlbum](#) with given ID

#### Parameters

<i>id</i>	the id for the <a href="#">SAlbum</a>
-----------	---------------------------------------

### 6.55.2.29 findSAlbumsBySearchType()

```
QList< SAlbum * > Digikam::AlbumManager::findSAlbumsBySearchType (
    int searchType ) const
```

**Returns**

SAlbums with given type, empty list if not found

## Parameters

<i>searchType</i>	the type of the search
-------------------	------------------------

**6.55.2.30 findTagsWithProperty()**

```
AlbumList Digikam::AlbumManager::findTagsWithProperty (
    const QString & property )
```

Returns a list of TAlbums which have the given property, or the given property/value combination.

**6.55.2.31 findTAlbum() [1/2]**

```
TAlbum * Digikam::AlbumManager::findTAlbum (
    const QString & tagPath ) const
```

## Returns

a [TAlbum](#) with given tag path, or 0 if not found

## Parameters

<i>tagPath</i>	the tag path ("People/Friend/John")
----------------	-------------------------------------

**6.55.2.32 findTAlbum() [2/2]**

```
TAlbum * Digikam::AlbumManager::findTAlbum (
    int id ) const
```

## Returns

a [TAlbum](#) with given ID

## Parameters

<i>id</i>	the id for the <a href="#">TAlbum</a>
-----------	---------------------------------------

**6.55.2.33 getDAlbumsCount()**

```
QMap< YearMonth, int > Digikam::AlbumManager::getDAlbumsCount ( ) const
```

Returns the latest count for DAlbums as also emitted via signalDAlbumsDirty.

## Returns

count map for DAlbums



### 6.55.2.34 getFaceCount()

```
QHash< int, int > Digikam::AlbumManager::getFaceCount ( ) const
```

Returns the latest count for faces as also emitted via signalFaceCountsDirty.

#### Returns

count map for faces (confirmed and unconfirmed combined)

### 6.55.2.35 getItemFromAlbum()

```
qlonglong Digikam::AlbumManager::getItemFromAlbum (
    Album *const album,
    const QString & fileName )
```

Returns the id of the item with the given filename in the given [Album](#).

#### Parameters

<i>album</i>	The album in which we search the item.
<i>fileName</i>	The name of the item file.

#### Returns

The item id or -1 if not existent.

### 6.55.2.36 getPAlbumsCount()

```
QHash< int, int > Digikam::AlbumManager::getPAlbumsCount ( ) const
```

Returns the latest count for PAlbums as also emitted via signalPAlbumsDirty.

#### Returns

count map for PAlbums

### 6.55.2.37 getRecentlyAssignedTags()

```
AlbumList Digikam::AlbumManager::getRecentlyAssignedTags (
    bool includeInternal = false ) const
```

Get a list of recently assigned tags (only last 6 tags are listed)

#### Returns

the list of recently assigned TAlbums

## Parameters

<i>includeInternal</i>	include internal tags in the returned list, or skip them
------------------------	--

**6.55.2.38 getTAlbumsCount()**

```
QHash< int, int > Digikam::AlbumManager::getTAlbumsCount ( ) const
```

Returns the latest count for TAlbums as also emitted via signalTAlbumsDirty.

## Returns

count map for TAlbums

**6.55.2.39 getUnconfirmedFaceCount()**

```
QHash< int, int > Digikam::AlbumManager::getUnconfirmedFaceCount ( ) const
```

Returns the latest count for unconfirmed faces only as also emitted via signalFaceCountsDirty.

## Returns

count map for unconfirmed faces only

**6.55.2.40 instance()**

```
AlbumManager * Digikam::AlbumManager::instance ( ) [static]
```

A convenience function to get the instance of the [AlbumManager](#)

**6.55.2.41 isMovingAlbum()**

```
bool Digikam::AlbumManager::isMovingAlbum (
    Album * album ) const
```

Returns if the given album is currently being moved, that is, if this album is in between signalAlbumAboutToBeMoved and signalAlbumMoved. In this case, you can preserve state of such an album because the object is guaranteed not to be deleted, even if signalAlbumAboutToBeDeleted is emitted.

**6.55.2.42 mergeTAlbum()**

```
bool Digikam::AlbumManager::mergeTAlbum (
    TAlbum * album,
    TAlbum * destAlbum,
    bool dialog,
    QString & errMsg )
```

Merge a [TAlbum](#) to a [TAlbum](#). This updates the image tags in the database

## Returns

true if the operation succeeds, false otherwise

## Parameters

<i>album</i>	the <a href="#">Album</a> which should be merged
<i>destAlbum</i>	the <a href="#">Album</a> to which album should be merged
<i>dialog</i>	show dialog to ask the user if he wants to merge
<i>errMsg</i>	this will contain the error message describing why the operation failed

**6.55.2.43 moveTAlbum()**

```
bool Digikam::AlbumManager::moveTAlbum (
    TAlbum * album,
    TAlbum * newParent,
    QString & errMsg )
```

Move a [TAlbum](#) to a new parent. This updates the tag parent ID in the database

## Returns

true if the operation succeeds, false otherwise

## Parameters

<i>album</i>	the <a href="#">Album</a> which should be moved
<i>newParent</i>	the Parent <a href="#">Album</a> to which album should be moved
<i>errMsg</i>	this will contain the error message describing why the operation failed

**6.55.2.44 prepareItemCounts()**

```
void Digikam::AlbumManager::prepareItemCounts ( )
```

Ensures that valid item counts for physical and tag albums are available

**6.55.2.45 refresh()**

```
void Digikam::AlbumManager::refresh ( )
```

This is similar to `startScan`, except that it assumes you have run `startScan` at least once. It checks the database to see if any new albums have been added and updates them accordingly. Use this when a change in the filesystem is detected (but the album library path hasn't changed)

## See also

[startScan](#)

**6.55.2.46 renamePAlbum()**

```
bool Digikam::AlbumManager::renamePAlbum (
    PAlbum * album,
    const QString & newName,
    QString & errMsg )
```

Renames a [PAlbum](#). This is equivalent to actually renaming the corresponding folder on the disk.

**Returns**

true if the operation succeeds, false otherwise

**Parameters**

<i>album</i>	the <a href="#">Album</a> which should be renamed
<i>newName</i>	the new name for the album
<i>errMsg</i>	this will contain the error message describing why the operation failed

**6.55.2.47 renameTAlbum()**

```
bool Digikam::AlbumManager::renameTAlbum (
    TAlbum * album,
    const QString & name,
    QString & errMsg )
```

Renames a [TAlbum](#). This updates the tag name in the database

**Returns**

true if the operation succeeds, false otherwise

**Parameters**

<i>album</i>	the <a href="#">Album</a> which should be renamed
<i>name</i>	the new name for the album
<i>errMsg</i>	this will contain the error message describing why the operation failed

**6.55.2.48 setCurrentAlbums()**

```
void Digikam::AlbumManager::setCurrentAlbums (
    const QList< Album * > & albums )
```

set current album to albums. [Filter](#) out the null pointers

Sort is needed to identify selection correctly, ex [AlbumHistory](#)

### 6.55.2.49 setDatabase()

```
bool Digikam::AlbumManager::setDatabase (
    const DbEngineParameters & params,
    bool priority,
    const QString & suggestedAlbumRoot = QString(),
    bool ignoreDisappearedLocations = false )
```

Initialize. Informs the user about failures. Returns true on success, false on failure. A return value of false during startup indicates termination of the program (user is informed)

ignoreDisappearedLocations is intended to be used in tests, because the path of the collection is hardcoded but when executing the test on different computers the collection might not be available at that path

### 6.55.2.50 signalAlbumAboutToBeAdded

```
void Digikam::AlbumManager::signalAlbumAboutToBeAdded (
    Album * album,
    Album * parent,
    Album * prev ) [signal]
```

Emitted when an album is about to be added to the given parent (0 if album is root) after the item given by prev (prev is 0 if parent has no children yet).

### 6.55.2.51 signalAlbumAboutToBeDeleted

```
void Digikam::AlbumManager::signalAlbumAboutToBeDeleted (
    Album * album ) [signal]
```

Emitted when the album is about to be deleted, but is still fully valid.

### 6.55.2.52 signalAlbumAboutToBeMoved

```
void Digikam::AlbumManager::signalAlbumAboutToBeMoved (
    Album * album ) [signal]
```

Emitted when an album is about to be moved. Signals for deleting and adding will be sent afterwards, but the album object is guaranteed not to be deleted until after signalAlbumMoved.

### 6.55.2.53 signalAlbumAdded

```
void Digikam::AlbumManager::signalAlbumAdded (
    Album * album ) [signal]
```

Emitted when the album has been added.

### 6.55.2.54 signalAlbumDeleted

```
void Digikam::AlbumManager::signalAlbumDeleted (
    Album * album ) [signal]
```

Emitted when the album is deleted, but the object can still be accessed.

### 6.55.2.55 signalAlbumHasBeenDeleted

```
void Digikam::AlbumManager::signalAlbumHasBeenDeleted (
    Album * album ) [signal]
```

Emitted when the album is deleted, the object can no longer be accessed. For identification purposes, the former album pointer is passed.

### 6.55.2.56 signalAlbumMoved

```
void Digikam::AlbumManager::signalAlbumMoved (
    Album * album ) [signal]
```

Emitted when the album is moved to its new parent. After signalAlbumAboutToBeMoved, all four signals for first deleting and then adding will have been sent.

### 6.55.2.57 signalShowOnlyAvailableAlbumsChanged

```
void Digikam::AlbumManager::signalShowOnlyAvailableAlbumsChanged (
    bool showsOnlyAvailableAlbums ) [signal]
```

Emitted when a change is done on available Albums. Please note that affected albums may appear or disappear after this signal has been emitted.

### 6.55.2.58 startScan()

```
void Digikam::AlbumManager::startScan ( )
```

starts scanning the libraryPath and listing the albums. If the libraryPath has not changed since the last scan, then nothing happens

See also

[setLibraryPath](#)  
[refresh](#)

### 6.55.2.59 subTags()

```
QList< int > Digikam::AlbumManager::subTags (
    int tagId,
    bool recursive = false ) const
```

TODO

### 6.55.2.60 tagNames() [1/2]

```
QHash< int, QString > Digikam::AlbumManager::tagNames (
    bool includeInternal = false ) const
```

Returns

A hash with the tag names for all tag IDs.

## Parameters

<i>includeInternal</i>	include internal tags in the returned list, or skip them
------------------------	--

**6.55.2.61 tagNames()** [2/2]

```
QStringList Digikam::AlbumManager::tagNames (
    const QList< int > & tagIDs,
    bool includeInternal = false ) const
```

## Returns

A list with the tag names for a list of tag IDs.

## Parameters

<i>tagIDs</i>	list of tag album IDs
<i>includeInternal</i>	include internal tags in the returned list, or skip them

**6.55.2.62 tagPaths()** [1/2]

```
QHash< int, QString > Digikam::AlbumManager::tagPaths (
    bool leadingSlash = true,
    bool includeInternal = false ) const
```

## Returns

A hash with the tag paths for all tag IDs.

## Parameters

<i>leadingSlash</i>	if <code>true</code> return tags with a leading slash
<i>includeInternal</i>	include internal tags in the returned list, or skip them

**6.55.2.63 tagPaths()** [2/2]

```
QStringList Digikam::AlbumManager::tagPaths (
    const QList< int > & tagIDs,
    bool leadingSlash = true,
    bool includeInternal = false ) const
```

Return A list with the tag paths for a list of tag IDs.

## Parameters

<i>tagIDs</i>	list of tag album IDs
<i>leadingSlash</i>	if <code>true</code> return tags with a leading slash
<i>includeInternal</i>	include internal tags in the returned list, or skip them

**6.55.2.64 updatePAlbumIcon()**

```
bool Digikam::AlbumManager::updatePAlbumIcon (
    PAlbum * album,
    qlonglong iconID,
    QString & errMsg )
```

Update the icon for an album. The `icon` is the name (and not full path) of the file in the album

**Returns**

true if the operation succeeds, false otherwise

**Parameters**

<i>album</i>	the album for which icon should be changed
<i>iconID</i>	the filename of the new icon
<i>errMsg</i>	if the operation fails, this will contain the error message describing why the operation failed

**6.55.2.65 updateSAlbum()**

```
bool Digikam::AlbumManager::updateSAlbum (
    SAlbum * album,
    const QString & changedQuery,
    const QString & changedName = QString(),
    DatabaseSearch::Type type = DatabaseSearch::UndefinedType )
```

Update the url for a [SAlbum](#)

**Returns**

true if the operation succeeds, false otherwise

**Parameters**

<i>album</i>	the album to update
<i>changedQuery</i>	the new query data of the album
<i>changedName</i>	a new name, or null to keep the current name
<i>type</i>	a new type, or UndefinedType to keep the current type

**6.55.2.66 updateTAlbumIcon()**

```
bool Digikam::AlbumManager::updateTAlbumIcon (
    TAlbum * album,
    const QString & iconKDE,
    qlonglong iconID,
    QString & errMsg )
```

Update the icon for a [TAlbum](#).



## Returns

true if the operation succeeds, false otherwise

## Parameters

<i>album</i>	the album for which icon should be changed
<i>iconKDE</i>	a simple filename which can be loaded by KIconLoader
<i>iconID</i>	id of the icon image file
<i>errMsg</i>	this will contain the error message describing why the operation failed

## Note

if iconKDE is not empty then iconID is used. So if you want to set the icon to a file which can be loaded by QIcon, pass it in as iconKDE. otherwise pass a null QString to iconKDE and set iconID

## 6.56 Digikam::AlbumManager::Private Class Reference

### Public Member Functions

- QString **labelForAlbumRootAlbum** (const [CollectionLocation](#) &location)

### Public Attributes

- QTimer \* **albumItemCountTimer** = nullptr
- [AlbumsDBJobsThread](#) \* **albumListJob** = nullptr
- QHash< [PAlbumPath](#), [PAlbum](#) \* > **albumPathHash**
- QHash< int, [PAlbum](#) \* > **albumRootAlbumHash**
- [AlbumWatch](#) \* **albumWatch** = nullptr
- QHash< int, [Album](#) \* > **allAlbumsIdHash**
- int **askMergeMessageBoxResult** = -1
- bool **changed** = false
- QSet< int > **changedPAlbums**
- bool **changingDB** = false
- QList< [Album](#) \* > **currentAlbums**
- [Album](#) \* **currentlyMovingAlbum** = nullptr
- QMap< YearMonth, int > **dAlbumsCount**
- [DatesDBJobsThread](#) \* **dateListJob** = nullptr
- QHash< int, int > **fAlbumsCount**
- QMultiHash< [Album](#) \*, [Album](#) \*\* > **guardedPointers**
- bool **hasPriorizedDbPath** = false
- int **longTimeMessageBoxResult** = -1
- QHash< int, int > **pAlbumsCount**
- [TagsDBJobsThread](#) \* **personListJob** = nullptr
- [DAAlbum](#) \* **rootDAAlbum** = nullptr
- [PAlbum](#) \* **rootPAlbum** = nullptr
- [SAAlbum](#) \* **rootSAAlbum** = nullptr
- [TAAlbum](#) \* **rootTAAlbum** = nullptr
- QTimer \* **scanDAAlbumsTimer** = nullptr
- QTimer \* **scanPAlbumsTimer** = nullptr
- QTimer \* **scanSAAlbumsTimer** = nullptr

- QTimer \* **scanTAlbumsTimer** = nullptr
- bool **showOnlyAvailableAlbums** = false
- QTimer \* **tagItemCountTimer** = nullptr
- [TagsDBJobsThread](#) \* **tagListJob** = nullptr
- QHash< int, int > **tAlbumsCount**
- QList< int > **toUpdatedFaces**
- QHash< int, int > **uAlbumsCount**  
*Unconfirmed face counts.*
- QTimer \* **updatePAlbumsTimer** = nullptr

## 6.56.1 Member Data Documentation

### 6.56.1.1 currentAlbums

`QList<Album*> Digikam::AlbumManager::Private::currentAlbums`

For multiple selection support

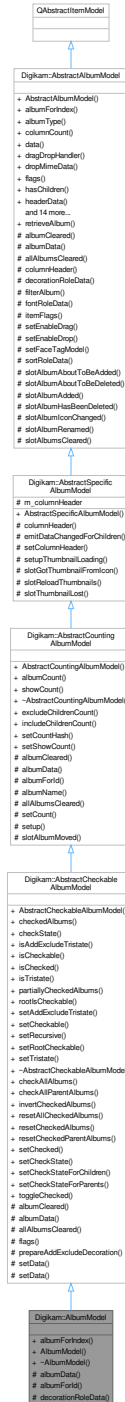
## 6.57 Digikam::AlbumManagerCreator Class Reference

### Public Attributes

- [AlbumManager](#) object

## 6.58 Digikam::AlbumModel Class Reference

Inheritance diagram for Digikam::AlbumModel:



### Public Member Functions

- [PAAlbum](#) \* [albumForIndex](#) (const QModelIndex &index) const
- [AlbumModel](#) ([RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumModel](#)

- [AbstractCheckableAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [QList](#)< [Album](#) \* > **checkedAlbums** () const  
*Returns a list of album with check state Checked.*
- [Qt::CheckState](#) **checkState** ([Album](#) \*album) const  
*Returns the check state of the album.*
- bool **isAddExcludeTristate** () const
- bool **isCheckable** () const
- bool **isChecked** ([Album](#) \*album) const  
*Returns if the given album has the check state Checked.*
- bool **isTristate** () const
- [QList](#)< [Album](#) \* > **partiallyCheckedAlbums** () const  
*Returns a list of album with partially check state Checked.*
- bool **rootsCheckable** () const
- void **setAddExcludeTristate** (bool b)
- void **setCheckable** (bool isCheckable)  
*Triggers if the albums in this model are checkable.*
- void **setRecursive** (bool recursive)
- void **setRootCheckable** (bool rootsCheckable)
- void **setTristate** (bool isTristate)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- [AbstractCountingAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Supports displaying a count alongside the album name in DisplayRole.*
- virtual int **albumCount** ([Album](#) \*album) const
- bool **showCount** () const

## Public Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- [AbstractSpecificAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Abstract base class, do not instantiate.*

## Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- [AbstractAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [Album](#) \* **albumForIndex** (const [QModelIndex](#) &index) const
- [Album::Type](#) albumType () const
- int **columnCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [AlbumModelDragDropHandler](#) \* **dragDropHandler** () const
- bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role=[Qt::DisplayRole](#)) const override

- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex **indexForAlbum** (Album \*album) const
- bool **isFaceTagModel** () const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const override
- QStringList **mimeTypes** () const override
- QModelIndex **parent** (const QModelIndex &index) const override
- Album \* **rootAlbum** () const
- RootAlbumBehavior **rootAlbumBehavior** () const
- QModelIndex **rootAlbumIndex** () const
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setDragDropHandler** (AlbumModelDragDropHandler \*handler)
- void **setDropIndex** (const QModelIndex &index)
- Qt::DropActions **supportedDropActions** () const override

### Protected Member Functions

- QVariant **albumData** (Album \*a, int role) const override  
*For subclassing convenience: A part of the implementation of data()*
- Album \* **albumForId** (int id) const override  
*need to implement in subclass*
- QVariant **decorationRoleData** (Album \*a) const override  
*For subclassing convenience: A part of the implementation of data()*

### Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumModel](#)

- void **albumCleared** (Album \*album) override  
*Notification when an entry is removed.*
- void **allAlbumsCleared** () override  
*Notification when all entries are removed.*
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- void **prepareAddExcludeDecoration** (Album \*a, QPixmap &icon) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role, bool recursive)
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- void **albumCleared** (Album \*album) override  
*Notification when an entry is removed.*
- virtual QString **albumName** (Album \*a) const  
*Can reimplement in subclass.*
- void **allAlbumsCleared** () override  
*Notification when all entries are removed.*
- void **setCount** (Album \*album, int count)  
*If you do not use setCountHash, excludeChildrenCount and includeChildrenCount, you can set a count here.*
- void **setup** ()

## Protected Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString [columnHeader](#) () const override  
*For subclassing convenience: A part of the implementation of headerData()*
- void **emitDataChangedForChildren** ([Album](#) \*album)
- void **setColumnHeader** (const QString &header)
- void **setupThumbnailLoading** ()  
*You need to call this from your constructor if you intend to load the thumbnail facilities of this class.*

## Protected Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- virtual bool [filterAlbum](#) ([Album](#) \*album) const
- virtual QVariant [fontRoleData](#) ([Album](#) \*a) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual Qt::ItemFlags **itemFlags** ([Album](#) \*album) const  
*For subclassing convenience: A part of the implementation of itemFlags()*
- void [setEnableDrag](#) (bool enable)
- void **setEnableDrop** (bool enable)
- void **setFaceTagModel** (bool enable)
- virtual QVariant [sortRoleData](#) ([Album](#) \*a) const  
*For subclassing convenience: A part of the implementation of data()*

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractAlbumModel](#)

- enum [AlbumDataRole](#) {  
[AlbumTitleRole](#) = Qt::UserRole , [AlbumTypeRole](#) = Qt::UserRole + 1 , [AlbumPointerRole](#) = Qt::UserRole + 2  
, [AlbumIdRole](#) = Qt::UserRole + 3 ,  
[AlbumGlobalIdRole](#) = Qt::UserRole + 4 , [AlbumSortRole](#) = Qt::UserRole + 5 }
- enum [RootAlbumBehavior](#) { [IncludeRootAlbum](#) , [IgnoreRootAlbum](#) }

## Public Slots inherited from [Digikam::AbstractCheckableAlbumModel](#)

- void **checkAllAlbums** (const QModelIndex &parent=QModelIndex())  
*Checks all albums beneath the given parent.*
- void **checkAllParentAlbums** (const QModelIndex &child)  
*Checks all parent albums starting at the child, including it.*
- void **invertCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Inverts the checked state of all albums under the given parent.*
- void **resetAllCheckedAlbums** ()  
*Resets the checked state of all albums to Qt::Unchecked.*
- void **resetCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Resets the checked state of all albums under the given parent.*
- void **resetCheckedParentAlbums** (const QModelIndex &child)  
*Resets the checked state of all parents of the child including it.*
- void **setChecked** ([Album](#) \*album, bool [isChecked](#))  
*Sets the check state of album to Checked or Unchecked.*
- void **setCheckState** ([Album](#) \*album, Qt::CheckState state)  
*Sets the check state of the album.*
- void **setCheckStateForChildren** ([Album](#) \*album, Qt::CheckState state)  
*Sets the checked state recursively for all children of but not for the given album.*
- void **setCheckStateForParents** ([Album](#) \*album, Qt::CheckState state)  
*Sets the checked state recursively for all parents of but not for the given album.*
- void **toggleChecked** ([Album](#) \*album)  
*Toggles the check state of album between Checked or Unchecked.*

### Public Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [excludeChildrenCount](#) (const QModelIndex &index)
- void [includeChildrenCount](#) (const QModelIndex &index)
- void [setCountHash](#) (const QHash< int, int > &idCountHash)
- void [setShowCount](#) (bool show)

*Call to enable or disable showing the count. Default is false.*

### Signals inherited from [Digikam::AbstractCheckableAlbumModel](#)

- void [checkStateChanged](#) (Album \*album, Qt::CheckState checkState)

### Signals inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [signalUpdateAlbumCount](#) (Album \*album)

### Signals inherited from [Digikam::AbstractAlbumModel](#)

- void [rootAlbumAvailable](#) ()

### Static Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- static Album \* [retrieveAlbum](#) (const QModelIndex &index)

### Protected Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [slotAlbumMoved](#) (Album \*album)

### Protected Slots inherited from [Digikam::AbstractSpecificAlbumModel](#)

- void [slotGotThumbnailFromIcon](#) (Album \*album, const QPixmap &thumbnail)
- void [slotReloadThumbnails](#) ()
- void [slotThumbnailLost](#) (Album \*album)

### Protected Slots inherited from [Digikam::AbstractAlbumModel](#)

- void [slotAlbumAboutToBeAdded](#) (Album \*album, Album \*parent, Album \*prev)
- void [slotAlbumAboutToBeDeleted](#) (Album \*album)
- void [slotAlbumAdded](#) (Album \*)
- void [slotAlbumHasBeenDeleted](#) (Album \*album)
- void [slotAlbumIconChanged](#) (Album \*album)
- void [slotAlbumRenamed](#) (Album \*album)
- void [slotAlbumsCleared](#) ()

### Protected Attributes inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString [m\\_columnHeader](#)

## 6.58.1 Constructor & Destructor Documentation

### 6.58.1.1 AlbumModel()

```
Digikam::AlbumModel::AlbumModel (
    RootAlbumBehavior rootBehavior = IncludeRootAlbum,
    QObject *const parent = nullptr ) [explicit]
```

Create a model containing all physical albums

## 6.58.2 Member Function Documentation

### 6.58.2.1 albumData()

```
QVariant Digikam::AlbumModel::albumData (
    Album * a,
    int role ) const [override], [protected], [virtual]
```

#### Note

these can be reimplemented in a subclass

Reimplemented from [Digikam::AbstractCheckableAlbumModel](#).

### 6.58.2.2 albumForId()

```
Album * Digikam::AlbumModel::albumForId (
    int id ) const [override], [protected], [virtual]
```

Implements [Digikam::AbstractCountingAlbumModel](#).

### 6.58.2.3 decorationRoleData()

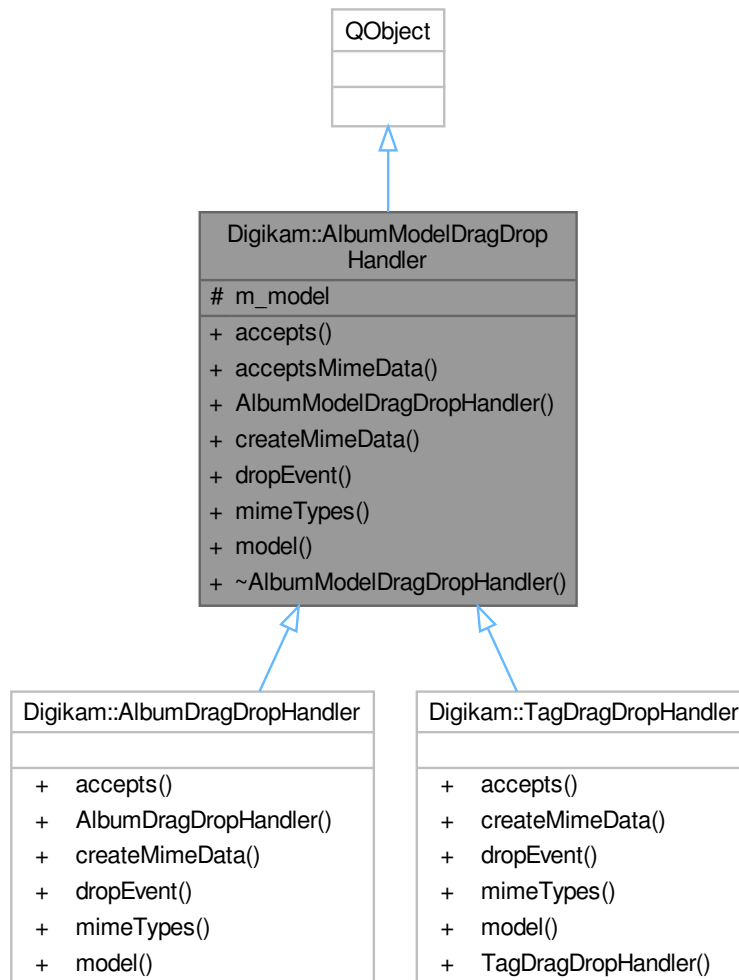
```
QVariant Digikam::AlbumModel::decorationRoleData (
    Album * a ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).



## 6.59 Digikam::AlbumModelDragDropHandler Class Reference

Inheritance diagram for Digikam::AlbumModelDragDropHandler:



### Public Member Functions

- virtual Qt::DropAction `accepts` (const QDropEvent \*e, const QModelIndex &dropIndex)
- virtual bool `acceptsMimeData` (const QMimeData \*data)
- **AlbumModelDragDropHandler** (`AbstractAlbumModel` \*model)
- virtual QMimeData \* `createMimeData` (const QList< Album \* > &)
- virtual bool `dropEvent` (QAbstractItemView \*view, const QDropEvent \*e, const QModelIndex &droppedOn)
- virtual QStringList `mimeTypes` () const
- `AbstractAlbumModel` \* **model** () const

### Protected Attributes

- `AbstractAlbumModel` \* **m\_model** = nullptr

## 6.59.1 Member Function Documentation

### 6.59.1.1 accepts()

```
Qt::DropAction Digikam::AlbumModelDragDropHandler::accepts (
    const QDropEvent * e,
    const QModelIndex & dropIndex ) [virtual]
```

Returns if the given mime data is accepted for drop on dropIndex. Returns the proposed action, or Qt::IgnoreAction if not accepted.

Reimplemented in [Digikam::AlbumDragDropHandler](#), and [Digikam::TagDragDropHandler](#).

### 6.59.1.2 acceptsMimeData()

```
bool Digikam::AlbumModelDragDropHandler::acceptsMimeData (
    const QMimeData * data ) [virtual]
```

Returns if the given mime data can be handled. `acceptsMimeData` shall return true if a drop of the given mime data will be accepted on any index or place at all. If this returns false, the more specific method [accepts\(\)](#) will not be called for this drag. The default implementation uses [mimeTypes\(\)](#) to check for supported mime types. There is usually no need to reimplement this.

### 6.59.1.3 createMimeData()

```
QMimeData * Digikam::AlbumModelDragDropHandler::createMimeData (
    const QList< Album * > & ) [virtual]
```

Create a mime data object for starting a drag from the given Albums

Reimplemented in [Digikam::AlbumDragDropHandler](#), and [Digikam::TagDragDropHandler](#).

### 6.59.1.4 dropEvent()

```
bool Digikam::AlbumModelDragDropHandler::dropEvent (
    QAbstractItemView * view,
    const QDropEvent * e,
    const QModelIndex & droppedOn ) [virtual]
```

Gives the view and the occurring drop event. The index is the index where the drop was dropped on. It may be invalid (dropped on decoration, viewport) Returns true if the event is to be accepted.

Reimplemented in [Digikam::AlbumDragDropHandler](#), and [Digikam::TagDragDropHandler](#).

### 6.59.1.5 mimeTypes()

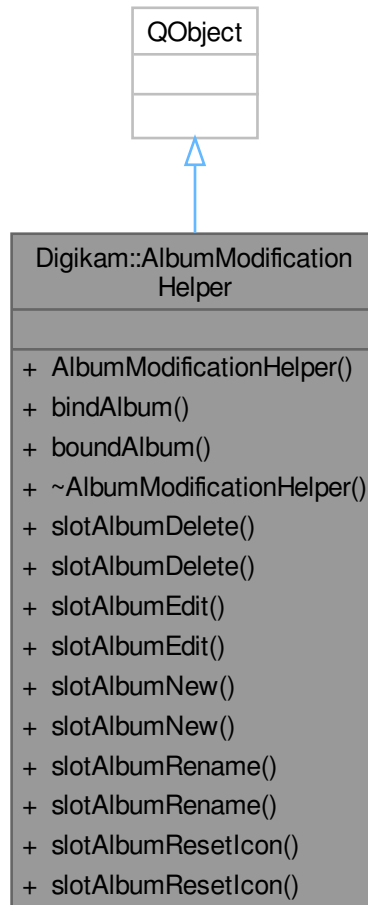
```
QStringList Digikam::AlbumModelDragDropHandler::mimeTypes ( ) const [virtual]
```

Returns the supported mime types. Called by the default implementation of model's [mimeTypes\(\)](#).

Reimplemented in [Digikam::AlbumDragDropHandler](#), and [Digikam::TagDragDropHandler](#).

## 6.60 Digikam::AlbumModificationHelper Class Reference

Inheritance diagram for Digikam::AlbumModificationHelper:



### Public Slots

- void **slotAlbumDelete** ()
- void **slotAlbumDelete** ([PAlbum](#) \*album)
- void **slotAlbumEdit** ()
- void **slotAlbumEdit** ([PAlbum](#) \*album)
- [PAlbum](#) \* **slotAlbumNew** ()
- [PAlbum](#) \* **slotAlbumNew** ([PAlbum](#) \*parentAlbum)
- void **slotAlbumRename** ()
- void **slotAlbumRename** ([PAlbum](#) \*album)
- void **slotAlbumResetIcon** ()
- void **slotAlbumResetIcon** ([PAlbum](#) \*album)

## Public Member Functions

- [AlbumModificationHelper](#) (QObject \*const parent, QWidget \*const dialogParent)
- void [bindAlbum](#) (QAction \*const action, [PAlbum](#) \*const parent) const
- [PAlbum](#) \* [boundAlbum](#) (QObject \*const action) const
- [~AlbumModificationHelper](#) () override

### 6.60.1 Detailed Description

Utility class providing methods to modify physical albums ([PAlbum](#)) in a way useful to implement views.

#### Author

jwienke

### 6.60.2 Constructor & Destructor Documentation

#### 6.60.2.1 AlbumModificationHelper()

```
Digikam::AlbumModificationHelper::AlbumModificationHelper (
    QObject *const parent,
    QWidget *const dialogParent ) [explicit]
```

Constructor.

#### Parameters

<i>parent</i>	the parent for qt parent child mechanism
<i>dialogParent</i>	parent widget for dialogs displayed by this object

#### 6.60.2.2 ~AlbumModificationHelper()

```
Digikam::AlbumModificationHelper::~~AlbumModificationHelper ( ) [override]
```

Destructor.

### 6.60.3 Member Function Documentation

#### 6.60.3.1 bindAlbum()

```
void Digikam::AlbumModificationHelper::bindAlbum (
    QAction *const action,
    PAlbum *const parent ) const
```

Sets the album that the given action operates on. You must call `bindTag` and then connect the action's triggered to the desired slot, `slotTagNew()`, `slotTagEdit()` or `slotTagDelete()`. Note: Changes the Action's user data.

### 6.60.3.2 boundAlbum()

```
PAlbum * Digikam::AlbumModificationHelper::boundAlbum (
    QObject *const action ) const
```

Returns the album bound with bindAlbum. The given QObject shall be a QAction, but for convenience the given object will be checked with qobject\_cast first, so you can pass QObject::sender().

### 6.60.3.3 slotAlbumDelete

```
void Digikam::AlbumModificationHelper::slotAlbumDelete (
    PAlbum * album ) [slot]
```

Deletes the given album after waiting for a graphical confirmation of the user.

#### Parameters

<i>album</i>	the album to delete
--------------	---------------------

### 6.60.3.4 slotAlbumEdit

```
void Digikam::AlbumModificationHelper::slotAlbumEdit (
    PAlbum * album ) [slot]
```

Graphically edits the properties of the given album.

#### Parameters

<i>album</i>	the album to edit
--------------	-------------------

### 6.60.3.5 slotAlbumNew

```
PAlbum * Digikam::AlbumModificationHelper::slotAlbumNew (
    PAlbum * parentAlbum ) [slot]
```

Creates a new album under the given parent. The user will be prompted for the settings of the new album.

#### Parameters

<i>parentAlbum</i>	parent album for the new one
--------------------	------------------------------

#### Returns

the new album or 0 if no album was created

### 6.60.3.6 slotAlbumRename

```
void Digikam::AlbumModificationHelper::slotAlbumRename (  
    PAlbum * album ) [slot]
```

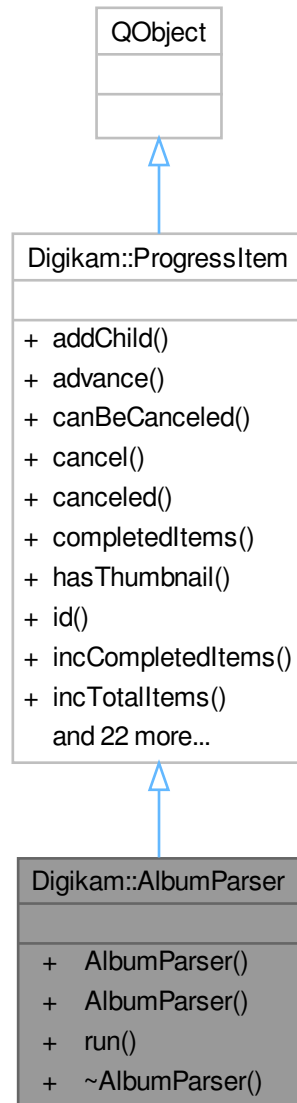
Renames the given album. The user will be prompted for a new name.

#### Parameters

<i>album</i>	the album to rename
--------------	---------------------

## 6.61 Digikam::AlbumParser Class Reference

Inheritance diagram for Digikam::AlbumParser:



### Signals

- void **signalComplete** (const QList< QUrl > &)

### Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)

- Emitted when a new [ProgressItem](#) is added.*

  - void [progressItemCanceled](#) ([ProgressItem](#) \*item)

*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)
- Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)
- Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)
- Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)
- Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)
- Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)
- Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- [AlbumParser](#) ([Album](#) \*const album)
- [AlbumParser](#) (const [ItemInfoList](#) &infoList)
- void **run** ()

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)
- Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- [ProgressItem](#) ([ProgressItem](#) \*const [parent](#), const QString &[id](#), const QString &[label](#), const QString &[status](#), bool [canBeCanceled](#), bool [hasThumb](#))
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
- Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()



Tell the item it has finished. This will emit `progressItemCompleted()` result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.

- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)
- void **setProgress** (unsigned int v)
  - Set the progress (percentage of completion) value of this item.*
- void **setShowAtStart** (bool showAtStart)
  - Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void **setStatus** (const QString &v)
  - Set the string to be used for showing this item's current status.*
- void **setThumbnail** (const QIcon &icon)
  - Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void **setUsesBusyIndicator** (bool useBusyIndicator)
  - Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling `setProgress()` from time to time to update the busy indicator.*
- bool **showAtStart** () const
- const QString & **status** () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
  - Recalculate progress according to total/completed items and update.*
- bool **usesBusyIndicator** () const

## 6.61.1 Constructor & Destructor Documentation

### 6.61.1.1 AlbumParser() [1/2]

```
Digikam::AlbumParser::AlbumParser (
    const ItemInfoList & infoList ) [explicit]
```

Constructor to work on image list

### 6.61.1.2 AlbumParser() [2/2]

```
Digikam::AlbumParser::AlbumParser (
    Album *const album ) [explicit]
```

Constructor to work on recursive mode from album

## 6.62 Digikam::AlbumPointer< T > Class Template Reference

### Public Member Functions

- **AlbumPointer** (const AlbumPointer< T > &p)
- **AlbumPointer** (T \*const a)
- **operator T\*** () const
- bool **operator!** () const
- T & **operator\*** () const
- T \* **operator->** () const
- AlbumPointer< T > & **operator=** (const AlbumPointer< T > &p)
- AlbumPointer< T > & **operator=** (T \*const a)

## Friends

- class `AlbumManager`

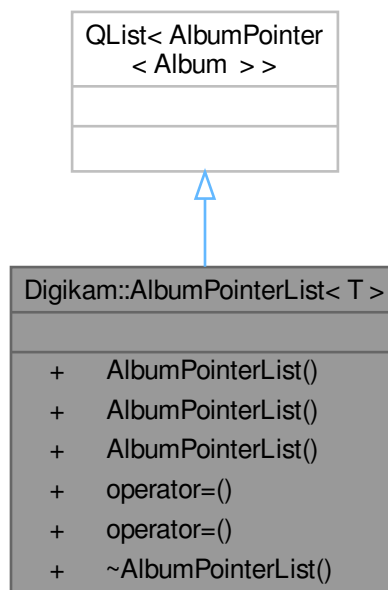
### 6.62.1 Detailed Description

```
template<class T = Album>
class Digikam::AlbumPointer< T >
```

You can use `AlbumPointer` to store a guarded pointer to `Album` or one of the subclasses (use template parameter). The pointer will be set to 0 when the album object is deleted.

## 6.63 Digikam::AlbumPointerList< T > Class Template Reference

Inheritance diagram for Digikam::AlbumPointerList< T >:

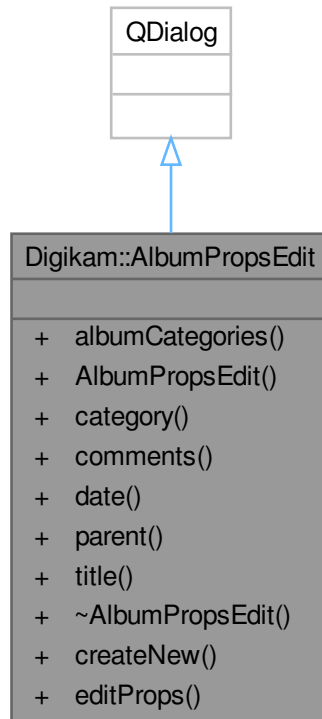


## Public Member Functions

- `AlbumPointerList` (const `AlbumPointerList< T >` &list)
- `AlbumPointerList` (const `QList< T * >` &list)
- `AlbumPointerList< T >` & `operator=` (const `AlbumPointerList< T >` &list)
- `AlbumPointerList< T >` & `operator=` (const `QList< T * >` &list)

## 6.64 Digikam::AlbumPropsEdit Class Reference

Inheritance diagram for Digikam::AlbumPropsEdit:



### Public Member Functions

- QStringList **albumCategories** () const
- **AlbumPropsEdit** ([PAlbum](#) \*const album, bool create=false)
- QString **category** () const
- QString **comments** () const
- QDate **date** () const
- int **parent** () const
- QString **title** () const

### Static Public Member Functions

- static bool **createNew** ([PAlbum](#) \*const parent, QString &title, QString &comments, QDate &date, QString &category, QStringList &albumCategories, int &parentSelector)
- static bool **editProps** ([PAlbum](#) \*const album, QString &title, QString &comments, QDate &date, QString &category, QStringList &albumCategories)

## 6.65 Digikam::AlbumRootChangeset Class Reference

### Public Types

- enum **Operation** { **Unknown** , **Added** , **Deleted** , **PropertiesChanged** }

### Public Member Functions

- **AlbumRootChangeset** (int albumRootId, Operation operation)
- int **albumRootId** () const
- Operation **operation** () const

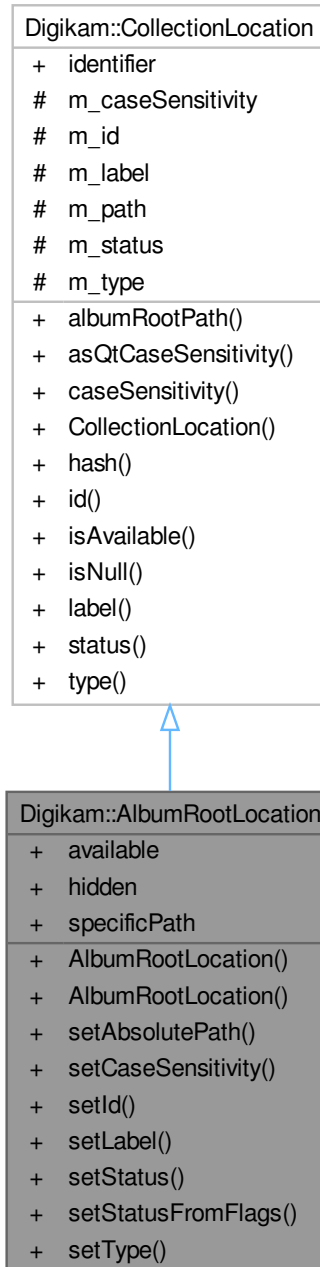
## 6.66 Digikam::AlbumRootInfo Class Reference

### Public Attributes

- int **caseSensitivity** = 0
- int **id** = 0
- QString **identifier**
- QString **label**
- QString **specificPath**
- int **status** = 0
- int **type** = 0

## 6.67 Digikam::AlbumRootLocation Class Reference

Inheritance diagram for Digikam::AlbumRootLocation:



### Public Member Functions

- **AlbumRootLocation** (const [AlbumRootInfo](#) &info)
- void **setAbsolutePath** (const QString &path)

- void **setCaseSensitivity** ([CollectionLocation::CaseSensitivity](#) c)
- void **setId** (int id)
- void **setLabel** (const QString &label)
- void **setStatus** ([CollectionLocation::Status](#) s)
- void **setStatusFromFlags** ()
- void **setType** ([Type](#) type)

### Public Member Functions inherited from [Digikam::CollectionLocation](#)

- QString **albumRootPath** () const
- Qt::CaseSensitivity **asQtCaseSensitivity** () const
- [CaseSensitivity](#) **caseSensitivity** () const
- size\_t **hash** () const
- int **id** () const
- bool **isAvailable** () const
- bool **isNull** () const
- QString **label** () const
- [Status](#) **status** () const
- [Type](#) **type** () const

### Public Attributes

- bool **available** = false
- bool **hidden** = false
- QString **specificPath**

### Public Attributes inherited from [Digikam::CollectionLocation](#)

- QString **identifier**

### Additional Inherited Members

### Public Types inherited from [Digikam::CollectionLocation](#)

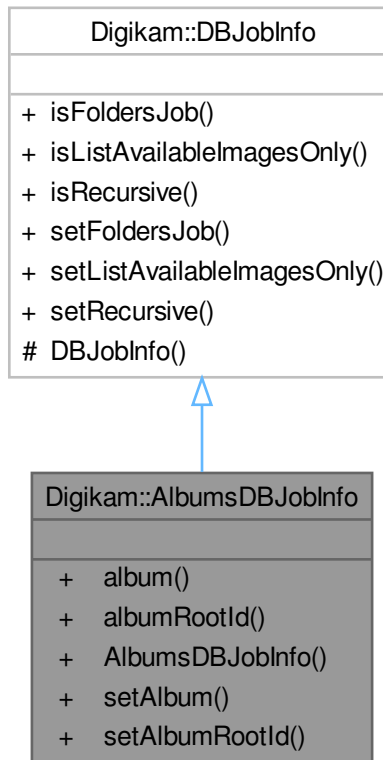
- enum [CaseSensitivity](#) { [UnknownCaseSensitivity](#) , [CaseInsensitive](#) , [CaseSensitive](#) }
- enum [Status](#) { [LocationNull](#) , [LocationAvailable](#) , [LocationHidden](#) , [LocationUnavailable](#) , [LocationDeleted](#) }
- enum [Type](#) { [Undefined](#) = 0 , [VolumeHardWired](#) = 1 , [VolumeRemovable](#) = 2 , [Network](#) = 3 }

### Protected Attributes inherited from [Digikam::CollectionLocation](#)

- [CaseSensitivity](#) **m\_caseSensitivity** = [UnknownCaseSensitivity](#)
- int **m\_id** = -1
- QString **m\_label**
- QString **m\_path**
- [Status](#) **m\_status** = [LocationNull](#)
- [Type](#) **m\_type** = [VolumeHardWired](#)

## 6.68 Digikam::AlbumsDBJobInfo Class Reference

Inheritance diagram for Digikam::AlbumsDBJobInfo:



### Public Member Functions

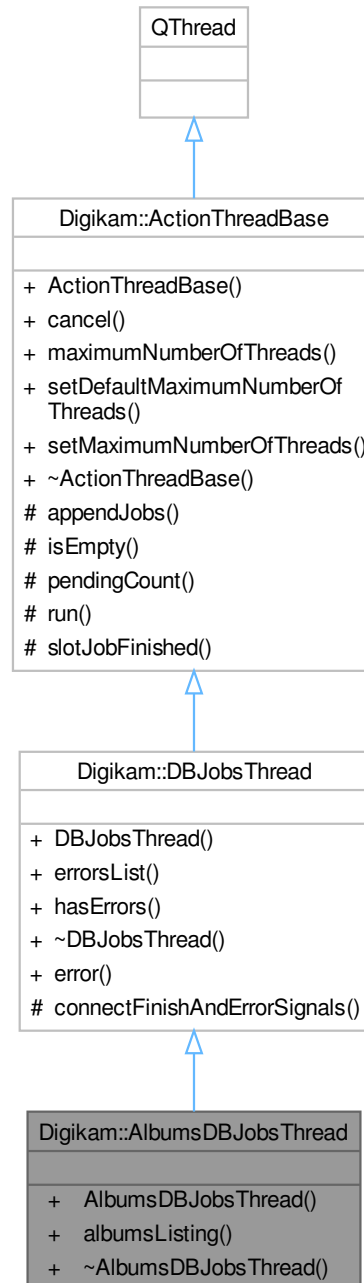
- `QString album ()`
- `int albumRootId ()`
- `void setAlbum (const QString &album)`
- `void setAlbumRootId (int id)`

### Public Member Functions inherited from [Digikam::DBJobInfo](#)

- `bool isFoldersJob () const`
- `bool isListAvailableImagesOnly () const`
- `bool isRecursive () const`
- `void setFoldersJob ()`
- `void setListAvailableImagesOnly ()`
- `void setRecursive ()`

## 6.69 Digikam::AlbumsDBJobsThread Class Reference

Inheritance diagram for Digikam::AlbumsDBJobsThread:



### Signals

- void **faceFoldersData** (const QMap< QString, QHash< int, int > > &)
- void **foldersData** (const QHash< int, int > &)



**Signals inherited from [Digikam::DBJobsThread](#)**

- void **data** (const QList< [ItemLISTERRecord](#) > &records)
- void **finished** ()

**Public Member Functions**

- **AlbumsDBJobsThread** (QObject \*const parent)
- void **albumsListing** (const [AlbumsDBJobInfo](#) &info)  
*Starts PAlbums listing and scanning job(s)*

**Public Member Functions inherited from [Digikam::DBJobsThread](#)**

- **DBJobsThread** (QObject \*const parent)
- QList< QString > & **errorsList** ()  
*A method to get all errors reported from jobs.*
- bool **hasErrors** ()  
*hasErrors: a method to check for jobs errors*

**Public Member Functions inherited from [Digikam::ActionThreadBase](#)**

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

**Additional Inherited Members****Public Slots inherited from [Digikam::DBJobsThread](#)**

- void **error** (const QString &errString)  
*Appends the error string to m\_errorsList.*

**Protected Slots inherited from [Digikam::ActionThreadBase](#)**

- void **slotJobFinished** ()

**Protected Member Functions inherited from [Digikam::DBJobsThread](#)**

- void **connectFinishAndErrorSignals** ([DBJob](#) \*const j)  
*Connects the signals of job to the signals of the thread.*

**Protected Member Functions inherited from [Digikam::ActionThreadBase](#)**

- void **appendJobs** (const [ActionJobCollection](#) &jobs)
- bool **isEmpty** () const
- int **pendingCount** () const
- void **run** () override

**6.69.1 Member Function Documentation****6.69.1.1 albumsListing()**

```
void Digikam::AlbumsDBJobsThread::albumsListing (
    const AlbumsDBJobInfo & info )
```

## Parameters

<i>info</i>	represents the albums job info
-------------	--------------------------------

## 6.70 Digikam::AlbumSelectComboBox Class Reference

Inheritance diagram for Digikam::AlbumSelectComboBox:



**Public Slots**

- void **hidePopup** () override
- virtual void **updateText** ()

**Public Member Functions**

- **AlbumSelectComboBox** (QWidget \*const parent=nullptr)
- QSortFilterProxyModel \* **filterModel** () const
- bool **isCheckable** () const
- **AbstractCheckableAlbumModel** \* **model** () const
- void **setAlbumModels** (**AbstractCheckableAlbumModel** \*model, **AlbumFilterModel** \*filterModel=nullptr)
- void **selectAllSelectedText** (bool all)
- void **setCheckable** (bool checkable)
- void **setCloseOnActivate** (bool close)
- void **setDefaultAlbumModel** ()
- void **setDefaultTagModel** ()
- void **setNoSelectionText** (const QString &text)
- void **setRecursive** (bool recursive)
- void **setShowCheckStateSummary** (bool show)

**Public Member Functions inherited from Digikam::TreeViewLineEditComboBox**

- void **installView** (QAbstractItemView \*view=nullptr) override
- void **setLineEdit** (QLineEdit \*edit)
- void **setLineEditText** (const QString &text)
- **TreeViewLineEditComboBox** (QWidget \*const parent=nullptr)

**Public Member Functions inherited from Digikam::TreeViewComboBox**

- **TreeViewComboBox** (QWidget \*parent=nullptr)
- QTreeView \* **view** () const

**Public Member Functions inherited from Digikam::StayPoppedUpComboBox**

- **StayPoppedUpComboBox** (QWidget \*const parent=nullptr)

**Public Member Functions inherited from Digikam::ModelIndexBasedComboBox**

- QModelIndex **currentIndex** () const
- void **hidePopup** () override
- **ModelIndexBasedComboBox** (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

**Protected Member Functions**

- void **installView** (QAbstractItemView \*view=nullptr) override

### Protected Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- virtual void [installLineEdit](#) ()

### Protected Member Functions inherited from [Digikam::TreeViewComboBox](#)

- void [sendViewportEventToView](#) (QEvent \*e) override

### Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool [eventFilter](#) (QObject \*watched, QEvent \*event) override
- void [installView](#) (QAbstractItemView \*view)

### Additional Inherited Members

### Protected Attributes inherited from [Digikam::TreeViewLineEditComboBox](#)

- QLineEdit \* [m\\_comboLineEdit](#) = nullptr

### Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

- QAbstractItemView \* [m\\_view](#) = nullptr

### Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex [m\\_currentIndex](#)

## 6.70.1 Member Function Documentation

### 6.70.1.1 [filterModel\(\)](#)

```
QSortFilterProxyModel * Digikam::AlbumSelectComboBox::filterModel ( ) const
```

Return the filter model in use.

### 6.70.1.2 [installView\(\)](#)

```
void Digikam::AlbumSelectComboBox::installView (
    QAbstractItemView * view = nullptr ) [override], [protected], [virtual]
```

Replace the standard combo box list view with a QTreeView. Call this after installing an appropriate model.

Reimplemented from [Digikam::TreeViewComboBox](#).

### 6.70.1.3 model()

```
AbstractCheckableAlbumModel * Digikam::AlbumSelectComboBox::model ( ) const
```

Returns the source model. Retrieve selection information from here.

### 6.70.1.4 setAllSelectedText()

```
void Digikam::AlbumSelectComboBox::setAllSelectedText (
    bool all )
```

Enable or disable the text used to describe the status when all album is selected.

### 6.70.1.5 setCheckable()

```
void Digikam::AlbumSelectComboBox::setCheckable (
    bool checkable )
```

Enable checkboxes next to the items. Default: true

### 6.70.1.6 setCloseOnActivate()

```
void Digikam::AlbumSelectComboBox::setCloseOnActivate (
    bool close )
```

Enable closing when an item was activated (clicked). Default: false.

### 6.70.1.7 setDefaultAlbumModel()

```
void Digikam::AlbumSelectComboBox::setDefaultAlbumModel ( )
```

Once after creation, call one of these three methods. Use the first one if you want a standard combo box for PAlbums and the second one for tags, while the third allows you to provide custom source and filter models. The first two also set a default noSelectionText. Customize afterwards if required.

### 6.70.1.8 setNoSelectionText()

```
void Digikam::AlbumSelectComboBox::setNoSelectionText (
    const QString & text )
```

Sets the text that is used to describe the state when no album is selected. This may be something like "Any album" or "No tag selected". Depends on the default line edit implementation of [TreeViewLineEditComboBox](#).

### 6.70.1.9 setRecursive()

```
void Digikam::AlbumSelectComboBox::setRecursive (
    bool recursive )
```

If all subalbums shall be selected when parent will be selected

### 6.70.1.10 setShowCheckStateSummary()

```
void Digikam::AlbumSelectComboBox::setShowCheckStateSummary (
    bool show )
```

If the box is checkable, enable showing a resume a la "3 Albums checked" in the combo box text. Default: True

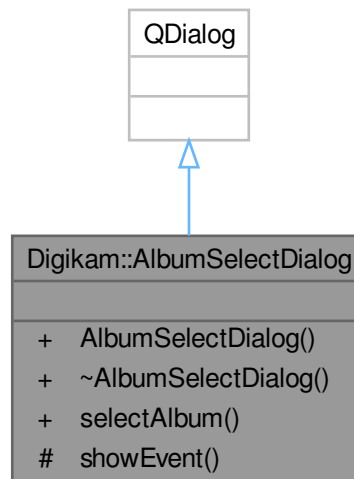
### 6.70.1.11 updateText

```
void Digikam::AlbumSelectComboBox::updateText ( ) [virtual], [slot]
```

Updates the text describing the selection ("3 Albums selected"). Can be overridden to customize the default text.

## 6.71 Digikam::AlbumSelectDialog Class Reference

Inheritance diagram for Digikam::AlbumSelectDialog:



### Public Member Functions

- **AlbumSelectDialog** (`QWidget *const parent`, `PAAlbum *const albumToSelect`, `const QString &header=QString()`)

### Static Public Member Functions

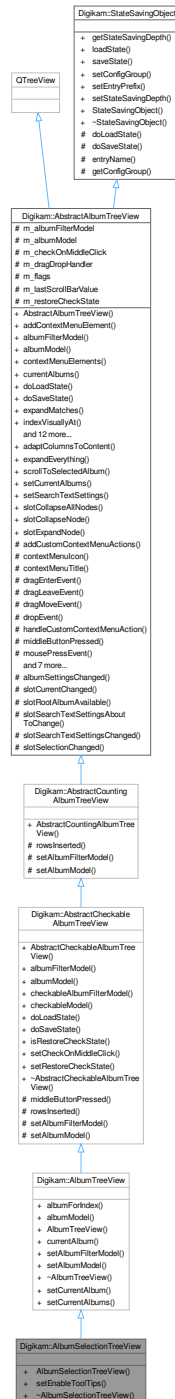
- static `PAAlbum * selectAlbum` (`QWidget *const parent`, `PAAlbum *const albumToSelect`, `const QString &header=QString()`)

Protected Member Functions

- void **showEvent** (QShowEvent \*) override

## 6.72 Digikam::AlbumSelectionTreeView Class Reference

Inheritance diagram for Digikam::AlbumSelectionTreeView:



## Signals

- void [signalFindDuplicates](#) (const QList< [PAlbum](#) \* > &albums)

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< [Album](#) \* > &selectedAlbums)

## Public Member Functions

- **[AlbumSelectionTreeView](#)** ([QWidget](#) \*const parent, [AlbumModel](#) \*const model, [AlbumModificationHelper](#) \*const albumModificationHelper)
- void [setEnabledToolTips](#) (bool enable)

## Public Member Functions inherited from [Digikam::AlbumTreeView](#)

- [PAlbum](#) \* **[albumForIndex](#)** (const [QModelIndex](#) &index) const
- [AlbumModel](#) \* **[albumModel](#)** () const
- **[AlbumTreeView](#)** ([QWidget](#) \*const parent=nullptr, [Flags](#) flags=DefaultFlags)
- [PAlbum](#) \* **[currentAlbum](#)** () const
- void **[setAlbumFilterModel](#)** ([CheckableAlbumFilterModel](#) \*const filterModel)
- void **[setAlbumModel](#)** ([AlbumModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- **[AbstractCheckableAlbumTreeView](#)** ([QWidget](#) \*const parent, [Flags](#) flags)
- [CheckableAlbumFilterModel](#) \* **[albumFilterModel](#)** () const
- [AbstractCheckableAlbumModel](#) \* **[albumModel](#)** () const
- [CheckableAlbumFilterModel](#) \* **[checkableAlbumFilterModel](#)** () const
- [AbstractCheckableAlbumModel](#) \* **[checkableModel](#)** () const
- void **[doLoadState](#)** () override
- void **[doSaveState](#)** () override
- bool **[isRestoreCheckState](#)** () const
- void **[setCheckOnMiddleClick](#)** (bool doThat)
- void **[setRestoreCheckState](#)** (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- **[AbstractCountingAlbumTreeView](#)** ([QWidget](#) \*const parent, [Flags](#) flags)



## Public Member Functions inherited from Digikam::AbstractAlbumTreeView

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from Digikam::StateSavingObject

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Additional Inherited Members

## Public Types inherited from Digikam::AbstractAlbumTreeView

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from Digikam::StateSavingObject

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots inherited from Digikam::AlbumTreeView

- void **setCurrentAlbum** (int albumId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()
  - slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()
  - slotExpandNode - expands recursively selected nodes*

## Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void [middleButtonPressed](#) (Album \*a) override
- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

## Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

## Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, Album \*album)
- virtual QPixmap [contextMenuIcon](#) () const
- virtual QString [contextMenuTitle](#) () const
- void [dragEnterEvent](#) (QDragEnterEvent \*e) override
- void [dragLeaveEvent](#) (QDragLeaveEvent \*e) override
- void [dragMoveEvent](#) (QDragMoveEvent \*e) override
- void [dropEvent](#) (QDropEvent \*e) override
- virtual void [handleCustomContextMenuAction](#) (QAction \*action, const [AlbumPointer](#)< Album > &album)
- void [mousePressEvent](#) (QMouseEvent \*e) override
  - Other helper methods.*
- virtual QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)
- void [rowsAboutToBeRemoved](#) (const QModelIndex &parent, int start, int end) override
- void [rowsInserted](#) (const QModelIndex &index, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
- virtual bool [showContextMenuAt](#) (QContextMenuEvent \*event, Album \*albumForEvent)
- void [startDrag](#) (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from Digikam::StateSavingObject

- QString `entryName` (const QString &base) const
- KConfigGroup `getConfigGroup` () const

## Protected Attributes inherited from Digikam::AbstractAlbumTreeView

- AlbumFilterModel \* `m_albumFilterModel` = nullptr
- AbstractSpecificAlbumModel \* `m_albumModel` = nullptr
- bool `m_checkOnMiddleClick` = false
- AlbumModelDragDropHandler \* `m_dragDropHandler` = nullptr
- Flags `m_flags` = DefaultFlags
- int `m_lastScrollBarValue` = 0
- bool `m_restoreCheckState` = false

### 6.72.1 Detailed Description

Album tree view used in the left sidebar to select PAlbums and perform operations on them via a context menu.

#### Author

jwienke

### 6.72.2 Member Function Documentation

#### 6.72.2.1 setEnableToolTips()

```
void Digikam::AlbumSelectionTreeView::setEnableToolTips (
    bool enable )
```

Sets whether this widget shall display tool tips or not.

#### 6.72.2.2 signalFindDuplicates

```
void Digikam::AlbumSelectionTreeView::signalFindDuplicates (
    const QList< PAlbum * > & albums ) [signal]
```

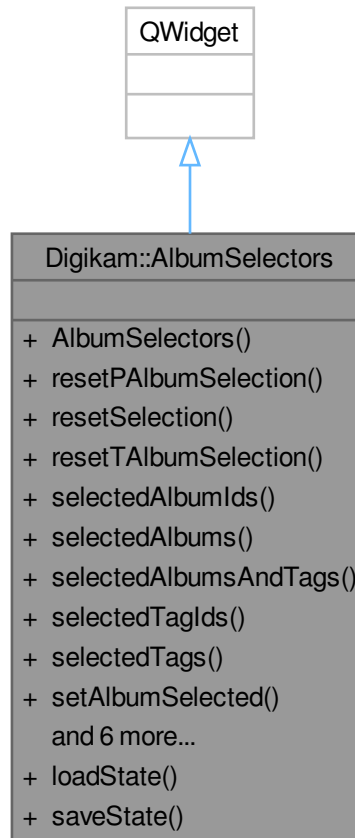
Emitted if a find duplicates search shall be invoked on the given album.

#### Parameters

<i>albums</i>	the album to find duplicates in
---------------	---------------------------------

## 6.73 Digikam::AlbumSelectors Class Reference

Inheritance diagram for Digikam::AlbumSelectors:



### Public Types

- enum **AlbumType** { **PhysAlbum** = 0 , **TagsAlbum** , **All** }
- enum **SelectionType** { **SingleSelection** = 0 , **MultipleSelection** }

### Public Slots

- void [loadState](#) ()
- void [saveState](#) ()

### Signals

- void **signalSelectionChanged** ()

## Public Member Functions

- [AlbumSelectors](#) (const QString &label, const QString &configName, QWidget \*const parent=nullptr, AlbumType albumType=All, bool allowRecursive=false)
- void [resetPAlbumSelection](#) ()
- void [resetSelection](#) ()
- void [resetTAlbumSelection](#) ()
- QList< int > [selectedAlbumIds](#) () const
- AlbumList [selectedAlbums](#) () const
- AlbumList [selectedAlbumsAndTags](#) () const
- QList< int > [selectedTagIds](#) () const
- AlbumList [selectedTags](#) () const
- void [setAlbumSelected](#) (Album \*const album, SelectionType type)
- void [setTagSelected](#) (Album \*const album, SelectionType type)
- void [setTypeSelection](#) (int albumType)
- int [typeSelection](#) () const
- bool [wholeAlbumsChecked](#) () const
- bool [wholeTagsChecked](#) () const

## 6.73.1 Constructor & Destructor Documentation

### 6.73.1.1 AlbumSelectors()

```
Digikam::AlbumSelectors::AlbumSelectors (
    const QString & label,
    const QString & configName,
    QWidget *const parent = nullptr,
    AlbumType albumType = All,
    bool allowRecursive = false ) [explicit]
```

Default Constructor. 'label' is front text of label which title widget. 'configName' is name used to store Albums configuration in settings file. 'parent' is parent widget.

## 6.73.2 Member Function Documentation

### 6.73.2.1 loadState

```
void Digikam::AlbumSelectors::loadState ( ) [slot]
```

Called in constructor. Restore previous settings saved in configuration file.

### 6.73.2.2 resetPAlbumSelection()

```
void Digikam::AlbumSelectors::resetPAlbumSelection ( )
```

Reset all Physical Albums selection.

### 6.73.2.3 resetSelection()

```
void Digikam::AlbumSelectors::resetSelection ( )
```

Reset all Physical and Tag Albums selection.

### 6.73.2.4 resetTAlbumSelection()

```
void Digikam::AlbumSelectors::resetTAlbumSelection ( )
```

Reset all Tag Albums selection.

### 6.73.2.5 saveState

```
void Digikam::AlbumSelectors::saveState ( ) [slot]
```

Save settings in configuration file. Must be called explicitly by host implementation.

### 6.73.2.6 selectedAlbumIds()

```
QList< int > Digikam::AlbumSelectors::selectedAlbumIds ( ) const
```

Return list of selected physical album ids

### 6.73.2.7 selectedAlbums()

```
AlbumList Digikam::AlbumSelectors::selectedAlbums ( ) const
```

Return list of selected physical albums

### 6.73.2.8 selectedAlbumsAndTags()

```
AlbumList Digikam::AlbumSelectors::selectedAlbumsAndTags ( ) const
```

Return list of selected physical and tag albums.

### 6.73.2.9 selectedTagIds()

```
QList< int > Digikam::AlbumSelectors::selectedTagIds ( ) const
```

Return list of selected tag album ids

### 6.73.2.10 selectedTags()

```
AlbumList Digikam::AlbumSelectors::selectedTags ( ) const
```

Return list of selected tag albums

### 6.73.2.11 setAlbumSelected()

```
void Digikam::AlbumSelectors::setAlbumSelected (
    Album *const album,
    SelectionType type )
```

Select Physical [Album](#) from list. If `singleSelection` is true, only this one is selected from tree-view and all others are deselected.

### 6.73.2.12 setTagSelected()

```
void Digikam::AlbumSelectors::setTagSelected (
    Album *const album,
    SelectionType type )
```

Select Tag [Album](#) from list. If `singleSelection` is true, only this one is selected from tree-view and all others are deselected.

### 6.73.2.13 setTypeSelection()

```
void Digikam::AlbumSelectors::setTypeSelection (
    int albumType )
```

Sets the search type selection with the `AlbumType`.

### 6.73.2.14 typeSelection()

```
int Digikam::AlbumSelectors::typeSelection ( ) const
```

Returns the selected album type.

### 6.73.2.15 wholeAlbumsChecked()

```
bool Digikam::AlbumSelectors::wholeAlbumsChecked ( ) const
```

Return true if whole Albums collection option is checked.

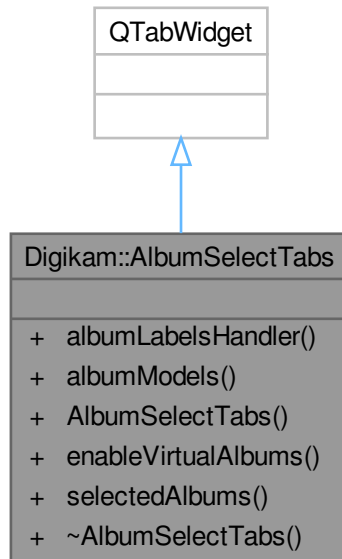
### 6.73.2.16 wholeTagsChecked()

```
bool Digikam::AlbumSelectors::wholeTagsChecked ( ) const
```

Return true if whole Tags collection option is checked.

## 6.74 Digikam::AlbumSelectTabs Class Reference

Inheritance diagram for Digikam::AlbumSelectTabs:



### Signals

- void `signalAlbumSelectionChanged ()`

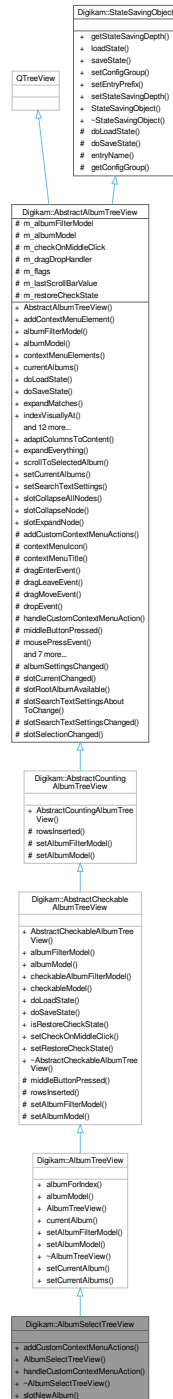
### Public Member Functions

- `AlbumLabelsSearchHandler * albumLabelsHandler () const`
- `QList< AbstractCheckableAlbumModel * > albumModels () const`
- `AlbumSelectTabs (const QString &name, QWidget *const parent=nullptr)`
- void `enableVirtualAlbums (bool flag=true)`
- `AlbumList selectedAlbums () const`



## 6.75 Digikam::AlbumSelectTreeView Class Reference

Inheritance diagram for Digikam::AlbumSelectTreeView:



### Public Slots

- void `slotNewAlbum()`

## Public Slots inherited from [Digikam::AlbumTreeView](#)

- void **setCurrentAlbum** (int albumId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)
- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)
- void **slotCollapseAllNodes** ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void **slotCollapseNode** ()
  - slotCollapseNode - collapse recursively selected nodes*
- void **slotExpandNode** ()
  - slotExpandNode - expands recursively selected nodes*

## Public Member Functions

- void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override
- [AlbumSelectTreeView](#) ([AlbumModel](#) \*const model, [AlbumModificationHelper](#) \*const albumModificationHelper, QWidget \*const parent=nullptr)
- void [handleCustomContextMenuAction](#) (QAction \*action, const [AlbumPointer](#)< [Album](#) > &album) override
- [~AlbumSelectTreeView](#) () override

## Public Member Functions inherited from [Digikam::AlbumTreeView](#)

- [PAlbum](#) \* **albumForIndex** (const QModelIndex &index) const
- [AlbumModel](#) \* **albumModel** () const
- [AlbumTreeView](#) (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)
- [PAlbum](#) \* **currentAlbum** () const
- void **setAlbumFilterModel** ([CheckableAlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AlbumModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- [CheckableAlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractCheckableAlbumModel](#) \* **albumModel** () const
- [CheckableAlbumFilterModel](#) \* **checkableAlbumFilterModel** () const
- [AbstractCheckableAlbumModel](#) \* **checkableModel** () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< [Album](#) \* > &selectedAlbums)

## Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

## Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

## Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual QPixmap [contextMenuIcon](#) () const
  - virtual QString [contextMenuTitle](#) () const
  - void [dragEnterEvent](#) (QDragEnterEvent \*e) override
  - void [dragLeaveEvent](#) (QDragLeaveEvent \*e) override
  - void [dragMoveEvent](#) (QDragMoveEvent \*e) override
  - void [dropEvent](#) (QDropEvent \*e) override
  - void [mousePressEvent](#) (QMouseEvent \*e) override
- Other helper methods.*
- virtual QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)
  - void [rowsAboutToBeRemoved](#) (const QModelIndex &parent, int start, int end) override
  - void [rowsInserted](#) (const QModelIndex &index, int start, int end) override
  - void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
  - void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
  - virtual bool [showContextMenuAt](#) (QContextMenuEvent \*event, [Album](#) \*albumForEvent)
  - void [startDrag](#) (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* `m_albumFilterModel` = nullptr
- [AbstractSpecificAlbumModel](#) \* `m_albumModel` = nullptr
- bool `m_checkOnMiddleClick` = false
- [AlbumModelDragDropHandler](#) \* `m_dragDropHandler` = nullptr
- Flags `m_flags` = DefaultFlags
- int `m_lastScrollBarValue` = 0
- bool `m_restoreCheckState` = false

### 6.75.1 Detailed Description

Enables a simple context menu only for creating a new album.

Author

jwienke

### 6.75.2 Constructor & Destructor Documentation

#### 6.75.2.1 AlbumSelectTreeView()

```
Digikam::AlbumSelectTreeView::AlbumSelectTreeView (
    AlbumModel *const model,
    AlbumModificationHelper *const albumModificationHelper,
    QWidget *const parent = nullptr )
```

Constructor.

Parameters

<i>model</i>	album model to work with
<i>albumModificationHelper</i>	helper object for modifying albums
<i>parent</i>	the parent for Qt's parent child mechanism

#### 6.75.2.2 ~AlbumSelectTreeView()

```
Digikam::AlbumSelectTreeView::~AlbumSelectTreeView ( ) [override]
```

Destructor.

### 6.75.3 Member Function Documentation

#### 6.75.3.1 addCustomContextMenuActions()

```
void Digikam::AlbumSelectTreeView::addCustomContextMenuActions (
    ContextMenuHelper & cmh,
    Album * album ) [override], [virtual]
```

Hook method to add custom actions to the generated context menu.

## Parameters

<i>cmh</i>	helper object to create the context menu
<i>album</i>	tag on which the context menu will be created. May be null if it is requested on no tag entry

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

**6.75.3.2 handleCustomContextMenuAction()**

```
void Digikam::AlbumSelectTreeView::handleCustomContextMenuAction (
    QAction * action,
    const AlbumPointer< Album > & album ) [override], [virtual]
```

Hook method to handle the custom context menu actions that were added with addCustomContextMenuActions.

## Parameters

<i>action</i>	the action that was chosen by the user, may be null if none of the custom actions were selected
<i>album</i>	the tag on which the context menu was requested. May be null if there was no

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

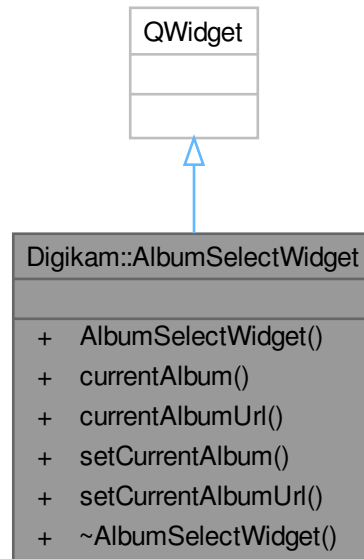
**6.75.3.3 slotNewAlbum**

```
void Digikam::AlbumSelectTreeView::slotNewAlbum ( ) [slot]
```

Shows a dialog to create a new album under the selected album in this view.

## 6.76 Digikam::AlbumSelectWidget Class Reference

Inheritance diagram for Digikam::AlbumSelectWidget:



### Signals

- void `completerActivated` ()
- void `itemSelectionChanged` ()

### Public Member Functions

- `AlbumSelectWidget` (`QWidget *const parent=nullptr`, `PAAlbum *const albumToSelect=nullptr`, `bool completerSelect=false`)
- `PAAlbum * currentAlbum` () const
- `QUrl currentAlbumUrl` () const
- void `setCurrentAlbum` (`PAAlbum *const albumToSelect`)
- void `setCurrentAlbumUrl` (`const QUrl &albumUrl`)

## 6.77 Digikam::AlbumShortInfo Class Reference

### Public Member Functions

- bool `isNull` () const

**Public Attributes**

- int **albumRootId** = 0
- int **id** = 0
- QString **relativePath**

## 6.78 Digikam::AlbumSimplified Class Reference

**Public Member Functions**

- **AlbumSimplified** (const QString &title)
- **AlbumSimplified** (const QString &title, bool uploadable)

**Public Attributes**

- QStringList **childrenIDs**
- QString **title**
- bool **uploadable** = true

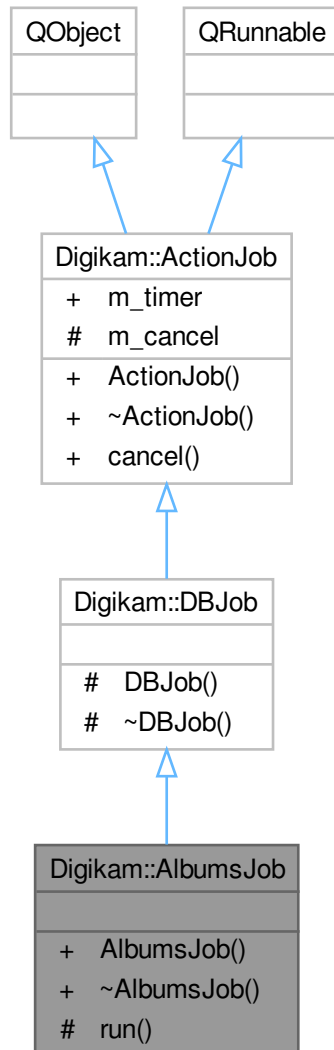
### 6.78.1 Detailed Description

This class is used when parsing response of listAlbums(). It contains only the most important attributes of an album, which is needed for further usage (e.g upload photos, create new album).



## 6.79 Digikam::AlbumsJob Class Reference

Inheritance diagram for Digikam::AlbumsJob:



### Signals

- void **foldersData** (const QHash< int, int > &)

### Signals inherited from [Digikam::DBJob](#)

- void **data** (const QList< [ItemListerRecord](#) > &records)
- void **error** (const QString &err)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Member Functions

- **AlbumsJob** (const [AlbumsDBJobInfo](#) &jobInfo)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void **run** () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

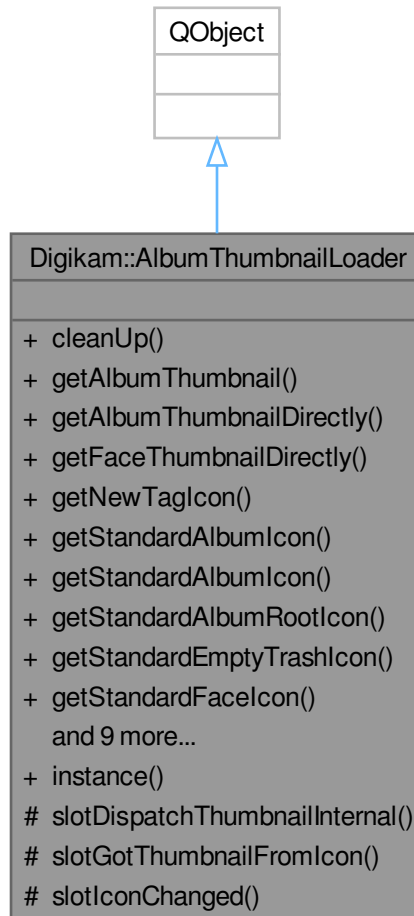
- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.80 Digikam::AlbumThumbnailLoader Class Reference

Inheritance diagram for Digikam::AlbumThumbnailLoader:



### Public Types

- enum [RelativeSize](#) { **NormalSize** , **SmallerSize** }

### Signals

- void [signalDispatchThumbnailInternal](#) (int albumID, const QPixmap &thumbnail)
- void [signalFailed](#) (Album \*album)
- void [signalReloadThumbnails](#) ()
- void [signalThumbnail](#) (Album \*album, const QPixmap &)

## Public Member Functions

- void **cleanUp** ()
- bool **getAlbumThumbnail** (PAlbum \*const album)
- QPixmap **getAlbumThumbnailDirectly** (PAAlbum \*const album)
- QPixmap **getFaceThumbnailDirectly** (TAlbum \*const album)
- QPixmap **getNewTagIcon** (RelativeSize size=NormalSize)
- QPixmap **getStandardAlbumIcon** (PAAlbum \*const album, RelativeSize size=NormalSize)
- QPixmap **getStandardAlbumIcon** (RelativeSize size=NormalSize)
- QPixmap **getStandardAlbumRootIcon** (RelativeSize size=NormalSize)
- QPixmap **getStandardEmptyTrashIcon** (RelativeSize size=NormalSize)
- QPixmap **getStandardFacelIcon** (TAlbum \*const album, RelativeSize size=NormalSize)
- QPixmap **getStandardFullTrashIcon** (RelativeSize size=NormalSize)
- QPixmap **getStandardOfflineIcon** (RelativeSize size=NormalSize)
- QPixmap **getStandardTagIcon** (RelativeSize size=NormalSize)
- QPixmap **getStandardTagIcon** (TAlbum \*const album, RelativeSize size=NormalSize)
- QPixmap **getStandardTagRootIcon** (RelativeSize size=NormalSize)
- bool **getTagThumbnail** (TAlbum \*const album, QPixmap &icon)
- QPixmap **getTagThumbnailDirectly** (TAlbum \*const album)
- void **setThumbnailSize** (int size, int face)
- int **thumbnailSize** () const

## Static Public Member Functions

- static AlbumThumbnailLoader \* **instance** ()

## Protected Slots

- void **slotDispatchThumbnailInternal** (int albumID, const QPixmap &thumbnail)
- void **slotGotThumbnailFromIcon** (const LoadingDescription &loadingDescription, const QPixmap & pixmap)
- void **slotIconChanged** (Album \*album)

## Friends

- class AlbumThumbnailLoaderCreator

## 6.80.1 Member Enumeration Documentation

### 6.80.1.1 RelativeSize

```
enum Digikam::AlbumThumbnailLoader::RelativeSize
```

Album thumbnail size is configurable via the settings menu. Some widgets use smaller icons than other widgets. These widgets do not need to know the currently set icon size from the setup and calculate a smaller size, but can simply request a relatively smaller icon. Depending on the user-chosen icon size, this size may in fact not be smaller than the normal size.

## 6.80.2 Member Function Documentation

### 6.80.2.1 getAlbumThumbnail()

```
bool Digikam::AlbumThumbnailLoader::getAlbumThumbnail (
    PAlbum *const album )
```

Request thumbnail for given album. The thumbnail will be loaded and returned asynchronously by the signals. If no thumbnail is associated with given album, no action will be taken, and false is returned.

### 6.80.2.2 getAlbumThumbnailDirectly()

```
QPixmap Digikam::AlbumThumbnailLoader::getAlbumThumbnailDirectly (
    PAlbum *const album )
```

Request thumbnail for given album, with slightly different behavior than the above method: If the thumbnail is already available in the cache, it is returned. If the icon is not yet loaded, it will be returned asynchronously by the signals, and a default icon is returned here. If no icon is associated, the default icon is returned.

### 6.80.2.3 getFaceThumbnailDirectly()

```
QPixmap Digikam::AlbumThumbnailLoader::getFaceThumbnailDirectly (
    TAlbum *const album )
```

Loads face tag thumbnail, like [getTagThumbnailDirectly\(\)](#) but loads thumbnails in the size for faces

### 6.80.2.4 getStandardTagIcon()

```
QPixmap Digikam::AlbumThumbnailLoader::getStandardTagIcon (
    RelativeSize size = NormalSize )
```

Return standard tag and album icons. The third methods check if album is the root, and returns the standard icon or the root standard icon.

### 6.80.2.5 getTagThumbnail()

```
bool Digikam::AlbumThumbnailLoader::getTagThumbnail (
    TAlbum *const album,
    QPixmap & icon )
```

Behaves similar to the above method. Tag thumbnails will be processed as appropriate. Tags may have associated an icon that is loaded synchronously by the system icon loader. In this case, icon is set to this icon, and false is returned. If no icon is associated with the tag, icon is set to null, and false is returned. If a custom icon is associated with the tag, it is loaded asynchronously, icon is set to null, and true is returned. Tag thumbnails are always smaller than album thumbnails - as small as an album thumbnail with SmallerSize. They are supposed to be blended into the standard tag icon obtained below, or used as is when SmallerSize is requested anyway.

#### Returns

Returns true if icon is loaded asynchronously.

### 6.80.2.6 `getTagThumbnailDirectly()`

```
QPixmap Digikam::AlbumThumbnailLoader::getTagThumbnailDirectly (
    TAlbum *const album )
```

Loads tag thumbnail, with slightly different behavior than the above method: If the thumbnail is already available in the cache, it is returned, already blended with the standard icon, if requested. If the icon is not yet loaded, it will be returned asynchronously by the signals (unblended), and a default icon is returned here. If no icon is associated, the default icon is returned.

### 6.80.2.7 `instance()`

```
AlbumThumbnailLoader * Digikam::AlbumThumbnailLoader::instance ( ) [static]
```

Return a preview of physical album directly without to use cache. Size of image can be passed as argument.

### 6.80.2.8 `setThumbnailSize()`

```
void Digikam::AlbumThumbnailLoader::setThumbnailSize (
    int size,
    int face )
```

Change the size of the thumbnails. If the size differs from the current size, `signalReloadThumbnails` will be emitted.

### 6.80.2.9 `signalDispatchThumbnailInternal`

```
void Digikam::AlbumThumbnailLoader::signalDispatchThumbnailInternal (
    int albumID,
    const QPixmap & thumbnail ) [signal]
```

Internal signal to dispatch `Album` thumbnail change.

### 6.80.2.10 `signalFailed`

```
void Digikam::AlbumThumbnailLoader::signalFailed (
    Album * album ) [signal]
```

This signal is emitted if thumbnail generation for given album failed. Same considerations as above.

### 6.80.2.11 `signalReloadThumbnails`

```
void Digikam::AlbumThumbnailLoader::signalReloadThumbnails ( ) [signal]
```

Indicates that all album and tag thumbnails need to be reloaded. This is usually because the icon size has changed in the setup.

### 6.80.2.12 signalThumbnail

```
void Digikam::AlbumThumbnailLoader::signalThumbnail (
    Album * album,
    const QPixmap & ) [signal]
```

This signal is emitted as soon as a thumbnail has become available for given album. This class is a singleton, so any object connected to this signal might not actually have requested a thumbnail for given url

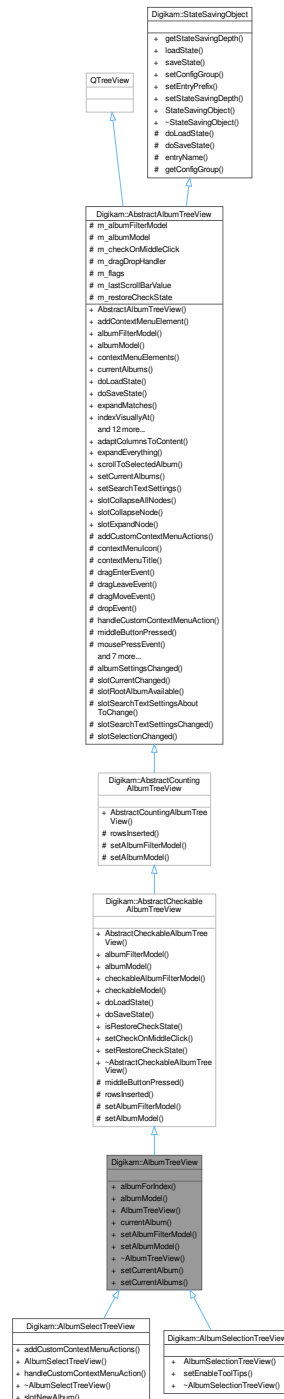
### 6.80.2.13 thumbnailSize()

```
int Digikam::AlbumThumbnailLoader::thumbnailSize ( ) const
```

Get the current default icon size

## 6.81 Digikam::AlbumTreeView Class Reference

Inheritance diagram for Digikam::AlbumTreeView:



### Public Slots

- void **setCurrentAlbum** (int albumId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< Album \* > &albums, bool selectInAlbumManager=true)



## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()
  - slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()
  - slotExpandNode - expands recursively selected nodes*

## Public Member Functions

- [Album](#) \* [albumForIndex](#) (const QModelIndex &index) const
- [AlbumModel](#) \* [albumModel](#) () const
- [AlbumTreeView](#) (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)
- [Album](#) \* [currentAlbum](#) () const
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AlbumModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- [CheckableAlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [albumModel](#) () const
- [CheckableAlbumFilterModel](#) \* [checkableAlbumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [checkableModel](#) () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void **currentAlbumChanged** ([Album](#) \*currentAlbum)
- void **selectedAlbumsChanged** (const QList< [Album](#) \* > &selectedAlbums)

### Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void **albumSettingsChanged** ()
- void **slotCurrentChanged** ()
- virtual void **slotRootAlbumAvailable** ()
- void **slotSearchTextSettingsAboutToChange** (bool searched, bool willSearch)
- void **slotSearchTextSettingsChanged** (bool wasSearching, bool searching)
- void **slotSelectionChanged** ()

### Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void **middleButtonPressed** (Album \*a) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **setAlbumFilterModel** (CheckableAlbumFilterModel \*const filterModel)
- void **setAlbumModel** (AbstractCheckableAlbumModel \*const model)

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **setAlbumFilterModel** (AlbumFilterModel \*const filterModel)
- void **setAlbumModel** (AbstractCountingAlbumModel \*const model)

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual void **addCustomContextMenuActions** (ContextMenuHelper &cmh, Album \*album)
- virtual QPixmap **contextMenuIcon** () const
- virtual QString **contextMenuTitle** () const
- void **dragEnterEvent** (QDragEnterEvent \*e) override
- void **dragLeaveEvent** (QDragLeaveEvent \*e) override
- void **dragMoveEvent** (QDragMoveEvent \*e) override
- void **dropEvent** (QDropEvent \*e) override
- virtual void **handleCustomContextMenuAction** (QAction \*action, const AlbumPointer< Album > &album)
- void **mousePressEvent** (QMouseEvent \*e) override

*Other helper methods.*

- virtual QPixmap  **pixmapForDrag** (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &index, int start, int end) override
- void **setAlbumFilterModel** (AlbumFilterModel \*const filterModel)
- void **setAlbumModel** (AbstractSpecificAlbumModel \*const model)
- virtual bool **showContextMenuAt** (QContextMenuEvent \*event, Album \*albumForEvent)
- void **startDrag** (Qt::DropActions supportedActions) override

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

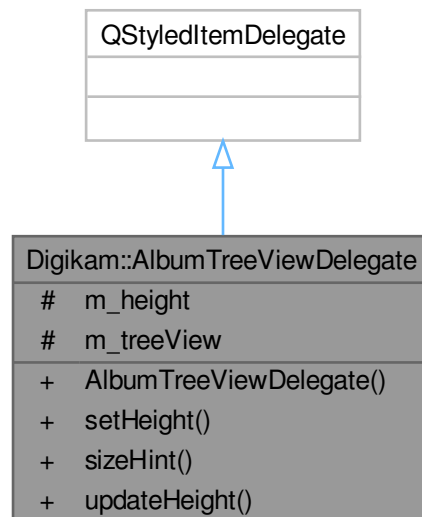
- QString **entryName** (const QString &base) const
- KConfigGroup **getConfigGroup** () const

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* **m\_albumFilterModel** = nullptr
- [AbstractSpecificAlbumModel](#) \* **m\_albumModel** = nullptr
- bool **m\_checkOnMiddleClick** = false
- [AlbumModelDragDropHandler](#) \* **m\_dragDropHandler** = nullptr
- Flags **m\_flags** = DefaultFlags
- int **m\_lastScrollBarValue** = 0
- bool **m\_restoreCheckState** = false

## 6.82 Digikam::AlbumTreeViewDelegate Class Reference

Inheritance diagram for Digikam::AlbumTreeViewDelegate:



### Public Member Functions

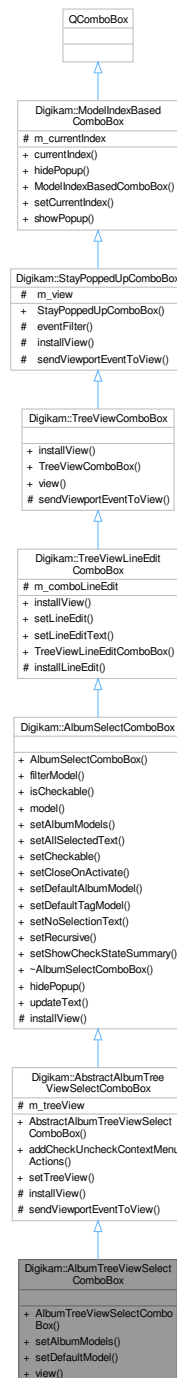
- **AlbumTreeViewDelegate** ([AbstractAlbumTreeView](#) \*const treeView=nullptr)
- void **setHeight** (int height)
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- void **updateHeight** ()

### Protected Attributes

- int **m\_height** = 0
- [AbstractAlbumTreeView](#) \* **m\_treeView** = nullptr

## 6.83 Digikam::AlbumTreeViewSelectComboBox Class Reference

Inheritance diagram for Digikam::AlbumTreeViewSelectComboBox:



### Public Member Functions

- **AlbumTreeViewSelectComboBox** (QWidget \*const parent=nullptr)
- void **setAlbumModels** (AlbumModel \*model, CheckableAlbumFilterModel \*filterModel=nullptr)
- void **setDefaultModel** ()
- AlbumTreeView \* **view** () const

### Public Member Functions inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- [AbstractAlbumTreeViewSelectComboBox](#) (QWidget \*const parent=nullptr)
- void [addCheckUncheckContextMenuActions](#) ()
- void [setTreeView](#) ([AbstractAlbumTreeView](#) \*const treeView)

### Public Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- [AlbumSelectComboBox](#) (QWidget \*const parent=nullptr)
- QSortFilterProxyModel \* [filterModel](#) () const
- bool [isCheckable](#) () const
- [AbstractCheckableAlbumModel](#) \* [model](#) () const
- void [setAlbumModels](#) ([AbstractCheckableAlbumModel](#) \*model, [AlbumFilterModel](#) \*filterModel=nullptr)
- void [selectAllSelectedText](#) (bool all)
- void [setCheckable](#) (bool checkable)
- void [setCloseOnActivate](#) (bool close)
- void [setDefaultAlbumModel](#) ()
- void [setDefaultTagModel](#) ()
- void [setNoSelectionText](#) (const QString &text)
- void [setRecursive](#) (bool recursive)
- void [setShowCheckStateSummary](#) (bool show)

### Public Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- void [installView](#) (QAbstractItemView \*view=nullptr) override
- void [setLineEdit](#) (QLineEdit \*edit)
- void [setLineEditText](#) (const QString &text)
- [TreeViewLineEditComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::TreeViewComboBox](#)

- [TreeViewComboBox](#) (QWidget \*parent=nullptr)
- QTreeView \* [view](#) () const

### Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex [currentIndex](#) () const
- void [hidePopup](#) () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void [setCurrentIndex](#) (const QModelIndex &index)
- void [showPopup](#) () override

## Additional Inherited Members

### Public Slots inherited from [Digikam::AlbumSelectComboBox](#)

- void `hidePopup` () override
- virtual void `updateText` ()

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- void `installView` (QAbstractItemView \*`view`=nullptr) override
- void `sendViewportEventToView` (QEvent \*`e`) override

### Protected Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- void `installView` (QAbstractItemView \*`view`=nullptr) override

### Protected Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- virtual void `installLineEdit` ()

### Protected Member Functions inherited from [Digikam::TreeViewComboBox](#)

- void `sendViewportEventToView` (QEvent \*`e`) override

### Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool `eventFilter` (QObject \*`watched`, QEvent \*`event`) override
- void `installView` (QAbstractItemView \*`view`)

### Protected Attributes inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- [AbstractAlbumTreeView](#) \* `m_treeView` = nullptr

### Protected Attributes inherited from [Digikam::TreeViewLineEditComboBox](#)

- QLineEdit \* `m_comboLineEdit` = nullptr

### Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

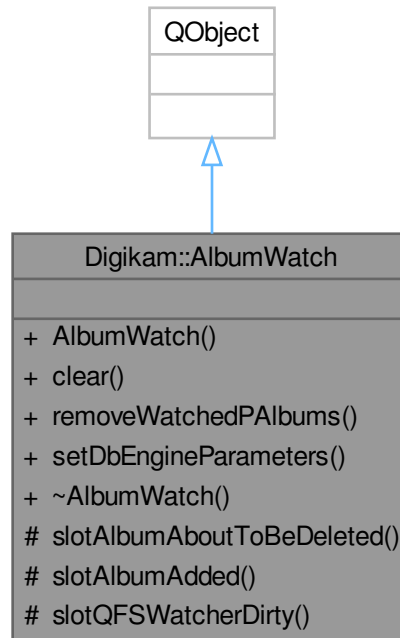
- QAbstractItemView \* `m_view` = nullptr

### Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex `m_currentIndex`

## 6.84 Digikam::AlbumWatch Class Reference

Inheritance diagram for Digikam::AlbumWatch:



### Public Member Functions

- **AlbumWatch** ([AlbumManager](#) \*const parent=nullptr)
- void **clear** ()
- void **removeWatchedPAIbums** (const [PAIbum](#) \*const album)
- void **setDbEngineParameters** (const [DbEngineParameters](#) &params)

### Protected Slots

- void **slotAlbumAboutToBeDeleted** ([Album](#) \*album)
- void **slotAlbumAdded** ([Album](#) \*album)
- void **slotQFSWatcherDirty** (const QString &path)



## 6.85 Digikam::AltLangStrEdit Class Reference

Inheritance diagram for Digikam::AltLangStrEdit:



### Classes

- class [Private](#)

### Public Slots

- void [slotEnabledInternalWidgets](#) (bool)

## Signals

- void [signalModified](#) (const QString &lang, const QString &text)
- void [signalSelectionChanged](#) (const QString &lang)
- void [signalValueAdded](#) (const QString &lang, const QString &text)
- void [signalValueDeleted](#) (const QString &lang)

## Public Member Functions

- void [addCurrent](#) ()
- [AltLangStrEdit](#) (QWidget \*const parent, unsigned int lines=3)
- bool [asDefaultAltLang](#) () const
- QString [currentLanguageCode](#) () const
- QString [defaultAltLang](#) () const
- QString [languageCode](#) (int index) const
- uint [linesVisible](#) () const
- void [reset](#) ()
- void [setCurrentLanguageCode](#) (const QString &lang)
- void [setLinesVisible](#) (uint lines)
- void [setPlaceholderText](#) (const QString &msg)
- void [setTitle](#) (const QString &title)
- void [setTitleWidget](#) (QWidget \*const twdg)
- virtual void [setValues](#) (const [MetaEngine::AltLangMap](#) &values)
- [DTextEdit](#) \* [textEdit](#) () const
- QWidget \* [titleWidget](#) () const
- [MetaEngine::AltLangMap](#) & [values](#) () const

## Static Public Member Functions

- static QStringList [allLanguagesRFC3066](#) ()
- static QString [languageNameRFC3066](#) (const QString &code)

## Protected Slots

- void [slotDeleteValue](#) ()
- void [slotSelectionChanged](#) ()
- void [slotTextChanged](#) ()

## Protected Member Functions

- void [changeEvent](#) (QEvent \*e) override
- void [populateLangAltListEntries](#) ()

## Friends

- class [Private](#)

## 6.85.1 Constructor & Destructor Documentation

### 6.85.1.1 AltLangStrEdit()

```
Digikam::AltLangStrEdit::AltLangStrEdit (
    QWidget *const parent,
    unsigned int lines = 3 ) [explicit]
```

Default constructor. Use lines to use a specific number of lines with text editor.

## 6.85.2 Member Function Documentation

### 6.85.2.1 addCurrent()

```
void Digikam::AltLangStrEdit::addCurrent ( )
```

Ensure that the current language is added to the list of entries, even if the text is empty. [signalValueAdded\(\)](#) will be emitted.

### 6.85.2.2 allLanguagesRFC3066()

```
QStringList Digikam::AltLangStrEdit::allLanguagesRFC3066 ( ) [static]
```

Return all language codes available following the RFC 3066.

### 6.85.2.3 languageNameRFC3066()

```
QString Digikam::AltLangStrEdit::languageNameRFC3066 (
    const QString & code ) [static]
```

Return the literal name of RFC 3066 language code (format FR-fr for ex).

### 6.85.2.4 reset()

```
void Digikam::AltLangStrEdit::reset ( )
```

Reset widget, clear all entries

### 6.85.2.5 setLinesVisible()

```
void Digikam::AltLangStrEdit::setLinesVisible (
    uint lines )
```

Fix lines visible in text editor to lines. If zero, do not fix layout to number of lines visible.

### 6.85.2.6 setTitle()

```
void Digikam::AltLangStrEdit::setTitle (
    const QString & title )
```

Create a title widget with a QLabel and relevant text. If a title widget already exists, it's replaced.

### 6.85.2.7 setTitleWidget()

```
void Digikam::AltLangStrEdit::setTitleWidget (
    QWidget *const twdg )
```

Create a title with a specific widget instance (aka a QCheckBox for ex). If a title widget already exists, it's replaced.

### 6.85.2.8 signalModified

```
void Digikam::AltLangStrEdit::signalModified (
    const QString & lang,
    const QString & text ) [signal]
```

Emitted when the user changes the text for the current language.

### 6.85.2.9 signalSelectionChanged

```
void Digikam::AltLangStrEdit::signalSelectionChanged (
    const QString & lang ) [signal]
```

Emitted when the current language changed.

### 6.85.2.10 signalValueAdded

```
void Digikam::AltLangStrEdit::signalValueAdded (
    const QString & lang,
    const QString & text ) [signal]
```

Emitted when an entry for a new language is added.

### 6.85.2.11 signalValueDeleted

```
void Digikam::AltLangStrEdit::signalValueDeleted (
    const QString & lang ) [signal]
```

Emitted when the entry for a language is removed.

### 6.85.2.12 slotEnabledInternalWidgets

```
void Digikam::AltLangStrEdit::slotEnabledInternalWidgets (
    bool b ) [slot]
```

Can be used to turn on/off visibility of internal widgets. This do not includes the title widget.

### 6.85.2.13 titleWidget()

```
QWidget * Digikam::AltLangStrEdit::titleWidget ( ) const
```

Return the current title widget instance. If no previous call of [setTitle\(\)](#) or [setWidgetTitle\(\)](#), this function will return nullptr.

## 6.86 Digikam::AltLangStrEdit::Private Class Reference

### Public Attributes

- QString **currentLanguage** = QLatin1String("x-default")
- QPushButton \* **delValueButton** = nullptr
- QGridLayout \* **grid** = nullptr
- QComboBox \* **languageCB** = nullptr
- uint **linesVisible** = 0
- [LocalizeSelector](#) \* **localizeSelector** = nullptr
- QWidget \* **titleWidget** = nullptr
- QString **trCode**
- [DOnlineTranslator](#) \* **trengine** = nullptr
- [DTextEdit](#) \* **valueEdit** = nullptr
- [MetaEngine::AltLangMap](#) **values**

## 6.87 Digikam::AnimatedClearButton Class Reference

Inheritance diagram for Digikam::AnimatedClearButton:



### Public Slots

- void **animateVisible** (bool visible)  
*Set visible, possibly with animation.*
- void **setDirectlyVisible** (bool visible)  
*Set visible without animation.*
- void **slotPixmapEnabled** (bool b)  
*Set enabled state for drawing the pixmap.*

## Signals

- void **clicked** ()
- void **visibleChanged** (bool v)

## Public Member Functions

- **AnimatedClearButton** (QWidget \*const parent=nullptr)
- QPixmap **pixmap** () const
- void **setPixmap** (const QPixmap &p)
- void **setShallBeShown** (bool show)
- QSize **sizeHint** () const override
- void **stayVisibleWhenAnimatedOut** (bool stayVisible)

## Protected Slots

- void **updateAnimationSettings** ()
- void **visibleChanged** ()

## Protected Member Functions

- void **mouseReleaseEvent** (QMouseEvent \*event) override
- void **paintEvent** (QPaintEvent \*event) override

## 6.87.1 Member Function Documentation

### 6.87.1.1 setShallBeShown()

```
void Digikam::AnimatedClearButton::setShallBeShown (
    bool show )
```

Sets a primary condition for the button to be shown. If false, [animateVisible\(\)](#) will have no effect.

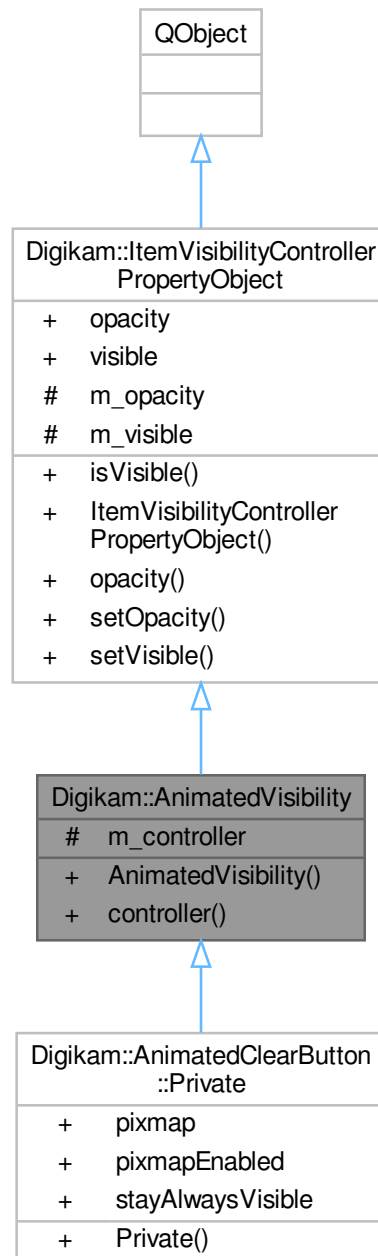
### 6.87.1.2 stayVisibleWhenAnimatedOut()

```
void Digikam::AnimatedClearButton::stayVisibleWhenAnimatedOut (
    bool stayVisible )
```

This parameter determines the behavior when the animation to hide the widget has finished: If stayVisible is true, the widget remains visible, but paints nothing. If stayVisible is false, setVisible(false) is called, which removes the widget for layouting etc. Default: false

## 6.88 Digikam::AnimatedVisibility Class Reference

Inheritance diagram for Digikam::AnimatedVisibility:



### Public Member Functions

- [AnimatedVisibility](#) (QObject \*const parent=nullptr)
- [ItemVisibilityController](#) \* **controller** () const



**Public Member Functions inherited from [Digikam::ItemVisibilityControllerPropertyObject](#)**

- bool **isVisible** () const
- [ItemVisibilityControllerPropertyObject](#) (QObject \*const parent=nullptr)
- qreal **opacity** () const
- void **setOpacity** (qreal opacity)
- void **setVisible** (bool visible)

**Protected Attributes**

- [ItemVisibilityController](#) \* **m\_controller** = nullptr

**Protected Attributes inherited from [Digikam::ItemVisibilityControllerPropertyObject](#)**

- qreal **m\_opacity** = 0.0
- bool **m\_visible** = false

**Additional Inherited Members****Signals inherited from [Digikam::ItemVisibilityControllerPropertyObject](#)**

- void **opacityChanged** ()
- void **visibleChanged** ()

**Properties inherited from [Digikam::ItemVisibilityControllerPropertyObject](#)**

- qreal **opacity**
- bool **visible**

**6.88.1 Constructor & Destructor Documentation****6.88.1.1 AnimatedVisibility()**

```
Digikam::AnimatedVisibility::AnimatedVisibility (
    QObject *const parent = nullptr ) [explicit]
```

A convenience class: The property object brings its own controller. Ready to use: Just construct an object and connect to the signals. Please note the difference between controller()->setVisible() and setVisible(): You want to call the controller's method!

**6.89 Digikam::AntiVignettingContainer Class Reference****Public Attributes**

- bool **addvignetting** = true
- double **density** = 2.0
- double **innerradius** = 1.0
- double **outerradius** = 1.0
- double **power** = 1.0
- double **xshift** = 0.0
- double **yshift** = 0.0

## 6.90 Digikam::AntiVignettingFilter Class Reference

Inheritance diagram for Digikam::AntiVignettingFilter:



### Public Member Functions

- **AntiVignettingFilter** (`DImg *const orgImage`, `QObject *const parent=nullptr`, `const AntiVignettingContainer &settings=AntiVignettingContainer()`)

- **AntiVignettingFilter** (QObject \*const parent=nullptr)
- [FilterAction filterAction](#) () override
- QString [filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) (DImg \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static QString [DisplayableName](#) ()
- static QString [FilterIdentifier](#) ()
- static QList< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.90.1 Member Function Documentation

### 6.90.1.1 filterAction()

`FilterAction` Digikam::AntiVignettingFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.90.1.2 filterIdentifier()

`QString` Digikam::AntiVignettingFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

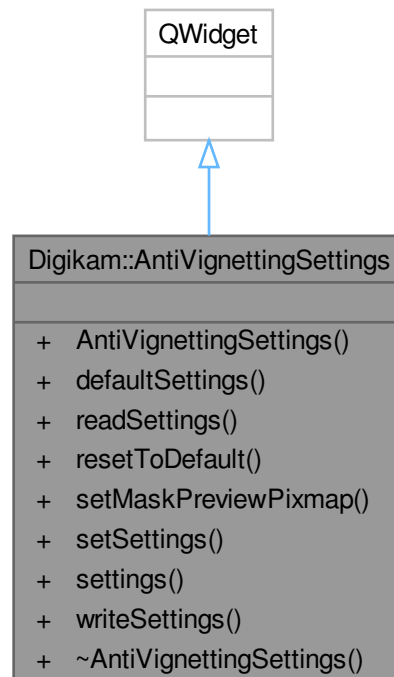
### 6.90.1.3 readParameters()

`void` Digikam::AntiVignettingFilter::readParameters (   
 const `FilterAction` & *action* ) [override], [virtual]

Implements [Digikam::DImgThreadedFilter](#).

## 6.91 Digikam::AntiVignettingSettings Class Reference

Inheritance diagram for Digikam::AntiVignettingSettings:



## Signals

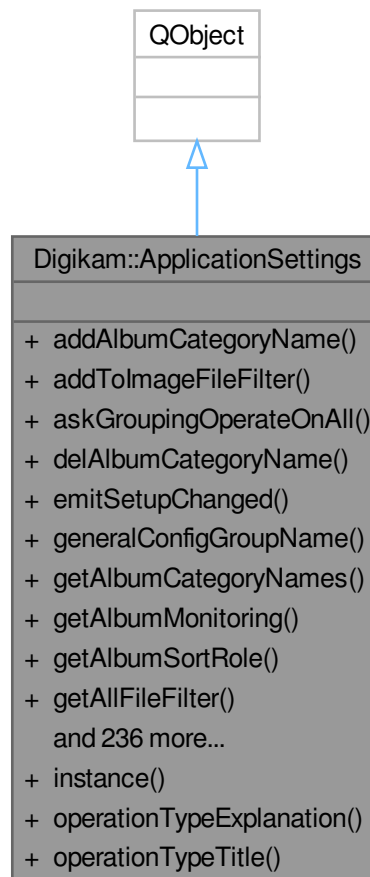
- void **signalSettingsChanged** ()

## Public Member Functions

- **AntiVignettingSettings** (QWidget \*parent)
- [AntiVignettingContainer](#) **defaultSettings** () const
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setMaskPreviewPixmap** (const QPixmap &pix)
- void **setSettings** (const [AntiVignettingContainer](#) &settings)
- [AntiVignettingContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.92 Digikam::ApplicationSettings Class Reference

Inheritance diagram for Digikam::ApplicationSettings:



## Classes

- class [Private](#)

## Public Types

- enum **AlbumSortRole** { **ByFolder** = 0 , **ByCategory** , **ByDate** }
- enum **ApplyToEntireGroup** { **No** = 0 , **Yes** , **Ask** }
- enum **ItemLeftClickAction** { **ShowPreview** = 0 , **StartEditor** , **ShowOnTable** , **OpenDefault** }
- typedef QHash< [OperationType](#), ApplicationSettings::ApplyToEntireGroup > **OperationModes**
- typedef QHash< [OperationType](#), QString > **OperationStrings**
- enum [StringComparisonType](#) { **Natural** = 0 , **Normal** }

## Signals

- void **balooSettingsChanged** ()
- void **recurseSettingsChanged** ()
- void **setupChanged** ()

## Public Member Functions

- bool **addAlbumCategoryName** (const QString &name) const
- void **addTolImageFileFilter** (const QString &extensions)
- bool [askGroupingOperateOnAll](#) ([OperationType](#) type)
- bool **delAlbumCategoryName** (const QString &name) const
- void **emitSetupChanged** ()
- QString **generalConfigGroupName** () const
- QStringList **getAlbumCategoryNames** () const
- bool **getAlbumMonitoring** () const
- AlbumSortRole **getAlbumSortRole** () const
- QString **getAllFileFilter** () const
- bool **getAllGroupsOpen** () const
- QFont **getApplicationFont** () const
- QString **getApplicationStyle** () const
- bool **getApplySidebarChangesDirectly** () const
- QString **getAudioFileFilter** () const
- bool **getCleanAtStart** () const
- QString **getCurrentTheme** () const
- bool **getDatabaseDirSetAtCmd** () const
- [DbEngineParameters](#) **getDbEngineParameters** () const
- int **getDefaultIconSize** () const
- bool **getDetectFacesInNewImages** () const
- bool **getDrawFramesToGrouped** () const
- int **getDuplicatesAlbumTagRelation** () const
- [Haarface::ReflmageSelMethod](#) **getDuplicatesReflmageSelMethod** () const
- int **getDuplicatesSearchLastMaxSimilarity** () const
- int **getDuplicatesSearchLastMinSimilarity** () const
- int **getDuplicatesSearchRestrictions** () const
- bool **getExpandNewCurrentItem** () const
- int **getFaceDetectionAccuracy** () const
- [FaceScanSettings::FaceDetectionModel](#) **getFaceDetectionModel** () const
- [FaceScanSettings::FaceDetectionSize](#) **getFaceDetectionSize** () const

- int **getFaceRecognitionAccuracy** () const
- [FaceScanSettings::FaceRecognitionModel](#) **getFaceRecognitionModel** () const
- ApplyToEntireGroup [getGroupingOperateOnAll](#) ([OperationType](#) type) const
- bool **getHelpBoxNotificationSeen** ()
- bool **getIconShowAspectRatio** () const
- bool **getIconShowColorLabel** () const
- bool **getIconShowComments** () const
- bool **getIconShowCoordinates** () const
- bool **getIconShowDate** () const
- bool **getIconShowFullscreen** () const
- bool **getIconShowImageFormat** () const
- bool **getIconShowModDate** () const
- bool **getIconShowName** () const
- bool [getIconShowOverlays](#) () const
- bool **getIconShowPickLabel** () const
- bool **getIconShowRating** () const
- bool **getIconShowResolution** () const
- bool **getIconShowSize** () const
- bool **getIconShowTags** () const
- bool **getIconShowTitle** () const
- QString **getIconTheme** () const
- QFont **getIconViewFont** () const
- QString **getImageFileFilter** () const
- int **getImageSeparationMode** () const
- int **getImageSeparationSortOrder** () const
- int **getImageSorting** () const
- int **getImageSortOrder** () const
- int **getItemLeftClickAction** () const
- int **getMinimumSimilarityBound** () const
- QString **getMovieFileFilter** () const
- [PreviewSettings](#) **getPreviewSettings** () const
- bool **getPreviewShowIcons** () const
- bool **getPreviewSmoothScaled** () const
- int **getRatingFilterCond** () const
- QString **getRawFileFilter** () const
- bool **getRecurseAlbums** () const
- bool **getRecurseTags** () const
- bool **getScaleFitToWindow** () const
- bool **getScanAtStart** () const
- bool **getScrollItemToCenter** () const
- bool **getSelectFirstAlbumItem** () const
- bool **getShowAlbumToolTips** () const
- bool **getShowFolderTreeViewItemsCount** () const
- bool **getShowPermanentDeleteDialog** () const
- bool **getShowSplashScreen** () const
- bool **getShowThumbbar** () const
- bool **getShowToolTips** () const
- bool **getShowTrashDeleteDialog** () const
- [DMultiTabBar::TextStyle](#) **getSidebarTitleStyle** () const
- [StringComparisonType](#) **getStringComparisonType** () const
- bool **getSyncBalooToDigikam** () const
- bool **getSyncDigikamToBaloo** () const
- QFont **getToolTipsFont** () const
- bool **getToolTipsShowAlbumCaption** () const
- bool **getToolTipsShowAlbumCategory** () const



- bool **getToolTipsShowAlbumCollection** () const
- bool **getToolTipsShowAlbumDate** () const
- bool **getToolTipsShowAlbumName** () const
- bool **getToolTipsShowAlbumPreview** () const
- bool **getToolTipsShowAlbumTitle** () const
- bool **getToolTipsShowComments** () const
- bool **getToolTipsShowFileDate** () const
- bool **getToolTipsShowFileName** () const
- bool **getToolTipsShowFileSize** () const
- bool **getToolTipsShowImageAR** () const
- bool **getToolTipsShowImageDim** () const
- bool **getToolTipsShowImageType** () const
- bool **getToolTipsShowLabelRating** () const
- bool **getToolTipsShowPhotoDate** () const
- bool **getToolTipsShowPhotoExpo** () const
- bool **getToolTipsShowPhotoFlash** () const
- bool **getToolTipsShowPhotoFocal** () const
- bool **getToolTipsShowPhotoLens** () const
- bool **getToolTipsShowPhotoMake** () const
- bool **getToolTipsShowPhotoMode** () const
- bool **getToolTipsShowPhotoWB** () const
- bool **getToolTipsShowTags** () const
- bool **getToolTipsShowTitles** () const
- bool **getToolTipsShowVideoAspectRatio** () const
- bool **getToolTipsShowVideoAudioBitRate** () const
- bool **getToolTipsShowVideoAudioChannelType** () const
- bool **getToolTipsShowVideoAudioCodec** () const
- bool **getToolTipsShowVideoDuration** () const
- bool **getToolTipsShowVideoFrameRate** () const
- bool **getToolTipsShowVideoVideoCodec** () const
- int **getTreeViewFaceSize** () const
- QFont **getTreeViewFont** () const
- int **getTreeViewIconSize** () const
- int **getUpdateType** () const
- bool **getUpdateWithDebug** () const
- bool **getUseNativeFileDialog** () const
- bool **getUseTrash** () const
- [VersionManagerSettings](#) **getVersionManagerSettings** () const
- bool **isStringTypeNatural** () const
- bool [readMsgBoxShouldBeShown](#) (const QString &dontShowAgainName)
- void **readSettings** ()
- void [saveMsgBoxShouldBeShown](#) (const QString &dontShowAgainName)
- void **saveSettings** ()
- void **setAlbumCategoryNames** (const QStringList &list)
- void **setAlbumMonitoring** (bool val)
- void **setAlbumSortRole** (const AlbumSortRole role)
- void **setAllGroupsOpen** (bool val)
- void **setApplicationFont** (const QFont &fnt)
- void **setApplicationStyle** (const QString &style)
- void **setApplySidebarChangesDirectly** (bool val)
- void **setCleanAtStart** (bool val)
- void **setCurrentTheme** (const QString &theme)
- void **setDatabaseDirSetAtCmd** (bool val)
- void **setDbEngineParameters** (const [DbEngineParameters](#) &params)
- void **setDefaultIconSize** (int val)

- void **setDetectFacesInNewImages** (bool val)
- void **setDrawFramesToGrouped** (bool val)
- void **setDuplicatesAlbumTagRelation** (int val)
- void **setDuplicatesReferenceImageSelectionMethod** ([Haarface::RefImageSelMethod](#) val)
- void **setDuplicatesSearchLastMaxSimilarity** (int val)
- void **setDuplicatesSearchLastMinSimilarity** (int val)
- void **setDuplicatesSearchRestrictions** (int val)
- void **setExpandNewCurrentItem** (bool val)
- void **setFaceDetectionAccuracy** (int value)
- void **setFaceDetectionModel** ([FaceScanSettings::FaceDetectionModel](#) model)
- void **setFaceDetectionSize** ([FaceScanSettings::FaceDetectionSize](#) size)
- void **setFaceRecognitionAccuracy** (int value)
- void **setFaceRecognitionModel** ([FaceScanSettings::FaceRecognitionModel](#) model)
- void **setGroupingOperateOnAll** ([OperationType](#) type, ApplyToEntireGroup applyAll)
- void **setHelpBoxNotificationSeen** (bool val)
- void **setIconShowAspectRatio** (bool val)
- void **setIconShowColorLabel** (bool val)
- void **setIconShowComments** (bool val)
- void **setIconShowCoordinates** (bool val)
- void **setIconShowDate** (bool val)
- void **setIconShowFullscreen** (bool val)
- void **setIconShowImageFormat** (bool val)
- void **setIconShowModDate** (bool val)
- void **setIconShowName** (bool val)
- void **setIconShowOverlays** (bool val)
- void **setIconShowPickLabel** (bool val)
- void **setIconShowRating** (bool val)
- void **setIconShowResolution** (bool val)
- void **setIconShowSize** (bool val)
- void **setIconShowTags** (bool val)
- void **setIconShowTitle** (bool val)
- void **setIconTheme** (const QString &theme)
- void **setIconViewFont** (const QFont &font)
- void **setImageSeparationMode** (int mode)
- void **setImageSeparationSortOrder** (int order)
- void **setImageSorting** (int sorting)
- void **setImageSortOrder** (int order)
- void **setItemLeftClickAction** (int action)
- void **setMinimumSimilarityBound** (int val)
- void **setPreviewSettings** (const [PreviewSettings](#) &settings)
- void **setPreviewShowIcons** (bool val)
- void **setPreviewSmoothScaled** (bool val)
- void **setRatingFilterCond** (int val)
- void **setRecurseAlbums** (bool val)
- void **setRecurseTags** (bool val)
- void **setScaleFitToWindow** (bool val)
- void **setScanAtStart** (bool val)
- void **setScrollItemToCenter** (bool val)
- void **setSelectFirstAlbumItem** (bool val)
- void **setShowAlbumToolTips** (bool val)
- void **setShowFolderTreeViewItemsCount** (bool val)
- void **setShowOnlyPersonTagsInPeopleSidebar** (bool val)
- void **setShowPermanentDeleteDialog** (bool val)
- void **setShowSplashScreen** (bool val)
- void **setShowThumbbar** (bool val)

- void **setShowToolTips** (bool val)
- void **setShowTrashDeleteDialog** (bool val)
- void **setSidebarTitleStyle** ([DMultiTabBar::TextStyle](#) style)
- void **setStringComparisonType** ([ApplicationSettings::StringComparisonType](#) val)
- void **setSyncBalooToDigikam** (bool val)
- void **setSyncDigikamToBaloo** (bool val)
- void **setToolTipsFont** (const QFont &font)
- void **setToolTipsShowAlbumCaption** (bool val)
- void **setToolTipsShowAlbumCategory** (bool val)
- void **setToolTipsShowAlbumCollection** (bool val)
- void **setToolTipsShowAlbumDate** (bool val)
- void **setToolTipsShowAlbumName** (bool val)
- void **setToolTipsShowAlbumPreview** (bool val)
- void **setToolTipsShowAlbumTitle** (bool val)
- void **setToolTipsShowComments** (bool val)
- void **setToolTipsShowFileDate** (bool val)
- void **setToolTipsShowFileName** (bool val)
- void **setToolTipsShowFileSize** (bool val)
- void **setToolTipsShowImageAR** (bool val)
- void **setToolTipsShowImageDim** (bool val)
- void **setToolTipsShowImageType** (bool val)
- void **setToolTipsShowLabelRating** (bool val)
- void **setToolTipsShowPhotoDate** (bool val)
- void **setToolTipsShowPhotoExpo** (bool val)
- void **setToolTipsShowPhotoFlash** (bool val)
- void **setToolTipsShowPhotoFocal** (bool val)
- void **setToolTipsShowPhotoLens** (bool val)
- void **setToolTipsShowPhotoMake** (bool val)
- void **setToolTipsShowPhotoMode** (bool val)
- void **setToolTipsShowPhotoWB** (bool val)
- void **setToolTipsShowTags** (bool val)
- void **setToolTipsShowTitles** (bool val)
- void **setToolTipsShowVideoAspectRatio** (bool val)
- void **setToolTipsShowVideoAudioBitRate** (bool val)
- void **setToolTipsShowVideoAudioChannelType** (bool val)
- void **setToolTipsShowVideoAudioCodec** (bool val)
- void **setToolTipsShowVideoDuration** (bool val)
- void **setToolTipsShowVideoFrameRate** (bool val)
- void **setToolTipsShowVideoVideoCodec** (bool val)
- void **setTreeViewFaceSize** (int val)
- void **setTreeViewFont** (const QFont &font)
- void **setTreeViewIconSize** (int val)
- void **setUpdateType** (int type)
- void **setUpdateWithDebug** (bool dbg)
- void **setUseNativeFileDialog** (bool val)
- void **setUseTrash** (bool val)
- void **setVersionManagerSettings** (const [VersionManagerSettings](#) &settings)
- bool **showAlbumToolTipsIsValid** () const
- bool **showOnlyPersonTagsInPeopleSidebar** () const
- bool **showToolTipsIsValid** () const

### Static Public Member Functions

- static [ApplicationSettings](#) \* **instance** ()
- static QString **operationTypeExplanation** ([OperationType](#) type)
- static QString **operationTypeTitle** ([OperationType](#) type)

## Friends

- class **ApplicationSettingsCreator**

## 6.92.1 Member Enumeration Documentation

### 6.92.1.1 StringComparisonType

```
enum Digikam::ApplicationSettings::StringComparisonType
```

Possible ways of comparing strings.

#### Enumerator

Natural	Natural compare using KStringHandler::naturalCompare.
Normal	Normal comparison using Qt's compare function.

## 6.92.2 Member Function Documentation

### 6.92.2.1 askGroupingOperateOnAll()

```
bool Digikam::ApplicationSettings::askGroupingOperateOnAll (
    OperationType type )
```

Asks the user whether the operation should be performed on all grouped images or just the first. Also supplies an option to remember the answer.

#### Parameters

<i>type</i>	Operation to be performed
-------------	---------------------------

#### Returns

Whether to apply to all images or just one

### 6.92.2.2 getGroupingOperateOnAll()

```
ApplicationSettings::ApplyToEntireGroup Digikam::ApplicationSettings::getGroupingOperateOnAll
(
    OperationType type ) const
```

Tells whether an operation should be performed on all grouped items or just the head item.

#### Parameters

<i>type</i>	Operation to be performed
-------------	---------------------------

**Returns**

Whether to apply to all images or just one, or ask

**6.92.2.3 getIconShowOverlays()**

```
bool Digikam::ApplicationSettings::getIconShowOverlays ( ) const
```

Determines whether the overlay buttons should be displayed on the icons.

**6.92.2.4 getStringComparisonType()**

```
ApplicationSettings::StringComparisonType Digikam::ApplicationSettings::getStringComparison←  
Type ( ) const
```

Tells in which way strings are compared at the moment.

**Returns**

string comparison type to use.

**6.92.2.5 operationTypeExplanation()**

```
QString Digikam::ApplicationSettings::operationTypeExplanation (   
    OperationType type ) [static]
```

Gives a translated explanation of the operation and an empty string, if there is none (e.g. for tooltips)

**Parameters**

<i>type</i>	Operation to be performed
-------------	---------------------------

**Returns**

Translated operation explanation

**6.92.2.6 operationTypeTitle()**

```
QString Digikam::ApplicationSettings::operationTypeTitle (   
    OperationType type ) [static]
```

Gives the translated title/short explanation of the operation

**Parameters**

<i>type</i>	Operation to be performed
-------------	---------------------------

**Returns**

Translated operation title/short explanation

**6.92.2.7 readMsgBoxShouldBeShown()**

```
bool Digikam::ApplicationSettings::readMsgBoxShouldBeShown (
    const QString & dontShowAgainName )
```

**Returns**

true if the corresponding message box should be shown.

**Parameters**

<i>dontShowAgainName</i>	the name that identify the message box.
--------------------------	---

**6.92.2.8 saveMsgBoxShouldBeShown()**

```
void Digikam::ApplicationSettings::saveMsgBoxShouldBeShown (
    const QString & dontShowAgainName )
```

Save the fact that the message box should not be shown again.

**Parameters**

<i>dontShowAgainName</i>	the name that identify the message box. If empty, this method does nothing.
--------------------------	---

**6.92.2.9 setGroupingOperateOnAll()**

```
void Digikam::ApplicationSettings::setGroupingOperateOnAll (
    OperationType type,
    ApplicationSettings::ApplyToEntireGroup applyAll )
```

Defines whether an operation should be performed on all grouped items or just the head item.

**Parameters**

<i>type</i>	Operation to be performed
<i>applyAll</i>	Whether to apply to all images or just one, or ask

**6.92.2.10 setIconShowOverlays()**

```
void Digikam::ApplicationSettings::setIconShowOverlays (
    bool val )
```

Sets the visibility of the overlay buttons on the image icons.

### 6.92.2.11 setImageSorting()

```
void Digikam::ApplicationSettings::setImageSorting (
    int sorting )
```

means ascending or descending

### 6.92.2.12 setStringComparisonType()

```
void Digikam::ApplicationSettings::setStringComparisonType (
    ApplicationSettings::StringComparisonType val )
```

Defines the way in which string comparisons are performed.

Parameters

<i>val</i>	new way to compare strings
------------	----------------------------

## 6.93 Digikam::ApplicationSettings::Private Class Reference

### Public Member Functions

- void **init** ()
- **Private** ([ApplicationSettings](#) \*const qq)

### Public Attributes

- QStringList **albumCategoryNames**
- bool **albumMonitoring** = false  
*album settings*
- ApplicationSettings::AlbumSortRole **albumSortRole** = ApplicationSettings::ByFolder  
*album view settings*
- bool **allGroupsOpen** = false
- QFont **applicationFont** = QFontDatabase::systemFont(QFontDatabase::GeneralFont)
- QString **applicationIcon**
- QString **applicationStyle** = qApp->style()->objectName()
- bool **cleanAtStart** = true
- KSharedConfigPtr **config**
- const QString **configAlbumCollectionsEntry** = QLatin1String("Album Collections")
- const QString **configAlbumMonitoringEntry** = QLatin1String("Album Monitoring")
- const QString **configAlbumSortRoleEntry** = QLatin1String("Album Sort Role")
- const QString **configAllGroupsOpenEntry** = QLatin1String("All Groups Open")
- const QString **configApplicationFontEntry** = QLatin1String("Application Font")
- const QString **configApplicationStyleEntry** = QLatin1String("Application Style")
- const QString **configApplySidebarChangesDirectlyEntry** = QLatin1String("Apply [Sidebar](#) Changes Directly")
- const QString **configCleanAtStartEntry** = QLatin1String("Clean core DB At Start")
- const QString **configDefaultIconSizeEntry** = QLatin1String("Default Icon Size")
- const QString **configDefaultTreeFaceSizeEntry** = QLatin1String("Default Tree Face Size")

- const QString **configDefaultTreeIconSizeEntry** = QLatin1String("Default Tree Icon Size")
- const QString **configDetectFacesInNewImagesEntry** = QLatin1String("Detect faces in newly added images")
- const QString **configDrawFramesToGroupedEntry** = QLatin1String("Draw Frames To Grouped Items")
- const QString **configDuplicatesSearchLastAlbumTagRelation** = QLatin1String("Last search album tag relation")
- const QString **configDuplicatesSearchLastMaxSimilarity** = QLatin1String("Last maximum similarity")
- const QString **configDuplicatesSearchLastMinSimilarity** = QLatin1String("Last minimum similarity")
- const QString **configDuplicatesSearchLastRestrictions** = QLatin1String("Last search results restriction")
- const QString **configDuplicatesSearchReferenceSelectionMethod** = QLatin1String("Last reference image method")
- const QString **configExpandNewCurrentItemEntry** = QLatin1String("Expand New Current Item On Click")
- const QString **configFaceDetectionAccuracyEntry** = QLatin1String("Face Detection Accuracy")
- const QString **configFaceDetectionModelEntry** = QLatin1String("Face Detection Model")
- const QString **configFaceDetectionSizeEntry** = QLatin1String("Face Detection Size")
- const QString **configFaceRecognitionAccuracyEntry** = QLatin1String("Face Recognition Accuracy")
- const QString **configFaceRecognitionModelEntry** = QLatin1String("Face Recognition Model")
- const QString **configGroupBaloo** = QLatin1String("Baloo Settings")
- const QString **configGroupDefault** = QLatin1String("Album Settings")
- const QString **configGroupDuplicatesSearch** = QLatin1String("Find Duplicates View")
- const QString **configGroupExif** = QLatin1String("EXIF Settings")
- const QString **configGroupFaceDetection** = QLatin1String("Face Detection Settings")
- const QString **configGroupGeneral** = QLatin1String("General Settings")
- const QString **configGroupGrouping** = QLatin1String("Grouping Behaviour")
- const ApplicationSettings::OperationStrings **configGroupingOperateOnAll** = createConfigGrouping←  
OperateOnAll()
- const QString **configGroupMetadata** = QLatin1String("Metadata Settings")
- const QString **configGroupVersioning** = QLatin1String("Versioning Settings")
- const QString **configIconShowAspectRatioEntry** = QLatin1String("Icon Show Aspect Ratio")
- const QString **configIconShowColorLabelEntry** = QLatin1String("Icon Show Color Label")
- const QString **configIconShowCommentsEntry** = QLatin1String("Icon Show Comments")
- const QString **configIconShowCoordinatesEntry** = QLatin1String("Icon Show Coordinates")
- const QString **configIconShowDateEntry** = QLatin1String("Icon Show Date")
- const QString **configIconShowFullscreenEntry** = QLatin1String("Icon Show Fullscreen")
- const QString **configIconShowImageFormatEntry** = QLatin1String("Icon Show Image Format")
- const QString **configIconShowModificationDateEntry** = QLatin1String("Icon Show Modification Date")
- const QString **configIconShowNameEntry** = QLatin1String("Icon Show Name")
- const QString **configIconShowOverlaysEntry** = QLatin1String("Icon Show Overlays")
- const QString **configIconShowPickLabelEntry** = QLatin1String("Icon Show Pick Label")
- const QString **configIconShowRatingEntry** = QLatin1String("Icon Show Rating")
- const QString **configIconShowResolutionEntry** = QLatin1String("Icon Show Resolution")
- const QString **configIconShowSizeEntry** = QLatin1String("Icon Show Size")
- const QString **configIconShowTagsEntry** = QLatin1String("Icon Show Tags")
- const QString **configIconShowTitleEntry** = QLatin1String("Icon Show Title")
- const QString **configIconThemeEntry** = QLatin1String("Icon Theme")
- const QString **configIconViewFontEntry** = QLatin1String("IconView Font")
- const QString **configImageSeparationModeEntry** = QLatin1String("Image Group Mode")
- const QString **configImageSeparationSortOrderEntry** = QLatin1String("Image Group Sort Order")
- const QString **configImageSortingEntry** = QLatin1String("Image Sorting")
- const QString **configImageSortOrderEntry** = QLatin1String("Image Sort Order")
- const QString **configItemLeftClickActionEntry** = QLatin1String("Item Left Click Action")
- const QString **configMinimumSimilarityBound** = QLatin1String("Lower bound for minimum similarity")
- const QString **configPreviewConvertToEightBitEntry** = QLatin1String("Preview Convert To Eight Bit")
- const QString **configPreviewImageSmoothScaledEntry** = QLatin1String("Preview Image Smooth Scaled")
- const QString **configPreviewLoadFullItemSizeEntry** = QLatin1String("Preview Load Full Image Size")



- const QString **configPreviewRawUseLoadingDataEntry** = QLatin1String("Preview Raw Use Loading Data")
- const QString **configPreviewScaleFitToWindowEntry** = QLatin1String("Preview Scale Fit To Window")
- const QString **configPreviewShowIconsEntry** = QLatin1String("Preview Show Icons")
- const QString **configRatingFilterConditionEntry** = QLatin1String("Rating Filter Condition")
- const QString **configRecursiveAlbumsEntry** = QLatin1String("Recursive Albums")
- const QString **configRecursiveTagsEntry** = QLatin1String("Recursive Tags")
- const QString **configScanAtStartEntry** = QLatin1String("Scan At Start")
- const QString **configScrollItemToCenterEntry** = QLatin1String("Scroll Current Item To Center")
- const QString **configSelectFirstAlbumItemEntry** = QLatin1String("Select First Album Item")
- const QString **configShowAlbumToolTipsEntry** = QLatin1String("Show Album ToolTips")
- const QString **configShowFolderTreeViewItemsCountEntry** = QLatin1String("Show Folder Tree View Items Count")
- const QString **configShowOnlyPersonTagsInPeopleSidebarEntry** = QLatin1String("Show Only Face Tags For Assigning Name")
- const QString **configShowPermanentDeleteDialogEntry** = QLatin1String("Show Permanent Delete Dialog")
- const QString **configShowSplashEntry** = QLatin1String("Show Splash")
- const QString **configShowThumbbarEntry** = QLatin1String("Show Thumbbar")
- const QString **configShowToolTipsEntry** = QLatin1String("Show ToolTips")
- const QString **configShowTrashDeleteDialogEntry** = QLatin1String("Show Trash Delete Dialog")
- const QString **configSidebarTitleStyleEntry** = QLatin1String("Sidebar Title Style")
- const QString **configStringComparisonTypeEntry** = QLatin1String("String Comparison Type")
- const QString **configSyncBalootoDigikamEntry** = QLatin1String("Sync Baloo to Digikam")
- const QString **configSyncDigikamtoBalooEntry** = QLatin1String("Sync Digikam to Baloo")
- const QString **configThemeEntry** = QLatin1String("Theme")
- const QString **configToolTipsFontEntry** = QLatin1String("ToolTips Font")
- const QString **configToolTipsShowAlbumCaptionEntry** = QLatin1String("ToolTips Show Album Caption")
- const QString **configToolTipsShowAlbumCategoryEntry** = QLatin1String("ToolTips Show Album Category")
- const QString **configToolTipsShowAlbumCollectionEntry** = QLatin1String("ToolTips Show Album Collection")
- const QString **configToolTipsShowAlbumDateEntry** = QLatin1String("ToolTips Show Album Date")
- const QString **configToolTipsShowAlbumNameEntry** = QLatin1String("ToolTips Show Album Name")
- const QString **configToolTipsShowAlbumPreviewEntry** = QLatin1String("ToolTips Show Album Preview")
- const QString **configToolTipsShowAlbumTitleEntry** = QLatin1String("ToolTips Show Album Title")
- const QString **configToolTipsShowCommentsEntry** = QLatin1String("ToolTips Show Comments")
- const QString **configToolTipsShowFileDateEntry** = QLatin1String("ToolTips Show File Date")
- const QString **configToolTipsShowFileNameEntry** = QLatin1String("ToolTips Show File Name")
- const QString **configToolTipsShowFileSizeEntry** = QLatin1String("ToolTips Show File Size")
- const QString **configToolTipsShowImageAREntry** = QLatin1String("ToolTips Show Image AR")
- const QString **configToolTipsShowImageDimEntry** = QLatin1String("ToolTips Show Image Dim")
- const QString **configToolTipsShowImageTypeEntry** = QLatin1String("ToolTips Show Image Type")
- const QString **configToolTipsShowLabelRatingEntry** = QLatin1String("ToolTips Show Label Rating")
- const QString **configToolTipsShowPhotoDateEntry** = QLatin1String("ToolTips Show Photo Date")
- const QString **configToolTipsShowPhotoExpoEntry** = QLatin1String("ToolTips Show Photo Expo")
- const QString **configToolTipsShowPhotoFlashEntry** = QLatin1String("ToolTips Show Photo Flash")
- const QString **configToolTipsShowPhotoFocalEntry** = QLatin1String("ToolTips Show Photo Focal")
- const QString **configToolTipsShowPhotoLensEntry** = QLatin1String("ToolTips Show Photo Lens")
- const QString **configToolTipsShowPhotoMakeEntry** = QLatin1String("ToolTips Show Photo Make")
- const QString **configToolTipsShowPhotoModeEntry** = QLatin1String("ToolTips Show Photo Mode")
- const QString **configToolTipsShowPhotoWBEEntry** = QLatin1String("ToolTips Show Photo WB")
- const QString **configToolTipsShowTagsEntry** = QLatin1String("ToolTips Show Tags")
- const QString **configToolTipsShowTitlesEntry** = QLatin1String("ToolTips Show Titles")
- const QString **configToolTipsShowVideoAspectRatioEntry** = QLatin1String("ToolTips Show Video Aspect Ratio")

- const QString **configToolTipsShowVideoAudioBitRateEntry** = QLatin1String("ToolTips Show Audio Bit Rate")
- const QString **configToolTipsShowVideoAudioChannelTypeEntry** = QLatin1String("ToolTips Show Audio Channel Type")
- const QString **configToolTipsShowVideoAudioCodecEntry** = QLatin1String("ToolTips Show Audio Codec")
- const QString **configToolTipsShowVideoDurationEntry** = QLatin1String("ToolTips Show Video Duration")
- const QString **configToolTipsShowVideoFrameRateEntry** = QLatin1String("ToolTips Show Video Frame Rate")
- const QString **configToolTipsShowVideoVideoCodecEntry** = QLatin1String("ToolTips Show Video Codec")
- const QString **configTreeViewFontEntry** = QLatin1String("TreeView Font")
- const QString **configUpdateType** = QLatin1String("Update Type")
- const QString **configUpdateWithDebug** = QLatin1String("Update With Debug")
- const QString **configUseNativeFileDialogEntry** = QLatin1String("Use Native File Dialog")
- const QString **configUseTrashEntry** = QLatin1String("Use Trash")
- QString **currentTheme**  
*theme settings*
- bool **databaseDirSetAtCmd** = false
- [DbEngineParameters](#) **databaseParams**  
*database settings*
- bool **detectFacesInNewImages** = false
- bool **drawFramesToGrouped** = true  
*grouped item draw setting*
- int **duplicatesSearchLastAlbumTagRelation** = 0
- int **duplicatesSearchLastMaxSimilarity** = 100
- int **duplicatesSearchLastMinSimilarity** = 90
- [Haarface::RefImageSelMethod](#) **duplicatesSearchLastReferenceImageSelectionMethod** = [Haarface::RefImageSelMethod::](#)
- int **duplicatesSearchLastRestrictions** = 0
- bool **expandNewCurrentItem** = true  
*expand item setting*
- int **faceDetectionAccuracy** = 7  
*face detection settings*
- [FaceScanSettings::FaceDetectionModel](#) **faceDetectionModel** = [FaceScanSettings::FaceDetectionModel::YuNet](#)
- [FaceScanSettings::FaceDetectionSize](#) **faceDetectionSize** = [FaceScanSettings::FaceDetectionSize::Medium](#)
- int **faceRecognitionAccuracy** = 7  
*face recognition settings*
- [FaceScanSettings::FaceRecognitionModel](#) **faceRecognitionModel** = [FaceScanSettings::FaceRecognitionModel::SFace](#)
- [ApplicationSettings::OperationModes](#) **groupingOperateOnAll** = [ApplicationSettings::OperationModes\(\)](#)  
*Grouping operation settings.*
- bool **iconShowAspectRatio** = false
- bool **iconShowColorLabel** = true
- bool **iconShowComments** = true
- bool **iconShowCoordinates** = true
- bool **iconShowDate** = true
- bool **iconShowFullscreen** = true
- bool **iconShowImageFormat** = true
- bool **iconShowModDate** = false
- bool **iconShowName** = true  
*icon view settings*
- bool **iconShowOverlays** = true
- bool **iconShowPickLabel** = false
- bool **iconShowRating** = true
- bool **iconShowResolution** = false

- bool **iconShowSize** = false
- bool **iconShowTags** = true
- bool **iconShowTitle** = true
- QFont **iconviewFont** = QFontDatabase::systemFont(QFontDatabase::GeneralFont)
- int **imageSeparationMode** = ItemSortSettings::CategoryByAlbum
- int **imageSeparationSortOrder** = ItemSortSettings::AscendingOrder
- bool **imageSmoothScaled** = true
- int **imageSorting** = ItemSortSettings::AscendingOrder
- int **imageSortOrder** = ItemSortSettings::SortByFileName
- icon view settings*
- int **itemLeftClickAction** = ApplicationSettings::ShowPreview
- int **minimumSimilarityBound** = 40
- [PreviewSettings](#) **previewSettings**
- preview settings*
- bool **previewShowIcons** = true
- int **ratingFilterCond** = ItemFilterSettings::GreaterEqualCondition
- bool **recursiveAlbums** = false
- bool **recursiveTags** = true
- bool **scaleFitToWindow** = false
- bool **scanAtStart** = true
- bool **scrollItemToCenter** = false
- item center setting*
- bool **selectFirstAlbumItem** = true
- select first item setting*
- bool **showAlbumToolTips** = false
- Folder-view tooltip settings.*
- bool **showFolderTreeViewItemsCount** = false
- bool **showOnlyPersonTagsInPeopleSidebar** = false
- tag filter setting*
- bool **showPermanentDeleteDialog** = true
- bool **showSplash** = true
- start up setting*
- bool **showThumbbar** = true
- bool **showToolTips** = false
- Icon-view tooltip settings.*
- bool **showTrashDeleteDialog** = true
- bool **sidebarApplyDirectly** = false
- metadata setting*
- [DMultiTabBar::TextStyle](#) **sidebarTitleStyle** = [DMultiTabBar::AllIconsText](#)
- [ApplicationSettings::StringComparisonType](#) **stringComparisonType** = [ApplicationSettings::Natural](#)
- misc*
- bool **syncToBaloo** = false
- bool **syncToDigikam** = false
- Baloo settings.*
- int **thumbnailSize** = ThumbnailSize::Medium
- icon view settings*
- QFont **toolTipsFont** = QFontDatabase::systemFont(QFontDatabase::GeneralFont)
- bool **tooltipShowAlbumCaption** = true
- bool **tooltipShowAlbumCategory** = true
- bool **tooltipShowAlbumCollection** = true
- bool **tooltipShowAlbumDate** = true
- bool **tooltipShowAlbumName** = false
- bool **tooltipShowAlbumPreview** = false

- bool **tooltipShowAlbumTitle** = true
  - bool **tooltipShowComments** = true
  - bool **tooltipShowFileDate** = false
  - bool **tooltipShowFileName** = true
  - bool **tooltipShowFileSize** = false
  - bool **tooltipShowImageAR** = true
  - bool **tooltipShowImageDim** = true
  - bool **tooltipShowImageType** = false
  - bool **tooltipShowLabelRating** = true
  - bool **tooltipShowPhotoDate** = true
  - bool **tooltipShowPhotoExpo** = true
  - bool **tooltipShowPhotoFlash** = false
  - bool **tooltipShowPhotoFocal** = true
  - bool **tooltipShowPhotoLens** = true
  - bool **tooltipShowPhotoMake** = true
  - bool **tooltipShowPhotoMode** = true
  - bool **tooltipShowPhotoWb** = false
  - bool **tooltipShowTags** = true
  - bool **tooltipShowTitles** = false
  - bool **tooltipShowVideoAspectRatio** = true
  - bool **tooltipShowVideoAudioBitRate** = true
  - bool **tooltipShowVideoAudioChannelType** = true
  - bool **tooltipShowVideoAudioCodec** = true
  - bool **tooltipShowVideoDuration** = true
  - bool **tooltipShowVideoFrameRate** = true
  - bool **tooltipShowVideoVideoCodec** = true
  - int **treeThumbFaceSize** = 48
  - int **treeThumbnailSize** = 22
- tree-view settings*
- QFont **treeviewFont** = QFontDatabase::systemFont(QFontDatabase::GeneralFont)
  - int **updateType** = 0
  - bool **updateWithDebug** = false
  - bool **useNativeFileDialog** = false
- file dialog setting*
- bool **useTrash** = true
- file ops settings*
- [VersionManagerSettings](#) **versionSettings**
- versioning settings*

## 6.94 Digikam::AssignedBatchTools Class Reference

### Public Member Functions

- QString **targetSuffix** (bool \*const extSet=nullptr) const

### Public Attributes

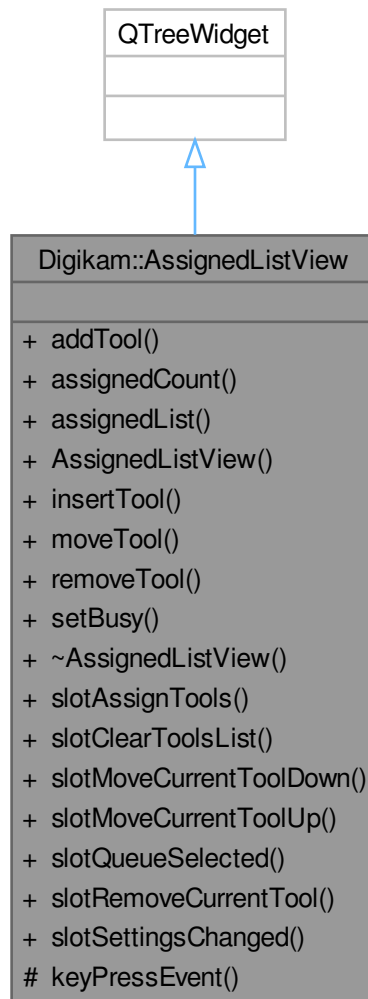
- QString **m\_destFileName**
- QUrl **m\_itemUrl**
- [BatchSetList](#) **m\_toolsList**

### 6.94.1 Detailed Description

Container to assign Batch tools and settings to an item by Url. Url is used only with [ActionThread](#) class.

## 6.95 Digikam::AssignedListView Class Reference

Inheritance diagram for Digikam::AssignedListView:



### Public Slots

- void **slotAssignTools** (const QMap< int, QString > &)
- void **slotClearToolsList** ()
- void **slotMoveCurrentToolDown** ()
- void **slotMoveCurrentToolUp** ()
- void **slotQueueSelected** (int, const [QueueSettings](#) &, const [AssignedBatchTools](#) &)
- void **slotRemoveCurrentTool** ()
- void **slotSettingsChanged** (const [BatchToolSet](#) &)

## Signals

- void **signalAssignedToolsChanged** (const [AssignedBatchTools](#) &)
- void **signalToolSelected** (const [BatchToolSet](#) &)

## Public Member Functions

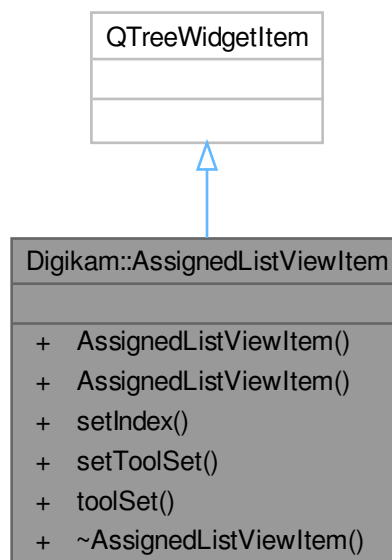
- [AssignedListViewItem](#) \* **addTool** (const [BatchToolSet](#) &set)
- int **assignedCount** ()
- [AssignedBatchTools](#) **assignedList** ()
- [AssignedListView](#) (QWidget \*const parent)
- [AssignedListViewItem](#) \* **insertTool** ([AssignedListViewItem](#) \*const preceding, const [BatchToolSet](#) &set)
- [AssignedListViewItem](#) \* **moveTool** ([AssignedListViewItem](#) \*const preceding, const [BatchToolSet](#) &set)
- bool **removeTool** (const [BatchToolSet](#) &set)
- void **setBusy** (bool b)

## Protected Member Functions

- void **keyPressEvent** (QKeyEvent \*) override

## 6.96 Digikam::AssignedListViewItem Class Reference

Inheritance diagram for Digikam::AssignedListViewItem:

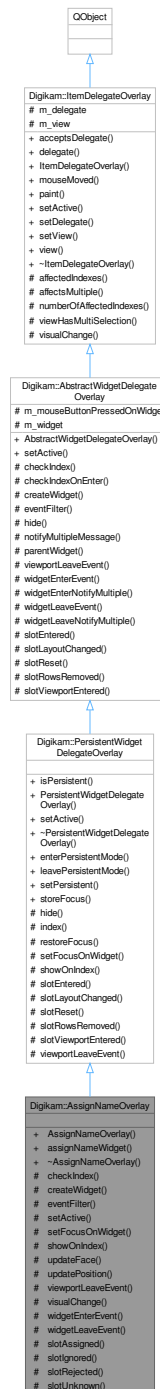


### Public Member Functions

- **AssignedListItem** (QTreeWidgetItem \*const parent)
- **AssignedListItem** (QTreeWidgetItem \*const parent, QTreeWidgetItem \*const preceding)
- void **setIndex** (int index)
- void **setToolSet** (const [BatchToolSet](#) &set)
- [BatchToolSet](#) **toolSet** ()

## 6.97 Digikam::AssignNameOverlay Class Reference

Inheritance diagram for Digikam::AssignNameOverlay:



### Signals

- void **confirmFaces** (const QList< QModelIndex > &indexes, int tagId)
- void **ignoreFaces** (const QList< QModelIndex > &indexes)
- void **removeFaces** (const QList< QModelIndex > &indexes)
- void **unknownFaces** (const QList< QModelIndex > &indexes)



**Signals inherited from [Digikam::ItemDelegateOverlay](#)**

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

**Public Member Functions**

- **AssignNameOverlay** (QObject \*const parent)
- [AssignNameWidget](#) \* **assignNameWidget** () const

**Public Member Functions inherited from [Digikam::PersistentWidgetDelegateOverlay](#)**

- bool **isPersistent** () const
- [PersistentWidgetDelegateOverlay](#) (QObject \*const parent)
- void **setActive** (bool active) override

**Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

**Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)**

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

**Protected Slots**

- void **slotAssigned** (const [TaggingAction](#) &action, const [ItemInfo](#) &, const QVariant &faceIdentifier)
- void **slotIgnored** (const [ItemInfo](#) &, const QVariant &faceIdentifier)
- void **slotRejected** (const [ItemInfo](#) &, const QVariant &faceIdentifier)
- void **slotUnknown** (const [ItemInfo](#) &, const QVariant &faceIdentifier)

**Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)****Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)****Protected Member Functions**

- bool **checkIndex** (const QModelIndex &index) const override
- QWidget \* **createWidget** () override
- bool **eventFilter** (QObject \*o, QEvent \*e) override
- void **setActive** (bool) override
- void **setFocusOnWidget** () override
- void **showOnIndex** (const QModelIndex &index) override  
*see [slotEntered\(\)](#)*
- void **updateFace** ()
- void **updatePosition** ()
- void **viewportLeaveEvent** (QObject \*obj, QEvent \*event) override
- void **visualChange** () override
- void **widgetEnterEvent** () override
- void **widgetLeaveEvent** () override

### Protected Member Functions inherited from [Digikam::PersistentWidgetDelegateOverlay](#)

- void [hide](#) () override
- QModelIndex [index](#) () const
- void [restoreFocus](#) ()
- void [slotEntered](#) (const QModelIndex &index) override
- void [slotLayoutChanged](#) () override
- void [slotReset](#) () override
- void [slotRowsRemoved](#) (const QModelIndex &parent, int start, int end) override
- void [slotViewportEntered](#) () override
- void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event) override

### Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- void [widgetLeaveNotifyMultiple](#) ()

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

### Additional Inherited Members

### Public Slots inherited from [Digikam::PersistentWidgetDelegateOverlay](#)

- void [enterPersistentMode](#) ()
- void [leavePersistentMode](#) ()
- void [setPersistent](#) (bool persistent)
- void [storeFocus](#) ()

### Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.97.1 Member Function Documentation

### 6.97.1.1 checkIndex()

```
bool Digikam::AssignNameOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.97.1.2 createWidget()

```
QWidget * Digikam::AssignNameOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.97.1.3 setActive()

```
void Digikam::AssignNameOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.97.1.4 setFocusOnWidget()

```
void Digikam::AssignNameOverlay::setFocusOnWidget ( ) [override], [protected], [virtual]
```

Reimplement to set the focus on the correct subwidget. Default implementation sets focus on widget()

Reimplemented from [Digikam::PersistentWidgetDelegateOverlay](#).

### 6.97.1.5 showOnIndex()

```
void Digikam::AssignNameOverlay::showOnIndex (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::PersistentWidgetDelegateOverlay](#).

### 6.97.1.6 updateFace()

```
void Digikam::AssignNameOverlay::updateFace ( ) [protected]
```

The order to plug these functions is important, since [setUserData\(\)](#) controls how the Overlay appears on a particular face.

### 6.97.1.7 viewportLeaveEvent()

```
void Digikam::AssignNameOverlay::viewportLeaveEvent (
    QObject * obj,
    QEvent * event ) [override], [protected], [virtual]
```

Called when a QEvent::Leave of the viewport is received. The default implementation [hide\(\)](#)s.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.97.1.8 visualChange()

```
void Digikam::AssignNameOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

### 6.97.1.9 widgetEnterEvent()

```
void Digikam::AssignNameOverlay::widgetEnterEvent ( ) [override], [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

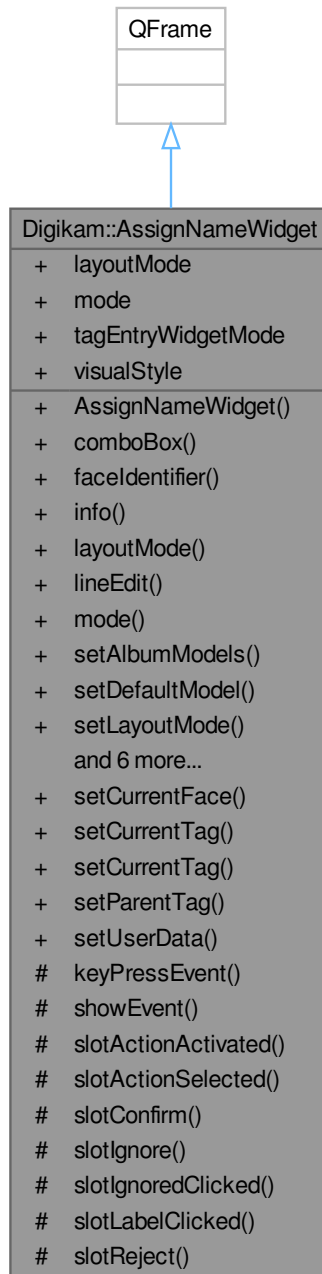
### 6.97.1.10 widgetLeaveEvent()

```
void Digikam::AssignNameOverlay::widgetLeaveEvent ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.98 Digikam::AssignNameWidget Class Reference

Inheritance diagram for Digikam::AssignNameWidget:



### Classes

- class [Private](#)

## Public Types

- enum **LayoutMode** { **InvalidLayout** , **FullLine** , **TwoLines** , **Compact** }
- enum **Mode** { **InvalidMode** , **UnconfirmedEditMode** , **ConfirmedMode** , **ConfirmedEditMode** , **IgnoredMode** }
- enum **TagEntryWidgetMode** { **InvalidTagEntryWidgetMode** , **AddTagsComboBoxMode** , **AddTagsLineEditMode** }
- enum **VisualStyle** { **InvalidVisualStyle** , **StyledFrame** , **TranslucentDarkRound** , **TranslucentThemedFrameless** }

## Public Slots

- void **setCurrentFace** (const [FaceTagsIface](#) &face)
- void **setCurrentTag** (int tagId)
- void **setCurrentTag** ([TAlbum](#) \*album)
- void **setParentTag** ([TAlbum](#) \*album)
- void **setUserData** (const [ItemInfo](#) &info, const [QVariant](#) &faceIdentifier=[QVariant](#)())

## Signals

- void **assigned** (const [TaggingAction](#) &action, const [ItemInfo](#) &info, const [QVariant](#) &faceIdentifier)
- void **ignored** (const [ItemInfo](#) &info, const [QVariant](#) &faceIdentifier)
- void **ignoredClicked** (const [ItemInfo](#) &info, const [QVariant](#) &faceIdentifier)
- void **labelClicked** (const [ItemInfo](#) &info, const [QVariant](#) &faceIdentifier)
- void **rejected** (const [ItemInfo](#) &info, const [QVariant](#) &faceIdentifier)
- void **selected** (const [TaggingAction](#) &action, const [ItemInfo](#) &info, const [QVariant](#) &faceIdentifier)

## Public Member Functions

- [AssignNameWidget](#) ([QWidget](#) \*const parent=nullptr)
- [AddTagsComboBox](#) \* **comboBox** () const  
*The combo box or line edit in use, if any.*
- [QVariant](#) **faceIdentifier** () const
- [ItemInfo](#) **info** () const
- [LayoutMode](#) **layoutMode** () const
- [AddTagsLineEdit](#) \* **lineEdit** () const
- [Mode](#) **mode** () const
- void **setAlbumModels** ([TagModel](#) \*const model, [TagPropertiesFilterModel](#) \*const filteredModel, [CheckableAlbumFilterModel](#) \*const filterModel)
- void **setDefaultModel** ()
- void **setLayoutMode** ([LayoutMode](#) mode)
- void **setMode** ([Mode](#) mode)
- void **setTagEntryWidgetMode** ([TagEntryWidgetMode](#) mode)
- void **setVisualStyle** ([VisualStyle](#) style)
- [TagEntryWidgetMode](#) **tagEntryWidgetMode** () const
- [VisualStyle](#) **visualStyle** () const

### Protected Slots

- void **slotActionActivated** (const [TaggingAction](#) &action)
- void **slotActionSelected** (const [TaggingAction](#) &action)
- void **slotConfirm** ()
- void **slotIgnore** ()
- void **slotIgnoredClicked** ()
- void **slotLabelClicked** ()
- void **slotReject** ()

### Protected Member Functions

- void **keyPressEvent** (QKeyEvent \*e) override
- void **showEvent** (QShowEvent \*e) override

### Properties

- LayoutMode **layoutMode**
- Mode **mode**
- TagEntryWidgetMode **tagEntryWidgetMode**
- VisualStyle **visualStyle**

## 6.98.1 Constructor & Destructor Documentation

### 6.98.1.1 AssignNameWidget()

```
Digikam::AssignNameWidget::AssignNameWidget (
    QWidget *const parent = nullptr ) [explicit]
```

Please take care: you must set all four modes before usage!

## 6.98.2 Member Function Documentation

### 6.98.2.1 assigned

```
void Digikam::AssignNameWidget::assigned (
    const TaggingAction & action,
    const ItemInfo & info,
    const QVariant & faceIdentifier ) [signal]
```

A name has been assigned to the associated face. This can be an existing tag, or a new tag, as described by [TaggingAction](#). For convenience, info() and faceIdentifier() are provided.

### 6.98.2.2 ignoredClicked

```
void Digikam::AssignNameWidget::ignoredClicked (
    const ItemInfo & info,
    const QVariant & faceIdentifier ) [signal]
```

In IgnoredMode, this signal is emitted when the user clicked on the label.

### 6.98.2.3 labelClicked

```
void Digikam::AssignNameWidget::labelClicked (
    const ItemInfo & info,
    const QVariant & faceIdentifier ) [signal]
```

In ConfirmedMode, this signal is emitted when the user clicked on the label.

### 6.98.2.4 rejected

```
void Digikam::AssignNameWidget::rejected (
    const ItemInfo & info,
    const QVariant & faceIdentifier ) [signal]
```

The suggestion has been rejected and the face will be moved to Unknown. For convenience, info() and faceIdentifier() are provided.

### 6.98.2.5 selected

```
void Digikam::AssignNameWidget::selected (
    const TaggingAction & action,
    const ItemInfo & info,
    const QVariant & faceIdentifier ) [signal]
```

An action has been selected. This purely signals user interaction, no fixed decision - mouse hover may be enough to emit this signal. The action may be invalid (user switched back to empty selection).

### 6.98.2.6 setAlbumModels()

```
void Digikam::AssignNameWidget::setAlbumModels (
    TagModel *const model,
    TagPropertiesFilterModel *const filteredModel,
    CheckableAlbumFilterModel *const filterModel )
```

Set the tag model to use for completion.

### 6.98.2.7 setCurrentTag

```
void Digikam::AssignNameWidget::setCurrentTag (
    int tagId ) [slot]
```

Sets the suggested (UnconfirmedEditMode) or assigned (ConfirmedMode) tag to be displayed.

### 6.98.2.8 setMode()

```
void Digikam::AssignNameWidget::setMode (
    Mode mode )
```

Reject tooltip and icon should be updated even if the same mode is passed, because Unconfirmed and Unknown. Faces have the same mode but different tooltips and icons.



### 6.98.2.9 setParentTag

```
void Digikam::AssignNameWidget::setParentTag (
    TAlbum * album ) [slot]
```

Set a parent tag for suggesting a parent tag for a new tag, and a default action.

### 6.98.2.10 setUserData

```
void Digikam::AssignNameWidget::setUserData (
    const ItemInfo & info,
    const QVariant & faceIdentifier = QVariant() ) [slot]
```

The identifying information emitted with the signals. Ignored faces are drawn over with a different overlay, as Reject button should be disabled.

## 6.99 Digikam::AssignNameWidget::Private Class Reference

### Public Member Functions

- bool **isValid** () const
- **Private** ([AssignNameWidget](#) \*const qq)
- void **updateContents** ()
- void **updateIgnoreButton** ()
- void **updateModes** ()
- void **updateRejectButton** ()

### Public Attributes

- [DClickLabel](#) \* **clickLabel** = nullptr
- [AddTagsComboBox](#) \* **comboBox** = nullptr
- [QToolButton](#) \* **confirmButton** = nullptr
- [AlbumPointer](#)< [TAlbum](#) > **currentTag**
- [QVariant](#) **faceIdentifier**
- [QToolButton](#) \* **ignoreButton** = nullptr
- [ItemInfo](#) **info**
- [QGridLayout](#) \* **layout** = nullptr
- [LayoutMode](#) **layoutMode** = [InvalidLayout](#)
- [AddTagsLineEdit](#) \* **lineEdit** = nullptr
- [Mode](#) **mode** = [InvalidMode](#)
- bool **modelsGiven** = false
- [AlbumPointer](#)< [TAlbum](#) > **parentTag**
- [AssignNameWidget](#) \*const **q** = nullptr
- [QToolButton](#) \* **rejectButton** = nullptr
- [TagPropertiesFilterModel](#) \* **tagFilteredModel** = nullptr
- [CheckableAlbumFilterModel](#) \* **tagFilterModel** = nullptr
- [TagModel](#) \* **tagModel** = nullptr
- [VisualStyle](#) **visualStyle** = [InvalidVisualStyle](#)
- [TagEntryWidgetMode](#) **widgetMode** = [InvalidTagEntryWidgetMode](#)

## 6.99.1 Member Function Documentation

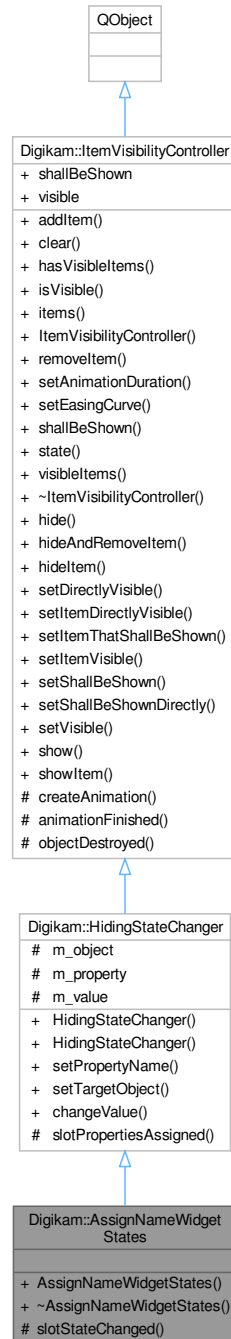
### 6.99.1.1 updateRejectButton()

```
void Digikam::AssignNameWidget::Private::updateRejectButton ( )
```

Reject button shows different Tooltips and icons for Unconfirmed and Unknown faces, however both are of type UnconfirmedEditMode. This method is responsible for the Tooltip updates.

## 6.100 Digikam::AssignNameWidgetStates Class Reference

Inheritance diagram for Digikam::AssignNameWidgetStates:



### Public Member Functions

- **AssignNameWidgetStates** ([FacelItem](#) \*const item)

## Public Member Functions inherited from [Digikam::HidingStateChanger](#)

- [HidingStateChanger](#) (QObject \*const parent=nullptr)
- [HidingStateChanger](#) (QObject \*const target, const QByteArray &property, QObject \*const parent=nullptr)
- void **setProperty** (const QByteArray &propertyName)
- void **setTargetObject** (QObject \*const object)

## Public Member Functions inherited from [Digikam::ItemVisibilityController](#)

- void **addItem** (QObject \*const object)
- void **clear** ()
- bool **hasVisibleItems** ([IncludeFadingOutMode](#) mode=[IncludeFadingOut](#)) const
- bool **isVisible** () const
- QList< QObject \* > **items** () const
- **ItemVisibilityController** (QObject \*const parent=nullptr)
- void **removeItem** (QObject \*const object)
- void **setAnimationDuration** (int msec)
- void **setEasingCurve** (const QEasingCurve &easing)
- bool **shallBeShown** () const
- **State** **state** () const
- QList< QObject \* > **visibleItems** ([IncludeFadingOutMode](#) mode=[IncludeFadingOut](#)) const

## Protected Slots

- void **slotStateChanged** ()

## Protected Slots inherited from [Digikam::HidingStateChanger](#)

- void **slotPropertiesAssigned** (bool)

## Protected Slots inherited from [Digikam::ItemVisibilityController](#)

- void **animationFinished** ()
- void **objectDestroyed** (QObject \*)

## Additional Inherited Members

## Public Types inherited from [Digikam::ItemVisibilityController](#)

- enum [IncludeFadingOutMode](#) { [IncludeFadingOut](#) , [ExcludeFadingOut](#) }
- enum [State](#) { [Hidden](#) , [FadingIn](#) , [Visible](#) , [FadingOut](#) }

## Public Slots inherited from [Digikam::HidingStateChanger](#)

- void **changeValue** (const QVariant &value)

## Public Slots inherited from [Digikam::ItemVisibilityController](#)

- void **hide** ()
- void [hideAndRemoveItem](#) (QObject \*item)
- void **hideItem** (QObject \*item)
- void **setDirectlyVisible** (bool visible)
- void **setItemDirectlyVisible** (QObject \*item, bool visible)
- void [setItemThatShallBeShown](#) (QObject \*item)
- void **setItemVisible** (QObject \*item, bool visible)
- void [setShallBeShown](#) (bool shallBeShown)
- void **setShallBeShownDirectly** (bool shallBeShown)
- void **setVisible** (bool visible)
- void [show](#) ()
- void [showItem](#) (QObject \*item)

## Signals inherited from [Digikam::HidingStateChanger](#)

- void [finished](#) ()
- void [stateChanged](#) ()

## Signals inherited from [Digikam::ItemVisibilityController](#)

- void [hiddenAndRemoved](#) (QObject \*item)
- void [propertiesAssigned](#) (bool visible)
- void [propertiesAssigned](#) (QObject \*item, bool visible)

## Protected Member Functions inherited from [Digikam::ItemVisibilityController](#)

- virtual QPropertyAnimation \* [createAnimation](#) (QObject \*item)

## Protected Attributes inherited from [Digikam::HidingStateChanger](#)

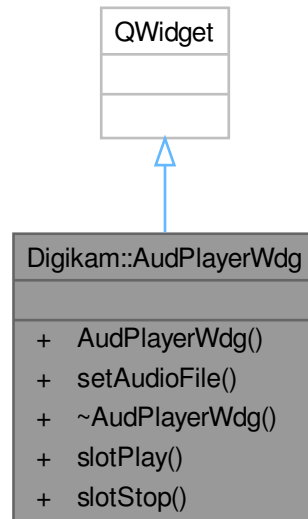
- QObject \* **m\_object** = nullptr
- QByteArray **m\_property**
- QVariant **m\_value**

## Properties inherited from [Digikam::ItemVisibilityController](#)

- bool **shallBeShown**
- bool **visible**

## 6.101 Digikam::AudPlayerWdg Class Reference

Inheritance diagram for Digikam::AudPlayerWdg:



### Public Slots

- void **slotPlay** ()
- void **slotStop** ()

### Public Member Functions

- **AudPlayerWdg** (QWidget \*const parent=nullptr)
- void **setAudioFile** (const QString &afile)

## 6.102 Digikam::AutoCrop Class Reference

Inheritance diagram for Digikam::AutoCrop:



### Public Member Functions

- [AutoCrop](#) (`DImg *const orgImage`, `QObject *const parent=nullptr`)
- `QRect autoInnerCrop () const`
- void [startAnalyse](#) () override

## Public Member Functions inherited from [Digikam::DImgThreadedAnalyser](#)

- **DImgThreadedAnalyser** ([DImg](#) \*const orgImage, [QObject](#) \*const parent=nullptr, const [QString](#) &name=QString())  
*Constructs an image analyser with all arguments (ready to use). The given original image will be copied. You need to call [startFilter\(\)](#) to start the threaded computation. To run analyser without to use multithreading, call [startFilterDirectly\(\)](#).*
- **DImgThreadedAnalyser** ([QObject](#) \*const parent=nullptr, const [QString](#) &name=QString())  
*Constructs a filter without argument. You need to call [setupFilter\(\)](#) and [startFilter\(\)](#) to start the threaded computation. To run filter without to use multithreading, call [startFilterDirectly\(\)](#).*

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=QString())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=QString())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > **multithreadedSteps** (int stop, int start=0) const
- virtual bool **parametersSuccessfullyRead** () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void **setFilterVersion** (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void **setupAndStartDirectly** (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void **setupFilter** (const [DImg](#) &orgImage)
- virtual void **startFilter** ()
- virtual void **startFilterDirectly** ()

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }



## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int **progress**)
- void **postProgress** (int **progress**)
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.102.1 Constructor & Destructor Documentation

### 6.102.1.1 AutoCrop()

```
Digikam::AutoCrop::AutoCrop (
    DImg *const orgImage,
    QObject *const parent = nullptr ) [explicit]
```

Standard constructor with image container to parse

## 6.102.2 Member Function Documentation

### 6.102.2.1 autoInnerCrop()

```
QRect Digikam::AutoCrop::autoInnerCrop ( ) const
```

Return inner crop area detected by [startAnalyse\(\)](#).

### 6.102.2.2 startAnalyse()

```
void Digikam::AutoCrop::startAnalyse ( ) [override], [virtual]
```

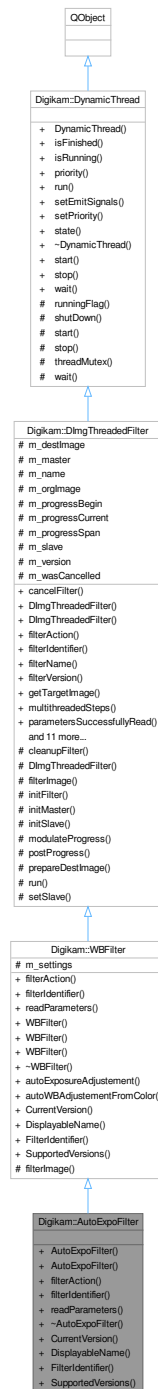
Perform auto-crop analyze to find best inner crop. Use [autoInnerCrop\(\)](#) to get computed area. This would be done in 4 steps

1. Search column wise: (a) From the left to the right, this is to get the left boundary (b) From the right to the left, this is to get the right boundary
2. Search row wise : (a) From the top to the bottom, this is to get the top boundary (b) From the bottom to the top, this is to get the bottom boundary

Implements [Digikam::DImgThreadedAnalyser](#).

## 6.103 Digikam::AutoExpoFilter Class Reference

Inheritance diagram for Digikam::AutoExpoFilter:



### Public Member Functions

- **AutoExpoFilter** (`Dimg *const orgImage`, `const Dimg *const reflImage`, `QObject *const parent=nullptr`)
- **AutoExpoFilter** (`QObject *const parent=nullptr`)

- [FilterAction filterAction](#) () override
- [QString filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::WBFilter](#)

- [FilterAction filterAction](#) () override
- [QString filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override
- [WBFilter](#) (const [WBContainer](#) &settings, [DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100)
- [WBFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent=nullptr, const [WBContainer](#) &settings=[WBContainer](#)())
- [WBFilter](#) ([QObject](#) \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

## Static Public Member Functions inherited from Digikam::WBFilter

- static void **autoExposureAdjustement** (const [DImg](#) \*const img, double &black, double &expo)
- static void **autoWBAdjustementFromColor** (const [QColor](#) &tc, double &temperature, double &green)
- static int **CurrentVersion** ()
- static [QString](#) **DisplayableName** ()
- static [QString](#) **FilterIdentifier** ()
- static [QList](#)< int > **SupportedVersions** ()

## Additional Inherited Members

## Public Types inherited from Digikam::DynamicThread

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::WBFilter

- void **filterImage** () override

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- [DImgThreadedFilter](#) ([DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const [QString](#) &name=[QString](#)())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** ([DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** ([DImgThreadedFilter](#) \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::WBFilter](#)

- [WBContainer](#) [m\\_settings](#)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- [DImg](#) [m\\_destImage](#)
- [DImgThreadedFilter](#) \* [m\\_master](#) = nullptr
- QString [m\\_name](#)
- [DImg](#) [m\\_orgImage](#)
- int [m\\_progressBegin](#) = 0
- int [m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int [m\\_progressSpan](#) = 0
- [DImgThreadedFilter](#) \* [m\\_slave](#) = nullptr
- int [m\\_version](#) = 1
- bool [m\\_wasCancelled](#) = false

### 6.103.1 Member Function Documentation

#### 6.103.1.1 [filterAction\(\)](#)

```
FilterAction Digikam::AutoExpoFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

#### 6.103.1.2 [filterIdentifier\(\)](#)

```
QString Digikam::AutoExpoFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

#### 6.103.1.3 [readParameters\(\)](#)

```
void Digikam::AutoExpoFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.104 Digikam::AutoLevelsFilter Class Reference

Inheritance diagram for Digikam::AutoLevelsFilter:



### Public Member Functions

- **AutoLevelsFilter** (`DImg *const orgImage`, `const DImg *const reflImage`, `QObject *const parent=nullptr`)
- **AutoLevelsFilter** (`QObject *const parent=nullptr`)

- [FilterAction filterAction](#) () override
- [QString filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }



## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.104.1 Member Function Documentation

### 6.104.1.1 filterAction()

`FilterAction` Digikam::AutoLevelsFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.104.1.2 filterIdentifier()

`QString` Digikam::AutoLevelsFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.104.1.3 readParameters()

```
void Digikam::AutoLevelsFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.105 Digikam::AutoTagsAssign Class Reference

### Public Member Functions

- **AutoTagsAssign** (`DetectorModel` model=`DetectorModel::YOLOV5NANO`)
- `QList< QString >` **generateTagsList** (const `DImg` &inputImage)
- `QList< QString >` **generateTagsList** (const `QImage` &inputImage)
- `QList< QList< QString > >` **generateTagsList** (const `QList< DImg >` &inputImages, int batchSize) const  
*Run in batch return the list of tags name corresponding to.*
- `QList< QList< QString > >` **generateTagsList** (const `QList< QString >` &inputImagePaths, int batchSize) const
- `QList< QString >` **generateTagsList** (const `QString` &inputImagePath)
- `QList< QString >` **getPredefinedTagsPath** ( ) const
- `cv::Mat` **prepareForDetection** (const `DImg` &inputImage) const
- `cv::Mat` **prepareForDetection** (const `QImage` &inputImage) const
- `std::vector< cv::Mat >` **prepareForDetection** (const `QList< DImg >` &inputImages, int batchSize) const
- `std::vector< cv::Mat >` **prepareForDetection** (const `QList< QString >` &inputImagePaths, int batchSize) const
- `cv::Mat` **prepareForDetection** (const `QString` &inputImagePath) const

## 6.105.1 Member Function Documentation

### 6.105.1.1 generateTagsList()

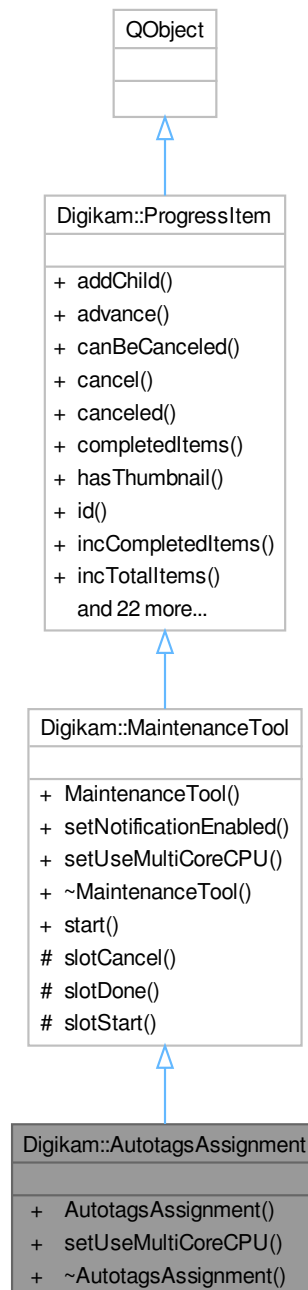
```
QList< QList< QString > > Digikam::AutoTagsAssign::generateTagsList (
    const QList< DImg > & inputImages,
    int batchSize ) const
```

#### Note

the batch size is fixed depending on the deep NN model we choose.

## 6.106 Digikam::AutotagsAssignment Class Reference

Inheritance diagram for Digikam::AutotagsAssignment:



### Public Types

- enum `AutotagsAssignmentScanMode` { `AllItems` = 0 , `NonAssignedItems` }

**Public Member Functions**

- [AutotagsAssignment](#) ([AutotagsAssignmentScanMode](#) mode, const AlbumList &list, int modelType, const QStringList &langs, [ProgressItem](#) \*const parent=nullptr)
- void [setUseMultiCoreCPU](#) (bool b) override

**Public Member Functions inherited from [Digikam::MaintenanceTool](#)**

- [MaintenanceTool](#) (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)

**Public Member Functions inherited from [Digikam::ProgressItem](#)**

- void [addChild](#) ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)
  - Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void [cancel](#) ()
- bool [canceled](#) () const
- unsigned int [completedItems](#) () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool [incCompletedItems](#) (unsigned int v=1)
- void [incTotalItems](#) (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool [canBeCanceled](#), bool hasThumb)
- void [removeChild](#) ([ProgressItem](#) \*const kiddo)
- void [reset](#) ()
  - Reset the progress value of this item to 0 and the status string to the empty string.*
- void [setComplete](#) ()
  - Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool [setCompletedItems](#) (unsigned int v)
- void [setLabel](#) (const QString &v)
- void [setProgress](#) (unsigned int v)
  - Set the progress (percentage of completion) value of this item.*
- void [setShowAtStart](#) (bool showAtStart)
  - Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void [setStatus](#) (const QString &v)
  - Set the string to be used for showing this item's current status.*
- void [setThumbnail](#) (const QIcon &icon)
  - Sets whether this item has a thumbnail.*
- void [setTotalItems](#) (unsigned int v)
- void [setUsesBusyIndicator](#) (bool useBusyIndicator)
  - Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool [showAtStart](#) () const
- const QString & [status](#) () const
- bool [totalCompleted](#) () const
- unsigned int [totalItems](#) () const
- void [updateProgress](#) ()
  - Recalculate progress according to total/completed items and update.*
- bool [usesBusyIndicator](#) () const

## Additional Inherited Members

### Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

### Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

### Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

### Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.106.1 Member Enumeration Documentation

### 6.106.1.1 AutotagsAssignmentScanMode

enum [Digikam::AutotagsAssignment::AutotagsAssignmentScanMode](#)

## Enumerator

AllItems	Clean all tags already assigned and re-scan all items.
NonAssignedItems	Scan only items with no tags assigned.

## 6.106.2 Constructor & Destructor Documentation

### 6.106.2.1 AutotagsAssignment()

```
Digikam::AutotagsAssignment::AutotagsAssignment (
    AutotagsAssignmentScanMode mode,
    const AlbumList & list,
    int modelType,
    const QStringList & langs,
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor using AlbumList as argument. If list is empty, whole Albums collection is processed.

## 6.106.3 Member Function Documentation

### 6.106.3.1 setUseMultiCoreCPU()

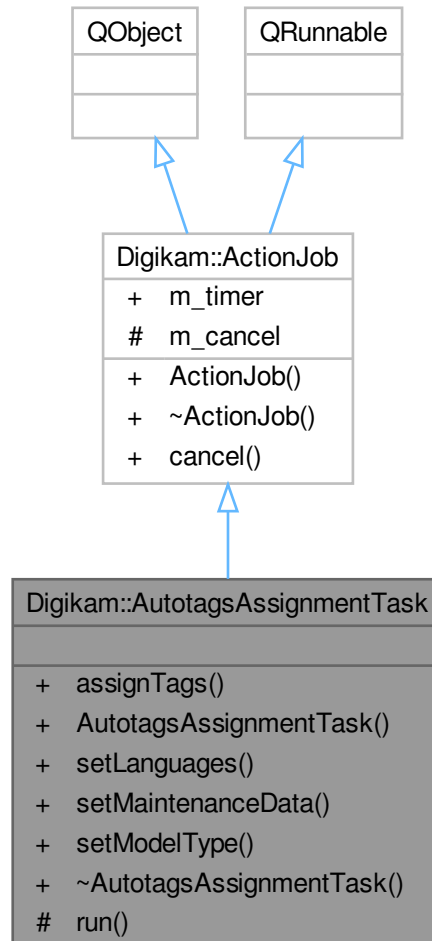
```
void Digikam::AutotagsAssignment::setUseMultiCoreCPU (
    bool ) [override], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented from [Digikam::MaintenanceTool](#).

## 6.107 Digikam::AutotagsAssignmentTask Class Reference

Inheritance diagram for Digikam::AutotagsAssignmentTask:



### Signals

- void **signalFinished** (const [ItemInfo](#) &, const QImage &, const QStringList &)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()



### Public Member Functions

- void **assignTags** (const QString &pathImage, const QList< QString > &tagsList)
- void **setLanguages** (const QStringList &langs)
- void **setMaintenanceData** ([MaintenanceData](#) \*const data=nullptr)
- void **setModelType** (int modelType)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Protected Member Functions

- void **run** () override

### Additional Inherited Members

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

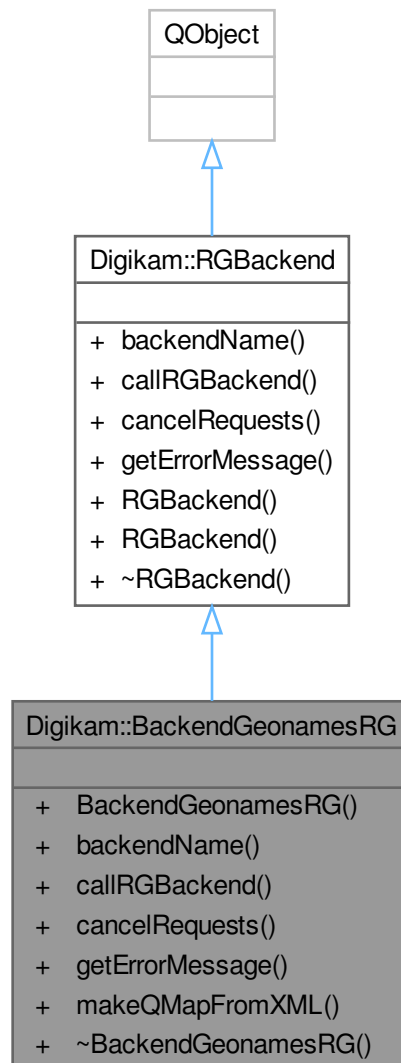
### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.108 Digikam::BackendGeonamesRG Class Reference

This class calls Geonames' reverse geocoding service.

Inheritance diagram for Digikam::BackendGeonamesRG:



### Public Member Functions

- [BackendGeonamesRG](#) (`QObject *const parent`)
- `QString` [backendName](#) () override
- `void` [callRGBBackend](#) (`const QList< RGInfo > &rgList`, `const QString &language`) override
- `void` [cancelRequests](#) () override
- `QString` [getErrorMessage](#) () override
- `QMap< QString, QString >` [makeQMapFromXML](#) (`const QString &xmlData`)
- [~BackendGeonamesRG](#) () override

### Public Member Functions inherited from [Digikam::RGBBackend](#)

- [RGBBackend](#) (`QObject *const parent`)

## Additional Inherited Members

## Signals inherited from [Digikam::RGBackend](#)

- void **signalRGReady** (const QList< [RGInfo](#) > &)  
*Emitted whenever some items are ready.*

## 6.108.1 Constructor & Destructor Documentation

### 6.108.1.1 BackendGeonamesRG()

```
Digikam::BackendGeonamesRG::BackendGeonamesRG (
    QObject *const parent ) [explicit]
```

Constructor

Parameters

<i>parent</i>	the parent object.
---------------	--------------------

### 6.108.1.2 ~BackendGeonamesRG()

```
Digikam::BackendGeonamesRG::~~BackendGeonamesRG ( ) [override]
```

Destructor

## 6.108.2 Member Function Documentation

### 6.108.2.1 backendName()

```
QString Digikam::BackendGeonamesRG::backendName ( ) [override], [virtual]
```

Returns

Backend name.

Reimplemented from [Digikam::RGBackend](#).

### 6.108.2.2 callRGBackend()

```
void Digikam::BackendGeonamesRG::callRGBackend (
    const QList< RGInfo > & rgList,
    const QString & language ) [override], [virtual]
```

Takes coordinates from each image and then connects to Open Street Map's reverse geocoding service.

## Parameters

<i>rgList</i>	A list containing information needed in reverse geocoding process. At this point, it contains only coordinates.
<i>language</i>	The language in which the data will be returned.

Implements [Digikam::RGBackend](#).

### 6.108.2.3 cancelRequests()

```
void Digikam::BackendGeonamesRG::cancelRequests ( ) [override], [virtual]
```

Implements [Digikam::RGBackend](#).

### 6.108.2.4 getErrorMessage()

```
QString Digikam::BackendGeonamesRG::getErrorMessage ( ) [override], [virtual]
```

## Returns

Error message, if any.

Reimplemented from [Digikam::RGBackend](#).

### 6.108.2.5 makeQMapFromXML()

```
QMap< QString, QString > Digikam::BackendGeonamesRG::makeQMapFromXML (
    const QString & xmlData )
```

The data is returned from Open Street Map in a XML. This function translates the XML into a QMap.

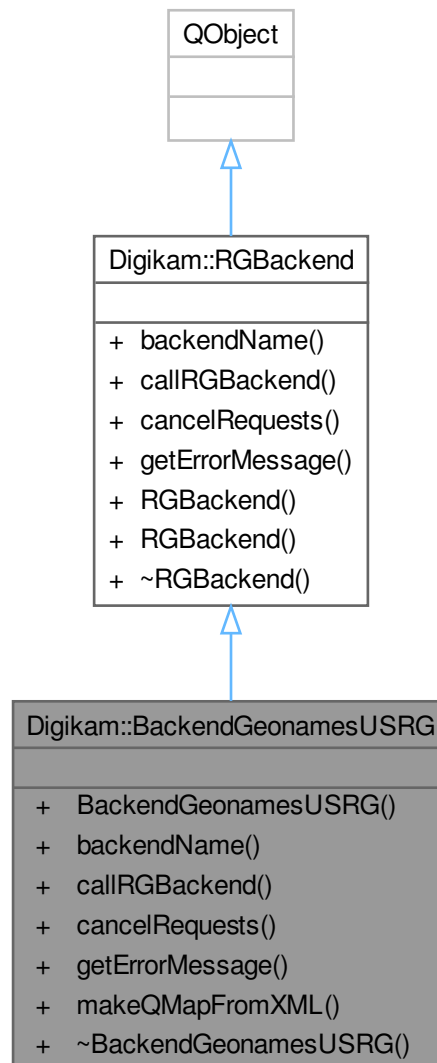
## Parameters

<i>xmlData</i>	The returned XML.
----------------	-------------------

## 6.109 Digikam::BackendGeonamesUSRG Class Reference

This class calls Geonames' get address service available only for USA locations.

Inheritance diagram for Digikam::BackendGeonamesUSRG:



### Public Member Functions

- [BackendGeonamesUSRG](#) (QObject \*const parent)
- QString [backendName](#) () override
- void [callRGBBackend](#) (const QList< [RGInfo](#) > &rgList, const QString &language) override
- void [cancelRequests](#) () override
- QString [getErrorMessage](#) () override
- QMap< QString, QString > [makeQMapFromXML](#) (const QString &xmlData)
- [~BackendGeonamesUSRG](#) () override

### Public Member Functions inherited from [Digikam::RGBBackend](#)

- [RGBBackend](#) (QObject \*const parent)

## Additional Inherited Members

### Signals inherited from [Digikam::RGBBackend](#)

- void **signalRGReady** (const QList< [RGInfo](#) > &)  
*Emitted whenever some items are ready.*

## 6.109.1 Constructor & Destructor Documentation

### 6.109.1.1 BackendGeonamesUSRG()

```
Digikam::BackendGeonamesUSRG::BackendGeonamesUSRG (
    QObject *const parent ) [explicit]
```

Constructor

Parameters

<i>parent</i>	the parent object.
---------------	--------------------

### 6.109.1.2 ~BackendGeonamesUSRG()

```
Digikam::BackendGeonamesUSRG::~~BackendGeonamesUSRG ( ) [override]
```

Destructor

## 6.109.2 Member Function Documentation

### 6.109.2.1 backendName()

```
QString Digikam::BackendGeonamesUSRG::backendName ( ) [override], [virtual]
```

Returns

Backend name.

Reimplemented from [Digikam::RGBBackend](#).

### 6.109.2.2 callRGBBackend()

```
void Digikam::BackendGeonamesUSRG::callRGBBackend (
    const QList< RGInfo > & rgList,
    const QString & language ) [override], [virtual]
```

Takes the coordinate of each image and then connects to Open Street Map's reverse geocoding service.

## Parameters

<i>rgList</i>	A list containing information needed in reverse geocoding process. At this point, it contains only coordinates.
<i>language</i>	The language in which the data will be returned.

Implements [Digikam::RGBackend](#).

### 6.109.2.3 cancelRequests()

```
void Digikam::BackendGeonamesUSRG::cancelRequests ( ) [override], [virtual]
```

Implements [Digikam::RGBackend](#).

### 6.109.2.4 getErrorMessage()

```
QString Digikam::BackendGeonamesUSRG::getErrorMessage ( ) [override], [virtual]
```

## Returns

Error message, if any.

Reimplemented from [Digikam::RGBackend](#).

### 6.109.2.5 makeQMapFromXML()

```
QMap< QString, QString > Digikam::BackendGeonamesUSRG::makeQMapFromXML (
    const QString & xmlData )
```

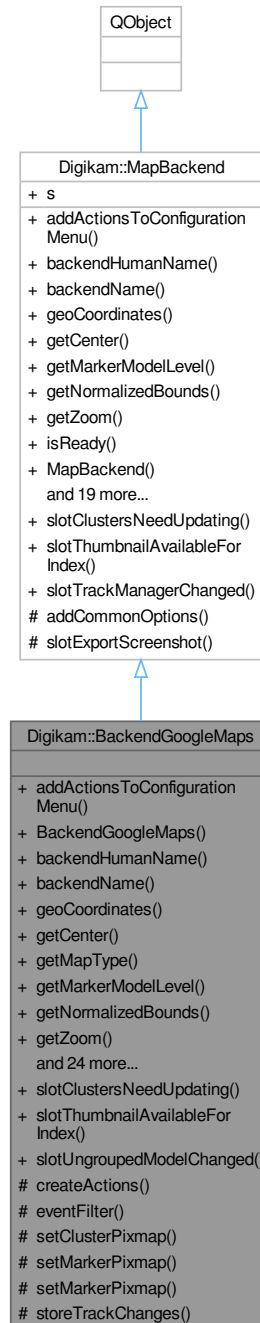
The data is returned from Open Street Map in a XML. This function translates the XML into a QMap.

## Parameters

<i>xmlData</i>	The returned XML.
----------------	-------------------

## 6.110 Digikam::BackendGoogleMaps Class Reference

Inheritance diagram for Digikam::BackendGoogleMaps:



### Public Slots

- void **slotClustersNeedUpdating** () override
- void **slotThumbnailAvailableForIndex** (const QVariant &index, const QPixmap &pixmap) override
- void **slotUngroupedModelChanged** (const int mindex)



## Public Slots inherited from [Digikam::MapBackend](#)

- virtual void **slotClustersNeedUpdating** ()=0
- virtual void **slotThumbnailAvailableForIndex** (const QVariant &index, const QPixmap &pixmap)
- virtual void **slotTrackManagerChanged** ()

## Public Member Functions

- void [addActionToConfigurationMenu](#) (QMenu \*const configurationMenu) override
- **BackendGoogleMaps** (const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > &sharedData, QObject \*const parent=nullptr)
- QString [backendHumanName](#) () const override
- QString [backendName](#) () const override
- bool [geoCoordinates](#) (const QPoint &point, [GeoCoordinates](#) \*const coordinates) const override
- [GeoCoordinates](#) [getCenter](#) () const override
- QString [getMapType](#) () const
- int [getMarkerModelLevel](#) () override
- [GeoCoordinates::PairList](#) [getNormalizedBounds](#) () override
- QString [getZoom](#) () const override
- bool [isReady](#) () const override
- QSize [mapSize](#) () const override
- QWidget \* [mapWidget](#) () override
- void [mapWidgetDocked](#) (const bool state) override
- void [mouseModeChanged](#) () override
- void [readSettingsFromGroup](#) (const KConfigGroup \*const group) override
- void [regionSelectionChanged](#) () override
- void [releaseWidget](#) ([GeofaceInternalWidgetInfo](#) \*const info) override
- void [reload](#) () override
- void [saveSettingsToGroup](#) (KConfigGroup \*const group) override
- bool [screenCoordinates](#) (const [GeoCoordinates](#) &coordinates, QPoint \*const point) override
- void [setActive](#) (const bool state) override
- void [setCenter](#) (const [GeoCoordinates](#) &coordinate) override
- void [setMapType](#) (const QString &newMapType)
- void [setShowMapTypeControl](#) (const bool state)
- void [setShowNavigationControl](#) (const bool state)
- void [setShowScaleControl](#) (const bool state)
- void [setZoom](#) (const QString &newZoom) override
- void [updateActionAvailability](#) () override
- void [updateClusters](#) () override
- void [updateMarkers](#) () override
- void [zoomIn](#) () override
- void [zoomOut](#) () override
- [~BackendGoogleMaps](#) () override

## Public Member Functions inherited from [Digikam::MapBackend](#)

- **MapBackend** (const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > &sharedData, QObject \*const parent)

### Protected Member Functions

- void **createActions** ()
- bool **eventFilter** (QObject \*object, QEvent \*event) override
- void **setClusterPixmap** (const int clusterId, const QPoint &centerPoint, const QPixmap &clusterPixmap)
- void **setMarkerPixmap** (const int modelId, const int markerId, const QPoint &centerPoint, const QPixmap &markerPixmap)
- void **setMarkerPixmap** (const int modelId, const int markerId, const QPoint &centerPoint, const QSize &iconSize, const QUrl &iconUrl)
- void **storeTrackChanges** (const TrackManager::TrackChanges trackChanges)

### Protected Member Functions inherited from [Digikam::MapBackend](#)

- void **addCommonOptions** (QMenu \*const configurationMenu)

### Additional Inherited Members

### Signals inherited from [Digikam::MapBackend](#)

- void **signalBackendReadyChanged** (const QString &backendName)
- void **signalClustersClicked** (const QList &clusterIndices)
- void **signalClustersMoved** (const QList &clusterIndices, const QPair< int, QModelIndex > &snapTarget)
- void **signalMarkersMoved** (const QList &markerIndices)
- void **signalSelectionHasBeenMade** (const Digikam::GeoCoordinates::Pair &coordinates)
- void **signalZoomChanged** (const QString &newZoom)

### Public Attributes inherited from [Digikam::MapBackend](#)

- const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > **s**

### Protected Slots inherited from [Digikam::MapBackend](#)

- void **slotExportScreenshot** ()

## 6.110.1 Constructor & Destructor Documentation

### 6.110.1.1 ~BackendGoogleMaps()

```
Digikam::BackendGoogleMaps::~BackendGoogleMaps ( ) [override]
```

## 6.110.2 Member Function Documentation

### 6.110.2.1 addActionstoConfigurationMenu()

```
void Digikam::BackendGoogleMaps::addActionstoConfigurationMenu (
    QMenu *const configurationMenu ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.2 backendHumanName()

```
QString Digikam::BackendGoogleMaps::backendHumanName ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.3 backendName()

```
QString Digikam::BackendGoogleMaps::backendName ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.4 geoCoordinates()

```
bool Digikam::BackendGoogleMaps::geoCoordinates (
    const QPoint & point,
    GeoCoordinates *const coordinates ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.5 getCenter()

```
GeoCoordinates Digikam::BackendGoogleMaps::getCenter ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.6 getMarkerModelLevel()

```
int Digikam::BackendGoogleMaps::getMarkerModelLevel ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.7 getNormalizedBounds()

```
GeoCoordinates::PairList Digikam::BackendGoogleMaps::getNormalizedBounds ( ) [override],
[virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.8 getZoom()

```
QString Digikam::BackendGoogleMaps::getZoom ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.9 isReady()

```
bool Digikam::BackendGoogleMaps::isReady ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.10 mapSize()

```
QSize Digikam::BackendGoogleMaps::mapSize ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.11 mapWidget()

```
QWidget * Digikam::BackendGoogleMaps::mapWidget ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.12 mapWidgetDocked()

```
void Digikam::BackendGoogleMaps::mapWidgetDocked (
    const bool state ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.13 mouseModeChanged()

```
void Digikam::BackendGoogleMaps::mouseModeChanged ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.14 readSettingsFromGroup()

```
void Digikam::BackendGoogleMaps::readSettingsFromGroup (
    const KConfigGroup *const group ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.15 regionSelectionChanged()

```
void Digikam::BackendGoogleMaps::regionSelectionChanged ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.16 releaseWidget()

```
void Digikam::BackendGoogleMaps::releaseWidget (
    GeoInterfaceInternalWidgetInfo *const info ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.17 reload()

```
void Digikam::BackendGoogleMaps::reload ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.18 saveSettingsToGroup()

```
void Digikam::BackendGoogleMaps::saveSettingsToGroup (
    KConfigGroup *const group ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.19 screenCoordinates()

```
bool Digikam::BackendGoogleMaps::screenCoordinates (
    const GeoCoordinates & coordinates,
    QPoint *const point ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.20 setActive()

```
void Digikam::BackendGoogleMaps::setActive (
    const bool state ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.21 setCenter()

```
void Digikam::BackendGoogleMaps::setCenter (
    const GeoCoordinates & coordinate ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.22 setMarkerPixmap()

```
void Digikam::BackendGoogleMaps::setMarkerPixmap (
    const int modelId,
    const int markerId,
    const QPoint & centerPoint,
    const QSize & iconSize,
    const QUrl & iconUrl ) [protected]
```

### 6.110.2.23 setZoom()

```
void Digikam::BackendGoogleMaps::setZoom (
    const QString & newZoom ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.24 updateActionAvailability()

```
void Digikam::BackendGoogleMaps::updateActionAvailability ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.25 updateClusters()

```
void Digikam::BackendGoogleMaps::updateClusters ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.26 updateMarkers()

```
void Digikam::BackendGoogleMaps::updateMarkers ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.110.2.27 zoomIn()

```
void Digikam::BackendGoogleMaps::zoomIn ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

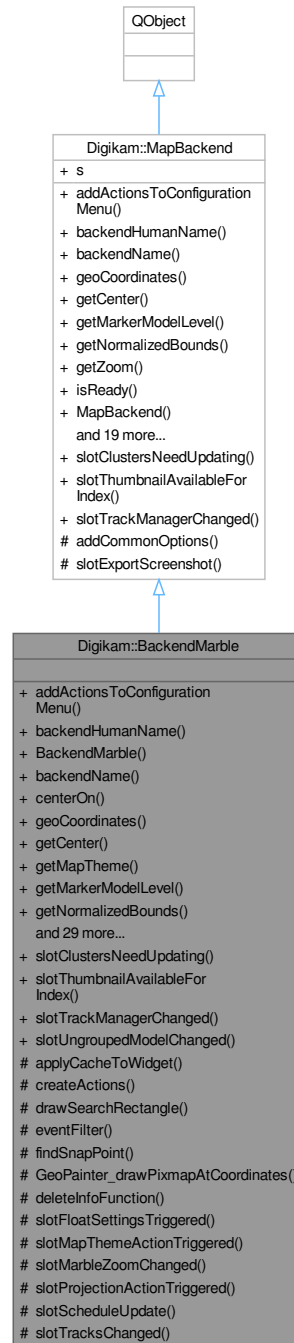
### 6.110.2.28 zoomOut()

```
void Digikam::BackendGoogleMaps::zoomOut ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

## 6.111 Digikam::BackendMarble Class Reference

Inheritance diagram for Digikam::BackendMarble:



### Public Slots

- void **slotClustersNeedUpdating** () override
- void **slotThumbnailAvailableForIndex** (const QVariant &index, const QPixmap &pixmap) override
- void **slotTrackManagerChanged** () override
- void **slotUngroupedModelChanged** (const int index)

## Public Slots inherited from [Digikam::MapBackend](#)

- virtual void **slotClustersNeedUpdating** ()=0
- virtual void **slotThumbnailAvailableForIndex** (const QVariant &index, const QPixmap &pixmap)
- virtual void **slotTrackManagerChanged** ()

## Public Member Functions

- void [addActionToConfigurationMenu](#) (QMenu \*const configurationMenu) override
- QString [backendHumanName](#) () const override
- **BackendMarble** (const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > &sharedData, QObject \*const parent=nullptr)
- QString [backendName](#) () const override
- void [centerOn](#) (const Marble::GeoDataLatLonBox &box, const bool useSaneZoomLevel) override
- bool [geoCoordinates](#) (const QPoint &point, [GeoCoordinates](#) \*const coordinates) const override
- [GeoCoordinates](#) [getCenter](#) () const override
- QString [getMapTheme](#) () const
- int [getMarkerModelLevel](#) () override
- GeoCoordinates::PairList [getNormalizedBounds](#) () override
- QString [getProjection](#) () const
- QString [getZoom](#) () const override
- bool [isReady](#) () const override
- QSize [mapSize](#) () const override
- QWidget \* [mapWidget](#) () override
- void [mapWidgetDocked](#) (const bool state) override
- void [marbleCustomPaint](#) (Marble::GeoPainter \*painter)
- void [mouseModeChanged](#) () override
- void [readSettingsFromGroup](#) (const KConfigGroup \*const group) override
- void [regionSelectionChanged](#) () override
- void [releaseWidget](#) ([GeofaceInternalWidgetInfo](#) \*const info) override
- void [reload](#) () override
- void [saveSettingsToGroup](#) (KConfigGroup \*const group) override
- bool [screenCoordinates](#) (const [GeoCoordinates](#) &coordinates, QPoint \*const point) override
- void [setActive](#) (const bool state) override
- void [setCenter](#) (const [GeoCoordinates](#) &coordinate) override
- void **setMapTheme** (const QString &newMapTheme)
- void **setProjection** (const QString &newProjection)
- void **setShowCompass** (const bool state)
- void **setShowNavigation** (const bool state)
- void **setShowOverviewMap** (const bool state)
- void **setShowScaleBar** (const bool state)
- void [setZoom](#) (const QString &newZoom) override
- void [updateActionAvailability](#) () override
- void [updateClusters](#) () override
- void [updateMarkers](#) () override
- void [zoomIn](#) () override
- void [zoomOut](#) () override
- [~BackendMarble](#) () override

## Public Member Functions inherited from [Digikam::MapBackend](#)

- **MapBackend** (const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > &sharedData, QObject \*const parent)



### Protected Slots

- void **slotFloatSettingsTriggered** (QAction \*action)
- void **slotMapThemeActionTriggered** (QAction \*action)
- void **slotMarbleZoomChanged** ()
- void **slotProjectionActionTriggered** (QAction \*action)
- void **slotScheduleUpdate** ()
- void **slotTracksChanged** (const QList< TrackManager::TrackChanges > &trackChanges)

### Protected Slots inherited from [Digikam::MapBackend](#)

- void **slotExportScreenshot** ()

### Protected Member Functions

- void **applyCacheToWidget** ()
- void **createActions** ()
- void **drawSearchRectangle** (Marble::GeoPainter \*const painter, const GeoCoordinates::Pair &searchRectangle, const bool isOldRectangle)
- bool **eventFilter** (QObject \*object, QEvent \*event) override
- bool **findSnapPoint** (const QPoint &actualPoint, QPoint \*const snapPoint, [GeoCoordinates](#) \*const snapCoordinates, QPair< int, QModelIndex > \*const snapTargetIndex)
- void **GeoPainter\_drawPixmapAtCoordinates** (Marble::GeoPainter \*const painter, const QPixmap &pixmap, const [GeoCoordinates](#) &coordinates, const QPoint &basePoint)

*Replacement for Marble::GeoPainter::drawPixmap which takes a pixel offset.*

### Protected Member Functions inherited from [Digikam::MapBackend](#)

- void **addCommonOptions** (QMenu \*const configurationMenu)

### Static Protected Member Functions

- static void **deleteInfoFunction** ([GeofaceInternalWidgetInfo](#) \*const info)

### Additional Inherited Members

### Signals inherited from [Digikam::MapBackend](#)

- void **signalBackendReadyChanged** (const QString &backendName)
- void **signalClustersClicked** (const QList &clusterIndices)
- void **signalClustersMoved** (const QList &clusterIndices, const QPair< int, QModelIndex > &snapTarget)
- void **signalMarkersMoved** (const QList &markerIndices)
- void **signalSelectionHasBeenMade** (const Digikam::GeoCoordinates::Pair &coordinates)
- void **signalZoomChanged** (const QString &newZoom)

### Public Attributes inherited from [Digikam::MapBackend](#)

- const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > **s**

## 6.111.1 Constructor & Destructor Documentation

### 6.111.1.1 ~BackendMarble()

```
Digikam::BackendMarble::~BackendMarble ( ) [override]
```

## 6.111.2 Member Function Documentation

### 6.111.2.1 addActionstoConfigurationMenu()

```
void Digikam::BackendMarble::addActionstoConfigurationMenu (
    QMenu *const configurationMenu ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.111.2.2 applyCacheToWidget()

```
void Digikam::BackendMarble::applyCacheToWidget ( ) [protected]
```

### 6.111.2.3 backendHumanName()

```
QString Digikam::BackendMarble::backendHumanName ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.111.2.4 backendName()

```
QString Digikam::BackendMarble::backendName ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

### 6.111.2.5 centerOn()

```
void Digikam::BackendMarble::centerOn (
    const Marble::GeoDataLatLonBox & box,
    const bool useSaneZoomLevel ) [override]
```

### 6.111.2.6 eventFilter()

```
bool Digikam::BackendMarble::eventFilter (
    QObject * object,
    QEvent * event ) [override], [protected]
```

**6.111.2.7 geoCoordinates()**

```
bool Digikam::BackendMarble::geoCoordinates (
    const QPoint & point,
    GeoCoordinates *const coordinates ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.8 GeoPainter\_drawPixmapAtCoordinates()**

```
void Digikam::BackendMarble::GeoPainter_drawPixmapAtCoordinates (
    Marble::GeoPainter *const painter,
    const QPixmap & pixmap,
    const GeoCoordinates & coordinates,
    const QPoint & offsetPoint ) [protected]
```

**Parameters**

<i>painter</i>	Marble::GeoPainter on which to draw the pixmap
<i>pixmap</i>	Pixmap to be drawn
<i>coordinates</i>	<a href="#">GeoCoordinates</a> where the image is to be drawn
<i>offsetPoint</i>	Point in the <code>pixmap</code> which should be at <code>coordinates</code>

**6.111.2.9 getCenter()**

```
GeoCoordinates Digikam::BackendMarble::getCenter ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.10 getMarkerModelLevel()**

```
int Digikam::BackendMarble::getMarkerModelLevel ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.11 getNormalizedBounds()**

```
GeoCoordinates::PairList Digikam::BackendMarble::getNormalizedBounds ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.12 getProjection()**

```
QString Digikam::BackendMarble::getProjection ( ) const
```

**6.111.2.13 getZoom()**

```
QString Digikam::BackendMarble::getZoom ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.14 isReady()**

```
bool Digikam::BackendMarble::isReady ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.15 mapSize()**

```
QSize Digikam::BackendMarble::mapSize ( ) const [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.16 mapWidget()**

```
QWidget * Digikam::BackendMarble::mapWidget ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.17 mapWidgetDocked()**

```
void Digikam::BackendMarble::mapWidgetDocked (
    const bool state ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.18 marbleCustomPaint()**

```
void Digikam::BackendMarble::marbleCustomPaint (
    Marble::GeoPainter * painter )
```

**6.111.2.19 mouseModeChanged()**

```
void Digikam::BackendMarble::mouseModeChanged ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.20 readSettingsFromGroup()**

```
void Digikam::BackendMarble::readSettingsFromGroup (
    const KConfigGroup *const group ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.21 regionSelectionChanged()**

```
void Digikam::BackendMarble::regionSelectionChanged ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.22 releaseWidget()**

```
void Digikam::BackendMarble::releaseWidget (
    GeoInterfaceInternalWidgetInfo *const info ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.23 reload()**

```
void Digikam::BackendMarble::reload ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.24 saveSettingsToGroup()**

```
void Digikam::BackendMarble::saveSettingsToGroup (
    KConfigGroup *const group ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.25 screenCoordinates()**

```
bool Digikam::BackendMarble::screenCoordinates (
    const GeoCoordinates & coordinates,
    QPoint *const point ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.26 setActive()**

```
void Digikam::BackendMarble::setActive (
    const bool state ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.27 setCenter()**

```
void Digikam::BackendMarble::setCenter (
    const GeoCoordinates & coordinate ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.28 setZoom()**

```
void Digikam::BackendMarble::setZoom (
    const QString & newZoom ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.29 slotScheduleUpdate**

```
void Digikam::BackendMarble::slotScheduleUpdate ( ) [protected], [slot]
```

**6.111.2.30 updateActionAvailability()**

```
void Digikam::BackendMarble::updateActionAvailability ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.31 updateClusters()**

```
void Digikam::BackendMarble::updateClusters ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.32 updateMarkers()**

```
void Digikam::BackendMarble::updateMarkers ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

**6.111.2.33 zoomIn()**

```
void Digikam::BackendMarble::zoomIn ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

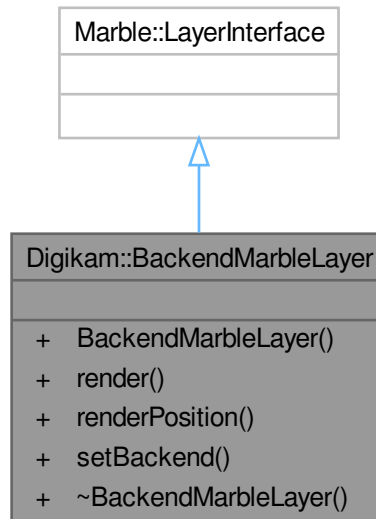
**6.111.2.34 zoomOut()**

```
void Digikam::BackendMarble::zoomOut ( ) [override], [virtual]
```

Implements [Digikam::MapBackend](#).

## 6.112 Digikam::BackendMarbleLayer Class Reference

Inheritance diagram for Digikam::BackendMarbleLayer:



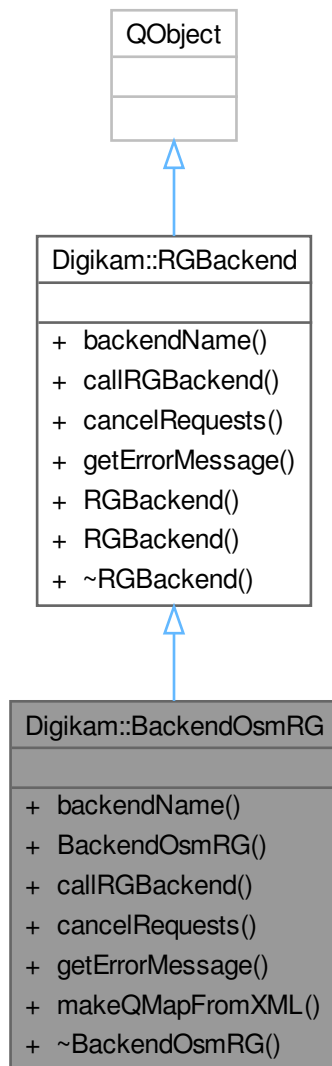
### Public Member Functions

- **BackendMarbleLayer** ([BackendMarble](#) \*const pMarbleBackend)
- bool **render** ([Marble::GeoPainter](#) \*painter, [Marble::ViewportParams](#) \*viewport, const [QString](#) &renderPos=[QLatin1String\("NONE"\)](#), [Marble::GeoSceneLayer](#) \*layer=nullptr) override
- [QStringList](#) **renderPosition** () const override
- void **setBackend** ([BackendMarble](#) \*const pMarbleBackend)

## 6.113 Digikam::BackendOsmRG Class Reference

This class calls Open Street Map's reverse geocoding service.

Inheritance diagram for Digikam::BackendOsmRG:



### Public Member Functions

- `QString backendName ()` override
- `BackendOsmRG (QObject *const parent)`
- `void callRGBBackend (const QList< RGInfo > &rgList, const QString &language)` override
- `void cancelRequests ()` override
- `QString getErrorMessage ()` override
- `QMap< QString, QString > makeQMapFromXML (const QString &xmlData)`
- `~BackendOsmRG ()` override

### Public Member Functions inherited from `Digikam::RGBBackend`

- `RGBBackend (QObject *const parent)`



## Additional Inherited Members

## Signals inherited from [Digikam::RGBackend](#)

- void **signalRGReady** (const QList< [RGInfo](#) > &)  
*Emitted whenever some items are ready.*

## 6.113.1 Constructor & Destructor Documentation

### 6.113.1.1 BackendOsmRG()

```
Digikam::BackendOsmRG::BackendOsmRG (  
    QObject *const parent ) [explicit]
```

Constructor

Parameters

<i>parent</i>	the parent object.
---------------	--------------------

### 6.113.1.2 ~BackendOsmRG()

```
Digikam::BackendOsmRG::~~BackendOsmRG ( ) [override]
```

Destructor

## 6.113.2 Member Function Documentation

### 6.113.2.1 backendName()

```
QString Digikam::BackendOsmRG::backendName ( ) [override], [virtual]
```

Returns

Backend name.

Reimplemented from [Digikam::RGBackend](#).

### 6.113.2.2 callRGBackend()

```
void Digikam::BackendOsmRG::callRGBackend (  
    const QList< RGInfo > & rgList,  
    const QString & language ) [override], [virtual]
```

Takes the coordinate of each image and then connects to Open Street Map's reverse geocoding service.

## Parameters

<i>rgList</i>	A list containing information needed in reverse geocoding process. At this point, it contains only coordinates.
<i>language</i>	The language in which the data will be returned.

Implements [Digikam::RGBackend](#).

### 6.113.2.3 cancelRequests()

```
void Digikam::BackendOsmRG::cancelRequests ( ) [override], [virtual]
```

Implements [Digikam::RGBackend](#).

### 6.113.2.4 getErrorMessage()

```
QString Digikam::BackendOsmRG::getErrorMessage ( ) [override], [virtual]
```

## Returns

Error message, if any.

Reimplemented from [Digikam::RGBackend](#).

### 6.113.2.5 makeQMapFromXML()

```
QMap< QString, QString > Digikam::BackendOsmRG::makeQMapFromXML (
    const QString & xmlData )
```

The data is returned from Open Street Map in a XML. This function translates the XML into a QMap.

## Parameters

<i>xmlData</i>	The returned XML.
----------------	-------------------

## 6.114 Digikam::BalooInfo Class Reference

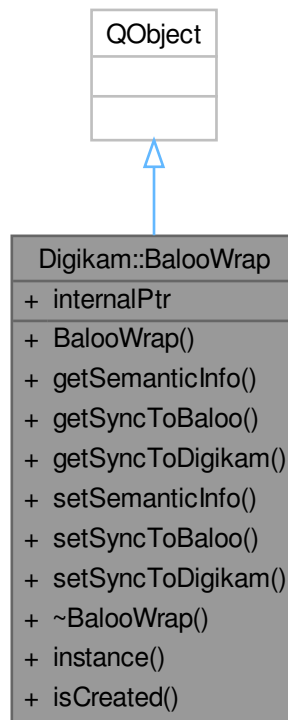
### Public Attributes

- QString **comment**
- int **rating** = -1
- QStringList **tags**

## 6.115 Digikam::BalooWrap Class Reference

The [BalooWrap](#) class is a singleton class which offer functionality for reading and writing image comment, tags and rating from Baloo to digiKam and from digiKam to Baloo.

Inheritance diagram for Digikam::BalooWrap:



### Public Member Functions

- [BalooInfo](#) [getSemanticInfo](#) (const [QUrl](#) &url) const  
*getSemanticInfo* - used by [ItemScanner](#) to retrieve all information tags, comment, rating
- bool [getSyncToBaloo](#) () const
- bool [getSyncToDigikam](#) () const
- void [setSemanticInfo](#) (const [QUrl](#) &url, const [BalooInfo](#) &bInfo)  
*setSemanticInfo* - generic method to set all data from digiKam to Baloo
- void [setSyncToBaloo](#) (bool value)
- void [setSyncToDigikam](#) (bool value)

### Static Public Member Functions

- static [BalooWrap](#) \* [instance](#) ()
- static bool [isCreated](#) ()

## Static Public Attributes

- static `QPointer< BalooWrap > internalPtr = QPointer<BalooWrap>()`  
*internalPtr - singleton implementation*

### 6.115.1 Detailed Description

The singleton functionality is required because it also watches for changes in Baloo and notify digiKam, so it could trigger a scan

### 6.115.2 Member Function Documentation

#### 6.115.2.1 getSemanticInfo()

```
BalooInfo Digikam::BalooWrap::getSemanticInfo (
    const QUrl & url ) const
```

##### Parameters

<i>url</i>	- image url
------------	-------------

##### Returns

- container class for tags, comment, rating

#### 6.115.2.2 setSemanticInfo()

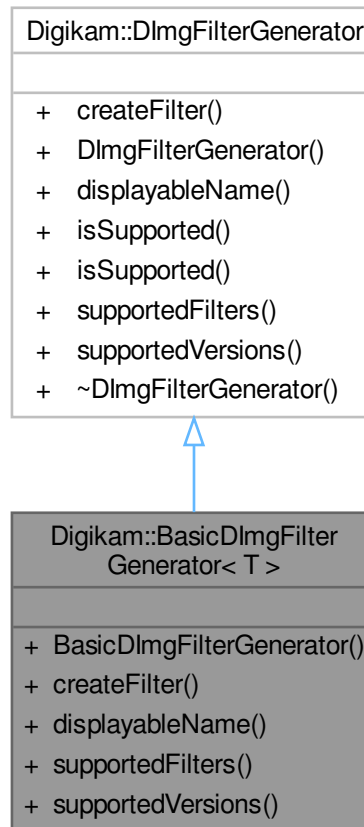
```
void Digikam::BalooWrap::setSemanticInfo (
    const QUrl & url,
    const BalooInfo & bInfo )
```

##### Parameters

<i>url</i>	- image url
<i>bInfo</i>	- container class for tags, comment, rating

## 6.116 Digikam::BasicDImgFilterGenerator< T > Class Template Reference

Inheritance diagram for Digikam::BasicDImgFilterGenerator< T > :



### Public Member Functions

- `BasicDImgFilterGenerator()`=default
- `DImgThreadedFilter * createFilter` (const QString &filterIdentifier, int version) override  
*Create the filter for the given combination of identifier and version.*
- QString `displayName` (const QString &filterIdentifier) override  
*Returns a QString with filter name for displaying in views.*
- QStringList `supportedFilters` () override  
*Returns a list with identifiers of supported filters.*
- QList< int > `supportedVersions` (const QString &filterIdentifier) override  
*Returns a list with the supported versions for the given identifier.*

### Public Member Functions inherited from `Digikam::DImgFilterGenerator`

- virtual bool `isSupported` (const QString &filterIdentifier)  
*Convenience methods.*
- virtual bool `isSupported` (const QString &filterIdentifier, int version)

## 6.116.1 Constructor & Destructor Documentation

### 6.116.1.1 BasicDImgFilterGenerator()

```
template<class T >
Digikam::BasicDImgFilterGenerator< T >::BasicDImgFilterGenerator ( ) [default]
```

A sample implementation for one [DImgThreadedFilter](#) class. The class must provide two static methods, [FilterIdentifier\(\)](#) and [SupportedVersions\(\)](#).

## 6.116.2 Member Function Documentation

### 6.116.2.1 createFilter()

```
template<class T >
DImgThreadedFilter * Digikam::BasicDImgFilterGenerator< T >::createFilter (
    const QString & filterIdentifier,
    int version ) [inline], [override], [virtual]
```

Implements [Digikam::DImgFilterGenerator](#).

### 6.116.2.2 displayableName()

```
template<class T >
QString Digikam::BasicDImgFilterGenerator< T >::displayableName (
    const QString & filterIdentifier ) [inline], [override], [virtual]
```

Implements [Digikam::DImgFilterGenerator](#).

### 6.116.2.3 supportedFilters()

```
template<class T >
QStringList Digikam::BasicDImgFilterGenerator< T >::supportedFilters ( ) [inline], [override],
[virtual]
```

Implements [Digikam::DImgFilterGenerator](#).

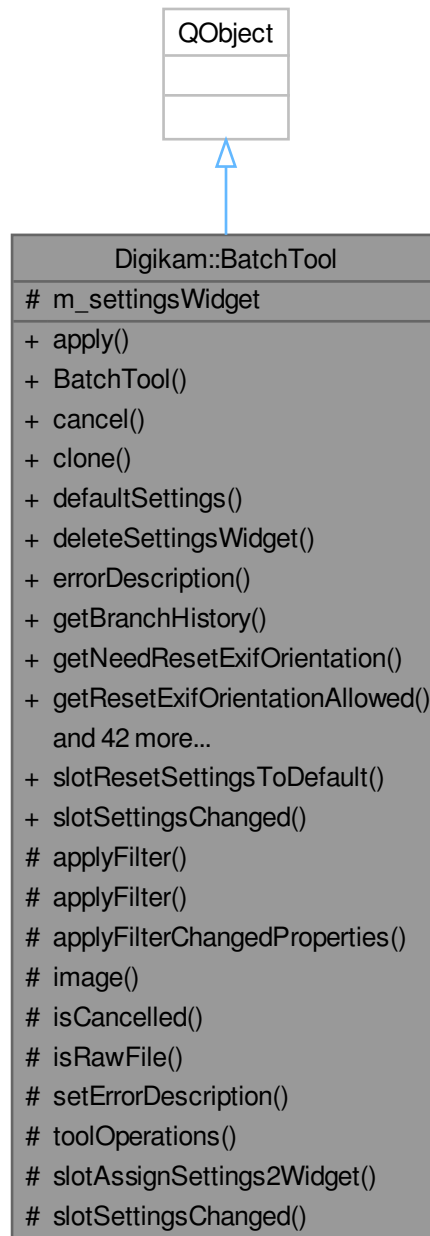
### 6.116.2.4 supportedVersions()

```
template<class T >
QList< int > Digikam::BasicDImgFilterGenerator< T >::supportedVersions (
    const QString & filterIdentifier ) [inline], [override], [virtual]
```

Implements [Digikam::DImgFilterGenerator](#).

## 6.117 Digikam::BatchTool Class Reference

Inheritance diagram for Digikam::BatchTool:



### Public Types

- enum `BatchToolGroup` {  
`BaseTool = 0`, `CustomTool`, `ColorTool`, `EnhanceTool`,  
`TransformTool`, `DecorateTool`, `FiltersTool`, `ConvertTool`,  
`MetadataTool` }

## Public Slots

- void **slotResetSettingsToDefault** ()
- void **slotSettingsChanged** (const [BatchToolSettings](#) &settings)

## Signals

- void [signalAssignSettings2Widget](#) ()
- void **signalSettingsChanged** (const [BatchToolSettings](#) &)
- void **signalVisible** (bool)

## Public Member Functions

- bool [apply](#) ()
- [BatchTool](#) (const QString &name, [BatchToolGroup](#) group, QObject \*const parent=nullptr)
- virtual void [cancel](#) ()
- virtual [BatchTool](#) \* [clone](#) (QObject \*const parent=nullptr) const =0
- virtual [BatchToolSettings](#) [defaultSettings](#) ()=0
- void [deleteSettingsWidget](#) ()
- QString [errorDescription](#) () const
- bool **[getBranchHistory](#)** () const
- bool [getNeedResetExifOrientation](#) () const
- bool [getResetExifOrientationAllowed](#) () const
- [DImg](#) [imageData](#) () const
- [ItemInfo](#) [imageInfo](#) () const
- [QUrl](#) [inputUrl](#) () const
- [IOFileSettings](#) [ioFileSettings](#) () const
- bool **[isLastChainedTool](#)** () const
- bool [loadToDImg](#) () const
- virtual QString [outputSuffix](#) () const
- [QUrl](#) [outputUrl](#) () const
- [DPluginBqm](#) \* **[plugin](#)** () const
- [DRawDecoderSettings](#) [rawDecodingSettings](#) () const
- virtual void [registerSettingsWidget](#) ()
- bool [savefromDImg](#) () const
- void [setBranchHistory](#) (bool branch=true)
- void [setDRawDecoderSettings](#) (const [DRawDecoderSettings](#) &settings)
- void [setImageData](#) (const [DImg](#) &img)
- void [setInputUrl](#) (const [QUrl](#) &inputUrl)
- void [setIOFileSettings](#) (const [IOFileSettings](#) &settings)
- void [setItemInfo](#) (const [ItemInfo](#) &info)
- void [setLastChainedTool](#) (bool last)
- void [setNeedResetExifOrientation](#) (bool reset)
- void [setOutputUrl](#) (const [QUrl](#) &outputUrl)
- void [setOutputUrlFromInputUrl](#) ()
- void **[setPlugin](#)** ([DPluginBqm](#) \*const plugin)
- void [setRawLoadingRules](#) ([QueueSettings::RawLoadingRule](#) rule)
- void [setResetExifOrientationAllowed](#) (bool reset)
- void [setSaveAsNewVersion](#) (bool fork=true)
- void [setSettings](#) (const [BatchToolSettings](#) &settings)
- [BatchToolSettings](#) **[settings](#)** () const
- [QWidget](#) \* [settingsWidget](#) () const



Settings widget management. NOTE: do not use these methods in multi-threading part ([ActionThread](#)), only in main thread (GUI)

- void [setToolDescription](#) (const QString &toolDescription)
- void **setToolIcon** (const QIcon &icon)
- void [setToolIconName](#) (const QString &iconName)
- void [setToolTitle](#) (const QString &toolTitle)
- void [setWorkingUrl](#) (const QUrl &workingUrl)
- QString **toolDescription** () const
- [BatchToolGroup](#) **toolGroup** () const
- QString [toolGroupToString](#) () const
- QIcon **toolIcon** () const
- QString **toolTitle** () const
- virtual int [toolVersion](#) () const
- QUrl **workingUrl** () const

### Protected Slots

- virtual void [slotAssignSettings2Widget](#) ()=0
- virtual void **slotSettingsChanged** ()=0

### Protected Member Functions

- void **applyFilter** ([DImgBuiltinFilter](#) \*const filter)
- void [applyFilter](#) ([DImgThreadedFilter](#) \*const filter)
- void **applyFilterChangedProperties** ([DImgThreadedFilter](#) \*const filter)
- [DImg](#) & [image](#) () const
- bool [isCancelled](#) () const
- bool [isRawFile](#) (const QUrl &url) const
- void [setErrorDescription](#) (const QString &errmsg)
- virtual bool [toolOperations](#) ()=0

### Protected Attributes

- QWidget \* [m\\_settingsWidget](#) = nullptr

## 6.117.1 Member Enumeration Documentation

### 6.117.1.1 BatchToolGroup

```
enum Digikam::BatchTool::BatchToolGroup
```

#### Enumerator

BaseTool	digikam core tools.
CustomTool	List of tools grouped and customized by users.
ColorTool	Tools to manage image colors (Curves, BCG, etc...)
EnhanceTool	Tools to enhance images (NR, sharp, etc...)
TransformTool	Tools to transform images geometry (resize, rotate, flip, etc...)
DecorateTool	Tools to decorate images (Border, watermark, etc...)
FiltersTool	Tools to apply filters and special effects (film grain, BlurFx, etc...)
ConvertTool	Tools to convert images format (PNG, JPEG, TIFF, etc...)
MetadataTool	Tools to play with metadata.

## 6.117.2 Constructor & Destructor Documentation

### 6.117.2.1 BatchTool()

```
Digikam::BatchTool::BatchTool (
    const QString & name,
    BatchToolGroup group,
    QObject *const parent = nullptr ) [explicit]
```

Tool data and properties management. NOTE: these methods can be used safely in multi-threading part ([ActionThread](#)).

## 6.117.3 Member Function Documentation

### 6.117.3.1 apply()

```
bool Digikam::BatchTool::apply ( )
```

Apply all change to perform by this tool. This method call customized [toolOperations\(\)](#).

### 6.117.3.2 applyFilter()

```
void Digikam::BatchTool::applyFilter (
    DImgThreadedFilter *const filter ) [protected]
```

Use this if you have a filter ready to run. Will call [startFilterDirectly](#) and apply the result to [image\(\)](#).

### 6.117.3.3 cancel()

```
void Digikam::BatchTool::cancel ( ) [virtual]
```

Re-implement this method is you want customize cancellation of tool, for ex. to call a dedicated method to kill sub-threads parented to this tool instance. Unforget to call parent [BatchTool::cancel\(\)](#) method in your customized implementation.

### 6.117.3.4 clone()

```
virtual BatchTool * Digikam::BatchTool::clone (
    QObject *const parent = nullptr ) const [pure virtual]
```

Clone this tool without to create settings widget. It's a safe construction of tools instance used in multithreading ([ActionThread](#)) to process items in parallel.

### 6.117.3.5 defaultSettings()

```
virtual BatchToolSettings Digikam::BatchTool::defaultSettings ( ) [pure virtual]
```

Re-implement this method to initialize Settings Widget value with default settings.

### 6.117.3.6 deleteSettingsWidget()

```
void Digikam::BatchTool::deleteSettingsWidget ( )
```

Delete dedicated settings widget registered with [registerSettingsWidget\(\)](#).

### 6.117.3.7 errorDescription()

```
QString Digikam::BatchTool::errorDescription ( ) const
```

Get description of an error which appear during [apply\(\)](#) method.

### 6.117.3.8 getNeedResetExifOrientation()

```
bool Digikam::BatchTool::getNeedResetExifOrientation ( ) const
```

Returns true if the Exif orientation tag should be reset after tool operation

### 6.117.3.9 getResetExifOrientationAllowed()

```
bool Digikam::BatchTool::getResetExifOrientationAllowed ( ) const
```

Returns true if the Exif orientation tag is allowed to be reset after tool operation

### 6.117.3.10 image()

```
DImg & Digikam::BatchTool::image ( ) const [protected]
```

Return a reference of internal [DImg](#) container used to modify image data.

### 6.117.3.11 ioFileSettings()

```
IOFileSettings Digikam::BatchTool::ioFileSettings ( ) const
```

Return IOFile settings used during tool operations.

### 6.117.3.12 isCancelled()

```
bool Digikam::BatchTool::isCancelled ( ) const [protected]
```

Return true if [cancel\(\)](#) have been called. Use this method to stop loop in your [toolOperations\(\)](#) implementation.

### 6.117.3.13 isRawFile()

```
bool Digikam::BatchTool::isRawFile (
    const QUrl & url ) const [protected]
```

Method to check if file pointed by url is a RAW image

#### 6.117.3.14 loadToDImg()

```
bool Digikam::BatchTool::loadToDImg ( ) const
```

Load image data using input Url set by [setInputUrl\(\)](#) to instance of internal [DImg](#) container.

#### 6.117.3.15 outputSuffix()

```
QString Digikam::BatchTool::outputSuffix ( ) const [virtual]
```

Re-implement this method if tool change file extension during batch process (ex: "png"). Typically, this is used with tool which convert to new file format. This method return and empty string by default.

#### 6.117.3.16 rawDecodingSettings()

```
DRawDecoderSettings Digikam::BatchTool::rawDecodingSettings ( ) const
```

Return RAW decoding settings used during tool operations.

#### 6.117.3.17 registerSettingsWidget()

```
void Digikam::BatchTool::registerSettingsWidget ( ) [virtual]
```

[Setup](#) dedicated settings widget. Default implementation assign no settings view (a message label is just displayed). You need to call default implementation in your child class to init default signals and slots connections, after to have instanced your dedicated settings widget.

#### 6.117.3.18 savefromDImg()

```
bool Digikam::BatchTool::savefromDImg ( ) const
```

Save image data from instance of internal [DImg](#) container using :

- output Url set by [setOutputUrl\(\)](#) or [setOutputUrlFromInputUrl\(\)](#)
- output file format set by [outputSuffix\(\)](#). If this one is empty, format of original image is used instead.

#### 6.117.3.19 setBranchHistory()

```
void Digikam::BatchTool::setBranchHistory (
    bool branch = true )
```

Applies only when the file is actually saved on disk, and takes the history since the loading from disk to set the first added step as creating a branch.

### 6.117.3.20 setDRawDecoderSettings()

```
void Digikam::BatchTool::setDRawDecoderSettings (
    const DRawDecoderSettings & settings )
```

Set-up RAW decoding settings no use during tool operations.

### 6.117.3.21 setErrorDescription()

```
void Digikam::BatchTool::setErrorDescription (
    const QString & errmsg ) [protected]
```

Set string to describe an error which appear during [apply\(\)](#) method.

### 6.117.3.22 setImageData()

```
void Digikam::BatchTool::setImageData (
    const DImg & img )
```

Manage instance of current image data container loaded by this tool.

### 6.117.3.23 setInputUrl()

```
void Digikam::BatchTool::setInputUrl (
    const QUrl & inputUrl )
```

Manage current input url processed by this tool.

### 6.117.3.24 setIOFileSettings()

```
void Digikam::BatchTool::setIOFileSettings (
    const IOFileSettings & settings )
```

Set-up IOFile settings no use during tool operations.

### 6.117.3.25 setItemInfo()

```
void Digikam::BatchTool::setItemInfo (
    const ItemInfo & info )
```

Manage instance of current image info loaded by this tool.

### 6.117.3.26 setLastChainedTool()

```
void Digikam::BatchTool::setLastChainedTool (
    bool last )
```

Manage flag properties to indicate if this tool is last one to process on current item.

### 6.117.3.27 `setNeedResetExifOrientation()`

```
void Digikam::BatchTool::setNeedResetExifOrientation (
    bool reset )
```

Set that the Exif orientation flag should be reset to NORMAL after tool operation

### 6.117.3.28 `setOutputUrl()`

```
void Digikam::BatchTool::setOutputUrl (
    const QUrl & outputUrl )
```

Manage current output url processed by this tool.

### 6.117.3.29 `setOutputUrlFromInputUrl()`

```
void Digikam::BatchTool::setOutputUrlFromInputUrl ( )
```

Set output url using input url content + annotation based on time stamp + file extension defined by `outputSuffix()`. if `outputSuffix()` return null, file extension is the same than original.

### 6.117.3.30 `setRawLoadingRules()`

```
void Digikam::BatchTool::setRawLoadingRules (
    QueueSettings::RawLoadingRule rule )
```

Set that RAW files loading rule to use (demosaiicing or JPEG embedded).

### 6.117.3.31 `setResetExifOrientationAllowed()`

```
void Digikam::BatchTool::setResetExifOrientationAllowed (
    bool reset )
```

Set that the Exif orientation flag is allowed be reset to NORMAL after tool operation

### 6.117.3.32 `setSaveAsNewVersion()`

```
void Digikam::BatchTool::setSaveAsNewVersion (
    bool fork = true )
```

Sets if the history added by tools shall be made a branch (new version).

### 6.117.3.33 `setSettings()`

```
void Digikam::BatchTool::setSettings (
    const BatchToolSettings & settings )
```

Manage settings values to tool. See BatchToolSettings container for details.

### 6.117.3.34 settingsWidget()

```
QWidget * Digikam::BatchTool::settingsWidget ( ) const
```

Return dedicated settings widget registered with [registerSettingsWidget\(\)](#).

### 6.117.3.35 setToolDescription()

```
void Digikam::BatchTool::setToolDescription (
    const QString & toolDescription )
```

Manage Tool description.

### 6.117.3.36 setToolIconName()

```
void Digikam::BatchTool::setToolIconName (
    const QString & iconName )
```

Manage Tool icon name.

### 6.117.3.37 setToolTitle()

```
void Digikam::BatchTool::setToolTitle (
    const QString & toolTitle )
```

Manage Tool title.

### 6.117.3.38 setWorkingUrl()

```
void Digikam::BatchTool::setWorkingUrl (
    const QUrl & workingUrl )
```

Manage current working url used by this tool to process items.

### 6.117.3.39 signalAssignSettings2Widget

```
void Digikam::BatchTool::signalAssignSettings2Widget ( ) [signal]
```

Only used internally. See [registerSettingsWidget\(\)](#) implementation.

### 6.117.3.40 slotAssignSettings2Widget

```
virtual void Digikam::BatchTool::slotAssignSettings2Widget ( ) [protected], [pure virtual],
[slot]
```

Re-implement this method to customize how all settings values must be assigned to settings widget. This method is called by [setSettings\(\)](#) through [signalAssignSettings2Widget\(\)](#).

### 6.117.3.41 toolGroup()

```
BatchTool::BatchToolGroup Digikam::BatchTool::toolGroup ( ) const
```

Return group of tool. See BatchToolGroup enum for details.

### 6.117.3.42 toolGroupToString()

```
QString Digikam::BatchTool::toolGroupToString ( ) const
```

Return group of tool name as string.

### 6.117.3.43 toolOperations()

```
virtual bool Digikam::BatchTool::toolOperations ( ) [protected], [pure virtual]
```

Re-implement this method to customize all batch operations done by this tool. This method is called by [apply\(\)](#).

### 6.117.3.44 toolVersion()

```
virtual int Digikam::BatchTool::toolVersion ( ) const [inline], [virtual]
```

Return version of tool. By default, ID is 1. Re-implement this method and increase this ID when tool settings change.

## 6.117.4 Member Data Documentation

### 6.117.4.1 m\_settingsWidget

```
QWidget* Digikam::BatchTool::m_settingsWidget = nullptr [protected]
```

Host settings widget instance.

## 6.118 Digikam::BatchToolSet Class Reference

### Public Member Functions

- bool [operator==](#) (const [BatchToolSet](#) &set) const

### Public Attributes

- [BatchTool::BatchToolGroup](#) **group** = [BatchTool::BaseTool](#)
- int **index** = -1  
*Tool identifier data. Index is tool ID from assigned list.*
- QString **name**
- [BatchToolSettings](#) **settings**  
*Settings hosted in this container.*
- int **version** = 0



### 6.118.1 Detailed Description

A container of associated batch tool and settings.

### 6.118.2 Member Function Documentation

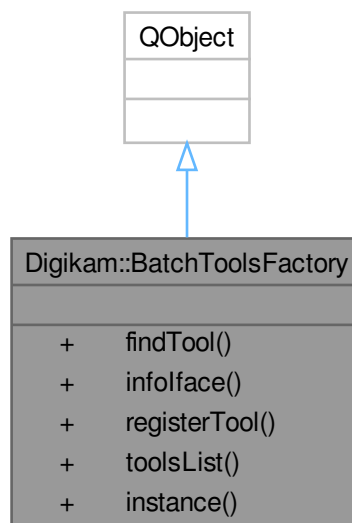
#### 6.118.2.1 operator==( )

```
bool Digikam::BatchToolSet::operator==(
    const BatchToolSet & set ) const
```

Equality operator which check index, version, name, and group data. Settings member is ignored.

## 6.119 Digikam::BatchToolsFactory Class Reference

Inheritance diagram for Digikam::BatchToolsFactory:



### Public Member Functions

- `BatchTool * findTool` (const QString &name, `BatchTool::BatchToolGroup` group) const
- `BqmlInfoface * infoface` () const
- void `registerTool` (`BatchTool *const` tool)
- `BatchToolsList toolsList` () const

### Static Public Member Functions

- static `BatchToolsFactory * instance` ()

## Friends

- class **BatchToolsFactoryCreator**

## 6.120 Digikam::BCGContainer Class Reference

### Public Member Functions

- bool **isDefault** () const
- bool **operator==** (const [BCGContainer](#) &other) const
- void **writeToFilterAction** ([FilterAction](#) &action, const QString &prefix=QString()) const

### Static Public Member Functions

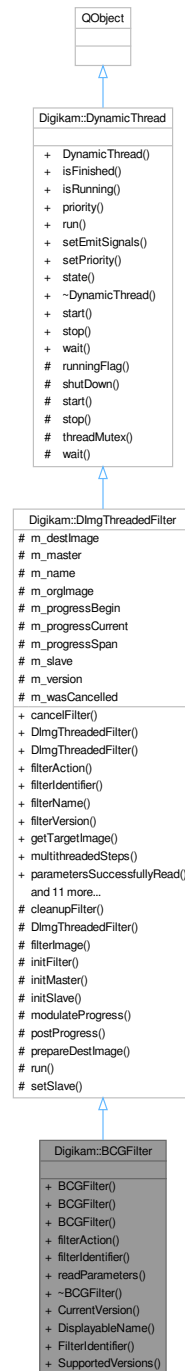
- static [BCGContainer](#) **fromFilterAction** (const [FilterAction](#) &action, const QString &prefix=QString())

### Public Attributes

- double **brightness** = 0.0
- int **channel** = LuminosityChannel
- double **contrast** = 0.0
- double **gamma** = 1.0

## 6.121 Digikam::BCGFilter Class Reference

Inheritance diagram for Digikam::BCGFilter:



### Public Member Functions

- **BCGFilter** (const [BCGContainer](#) &settings, [DimgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100)

- **BCGFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [BCGContainer](#) &settings=[BCGContainer](#)())
- **BCGFilter** (QObject \*const parent=nullptr)
- [FilterAction](#) filterAction () override
- QString [filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int [CurrentVersion](#) ()
- static QString [DisplayableName](#) ()
- static QString [FilterIdentifier](#) ()
- static QList< int > [SupportedVersions](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.121.1 Member Function Documentation

### 6.121.1.1 filterAction()

`FilterAction` Digikam::BCGFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.121.1.2 filterIdentifier()

`QString` Digikam::BCGFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

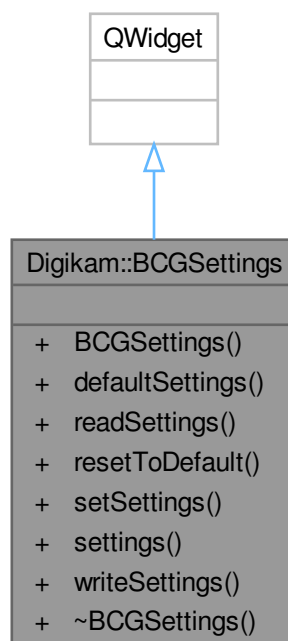
### 6.121.1.3 readParameters()

```
void Digikam::BCGFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.122 Digikam::BCGSettings Class Reference

Inheritance diagram for Digikam::BCGSettings:



## Signals

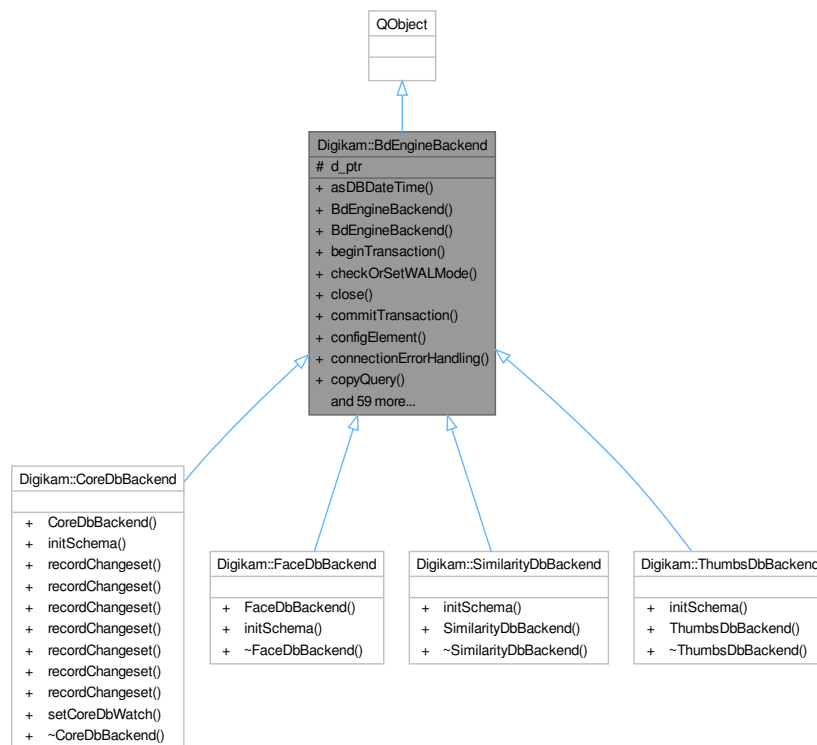
- void **signalSettingsChanged** ()

## Public Member Functions

- **BCGSettings** (QWidget \*const parent)
- **BCGContainer defaultSettings** () const
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const **BCGContainer** &settings)
- **BCGContainer settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.123 Digikam::BdEngineBackend Class Reference

Inheritance diagram for Digikam::BdEngineBackend:



## Classes

- class **QueryState**

## Public Types

- enum **DbType** { **SQLite** , **MySQL** }
- enum **QueryOperationStatus** { **ExecuteNormal** , **Wait** , **AbortQueries** }
- enum **QueryStateEnum** { **NoErrors** , **SQLException** , **ConnectionError** }
- enum **Status** { **Unavailable** , **Open** , **OpenSchemaChecked** }

## Public Member Functions

- QDateTime **asDBDateTime** (const QDateTime &dateTime) const
- **BdEngineBackend** (const QString &backendName, **DbEngineLocking** \*const locking)
- **BdEngineBackend** (const QString &backendName, **DbEngineLocking** \*const locking, **BdEngineBackendPrivate** &dd)
- **BdEngineBackend::QueryState** **beginTransaction** ()
- bool **checkOrSetWALMode** ()
- void **close** ()
- **BdEngineBackend::QueryState** **commitTransaction** ()
- **DbEngineConfigSettings** **configElement** () const
- bool **connectionErrorHandling** (int retries)
- **DbEngineSqlQuery** **copyQuery** (const **DbEngineSqlQuery** &old)
- DbType **databaseType** () const
- bool **exec** (**DbEngineSqlQuery** &query)
- bool **execBatch** (**DbEngineSqlQuery** &query)
- **QueryState** **execDBAction** (const **DbEngineAction** &action, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- **QueryState** **execDBAction** (const **DbEngineAction** &action, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- **QueryState** **execDBAction** (const QString &action, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- **QueryState** **execDBAction** (const QString &action, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- QSqlQuery **execDBActionQuery** (const **DbEngineAction** &action, const QMap< QString, QVariant > &bindingMap)
- QSqlQuery **execDBActionQuery** (const QString &action, const QMap< QString, QVariant > &bindingMap)
- **QueryState** **execDirectSql** (const QString &query)
- **QueryState** **execDirectSqlWithResult** (const QString &query, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- **DbEngineSqlQuery** **execQuery** (const QString &sql)
- **DbEngineSqlQuery** **execQuery** (const QString &sql, const QList< QVariant > &boundValues)
- **DbEngineSqlQuery** **execQuery** (const QString &sql, const QMap< QString, QVariant > &bindingMap)
- **DbEngineSqlQuery** **execQuery** (const QString &sql, const QVariant &boundValue1)
- **DbEngineSqlQuery** **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2)
- **DbEngineSqlQuery** **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- **DbEngineSqlQuery** **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)
- void **execQuery** (**DbEngineSqlQuery** &preparedQuery, const QList< QVariant > &boundValues)
- void **execQuery** (**DbEngineSqlQuery** &preparedQuery, const QVariant &boundValue1)
- void **execQuery** (**DbEngineSqlQuery** &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2)
- void **execQuery** (**DbEngineSqlQuery** &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- void **execQuery** (**DbEngineSqlQuery** &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)



- [QueryState execSql](#) (const QString &sql, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (DbEngineSqlQuery &preparedQuery, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (DbEngineSqlQuery &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (DbEngineSqlQuery &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (DbEngineSqlQuery &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (DbEngineSqlQuery &preparedQuery, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (DbEngineSqlQuery &preparedQuery, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execUpsertDBAction](#) (const DbEngineAction &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [QueryState execUpsertDBAction](#) (const QString &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [DbEngineAction getDBAction](#) (const QString &actionName) const
- [DbEngineSqlQuery getQuery](#) ()
- [QueryState handleQueryResult](#) (DbEngineSqlQuery &query, QList< QVariant > \*const values, QVariant \*const lastInsertId)
- bool [isCompatible](#) (const DbEngineParameters &parameters)
- bool [isInTransaction](#) () const
- bool [isOpen](#) () const
- bool [isReady](#) () const
- QString [lastError](#) ()
- QSqlError [lastSQLError](#) ()
- int [maximumBoundValues](#) () const
- bool [open](#) (const DbEngineParameters &parameters)
- [DbEngineSqlQuery prepareQuery](#) (const QString &sql)
- bool [queryErrorHandling](#) (DbEngineSqlQuery &query, int retries)
- QList< QVariant > [readToList](#) (DbEngineSqlQuery &query)
- void [rollbackTransaction](#) ()
- void [setDbEngineErrorHandler](#) (DbEngineErrorHandler \*const handler)
- void [setForeignKeyChecks](#) (bool check)
- [Status status](#) () const
- QStringList [tables](#) ()
- bool [transactionErrorHandling](#) (const QSqlError &lastError, int retries)

## Protected Attributes

- `BdEngineBackendPrivate` \*const `d_ptr` = nullptr

## 6.123.1 Member Enumeration Documentation

### 6.123.1.1 QueryStateEnum

```
enum Digikam::BdEngineBackend::QueryStateEnum
```

#### Enumerator

NoErrors	No errors occurred while executing the query.
SQLException	An SQLException has occurred while executing the query.
ConnectionError	An connection error has occurred while executing the query.

### 6.123.1.2 Status

```
enum Digikam::BdEngineBackend::Status
```

#### Enumerator

Unavailable	The database is not available, because it has not been opened yet or because of an error condition.
Open	The database is open. It has not been verified that the schema is up to date. This status is sufficient for use in a context where it can be assumed that the necessary schema check has been carried out by a master process.
OpenSchemaChecked	The database is open, and it has been verified that the schema is up to date, or the schema has been updated.

## 6.123.2 Constructor & Destructor Documentation

### 6.123.2.1 BdEngineBackend()

```
Digikam::BdEngineBackend::BdEngineBackend (
    const QString & backendName,
    DbEngineLocking *const locking ) [explicit]
```

Creates a database backend. The backend name is an arbitrary string that shall be unique for this backend object. It will be used to create unique connection names per backend and thread.

## 6.123.3 Member Function Documentation

### 6.123.3.1 asDBDateTime()

```
QDateTime Digikam::BdEngineBackend::asDBDateTime (
    const QDateTime & dateTime ) const
```

Depending on the database backend return a local or UTC date format. SQLite: local date format MySQL: UTC date format

### 6.123.3.2 beginTransaction()

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::beginTransaction ( )
```

Begin a database transaction

### 6.123.3.3 checkOrSetWALMode()

```
bool Digikam::BdEngineBackend::checkOrSetWALMode ( )
```

Check or set WAL mode for SQLite database if enabled in settings.

#### Returns

true the WAL mode is confirmed enabled.

### 6.123.3.4 close()

```
void Digikam::BdEngineBackend::close ( )
```

Close the database connection. Shall only be called from the thread that called [open\(\)](#).

### 6.123.3.5 commitTransaction()

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::commitTransaction ( )
```

Commit the current database transaction

### 6.123.3.6 configElement()

```
DbEngineConfigSettings Digikam::BdEngineBackend::configElement ( ) const
```

Return config read from XML, corresponding to this backend's database type.

### 6.123.3.7 connectionErrorHandling()

```
bool Digikam::BdEngineBackend::connectionErrorHandling (
    int retries )
```

Called when an attempted connection to the database failed. If it returns true, retry; if it returns false, bail out. Pass the number of connection retries to help with some decisions.

### 6.123.3.8 copyQuery()

```
DbEngineSqlQuery Digikam::BdEngineBackend::copyQuery (
    const DbEngineSqlQuery & old )
```

Creates a faithful copy of the passed query, with the current db connection.

**6.123.3.9 databaseType()**

```
BdEngineBackend::DbType Digikam::BdEngineBackend::databaseType ( ) const
```

Return the database type.

**6.123.3.10 exec()**

```
bool Digikam::BdEngineBackend::exec (
    DbEngineSqlQuery & query )
```

Calls exec/execBatch on the query, and handles debug output if something went wrong

**6.123.3.11 execDBAction() [1/2]**

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::execDBAction (
    const DbEngineAction & action,
    const QMap< QString, QVariant > & bindingMap,
    QList< QVariant > *const values = nullptr,
    QVariant *const lastInsertId = nullptr )
```

Performs the database action on the current database. Queries by the specified parameters can have named parameters which are substituted with values from the bindingMap parameter. The result values (if any) are stored within the values list.

**6.123.3.12 execDBAction() [2/2]**

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::execDBAction (
    const DbEngineAction & action,
    QList< QVariant > *const values = nullptr,
    QVariant *const lastInsertId = nullptr )
```

Performs the database action on the current database. Queries by the specified parameters mustn't have named parameters. The result values (if any) are stored within the values list.

**6.123.3.13 execDBActionQuery()**

```
QString Digikam::BdEngineBackend::execDBActionQuery (
    const DbEngineAction & action,
    const QMap< QString, QVariant > & bindingMap )
```

Performs the database action on the current database. Queries by the specified parameters can have named parameters which are substituted with values from the bindingMap parameter. The result values (if any) are stored within the values list. This method returns the last query, which is used to handle special cases.

**6.123.3.14 execDirectSql()**

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::execDirectSql (
    const QString & query )
```

Calls exec on the query, and handles debug output if something went wrong. The query is not prepared, which can be fail in certain situations (e.g. trigger statements on QMYSQL).

### 6.123.3.15 execDirectSqlWithResult()

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::execDirectSqlWithResult (
    const QString & query,
    QList< QVariant > *const values = nullptr,
    QVariant *const lastInsertId = nullptr )
```

Calls exec on the query, and handles debug output if something went wrong. The query is not prepared, which can fail in certain situations (e.g. trigger statements on QMYSQL).

### 6.123.3.16 execQuery() [1/3]

```
DbEngineSqlQuery Digikam::BdEngineBackend::execQuery (
    const QString & sql )
```

Executes the statement and returns the query object. Methods are provided for up to four bound values (positional binding), or for a list of bound values.

### 6.123.3.17 execQuery() [2/3]

```
DbEngineSqlQuery Digikam::BdEngineBackend::execQuery (
    const QString & sql,
    const QMap< QString, QVariant > & bindingMap )
```

Method which accept a hashmap with key, values which are used for named binding

### 6.123.3.18 execQuery() [3/3]

```
void Digikam::BdEngineBackend::execQuery (
    DbEngineSqlQuery & preparedQuery,
    const QVariant & boundValue1 )
```

Binds the values and executes the prepared query.

### 6.123.3.19 execSql() [1/2]

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::execSql (
    const QString & sql,
    const QMap< QString, QVariant > & bindingMap,
    QList< QVariant > *const values = nullptr,
    QVariant *const lastInsertId = nullptr )
```

Method which accepts a map for named binding. For special cases it's also possible to add a [DbEngineActionType](#) which wraps another data object (also lists or maps) which can be used as field entry or as value (where it's prepared with positional binding). See more on [DbEngineActionType](#) class. If the wrapped data object is an instance of list, then the elements are separated by comma. If the wrapped data object is an instance of map, then the elements are inserted in the following way: key1=value1, key2=value2,...,keyN=valueN.

**6.123.3.20 execSql() [2/2]**

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::execSql (
    const QString & sql,
    QList< QVariant > *const values = nullptr,
    QVariant *const lastInsertId = nullptr )
```

Executes the SQL statement, and write the returned data into the values list. If you are not interested in the returned data, set values to 0. Methods are provided for up to four bound values (positional binding), or for a list of bound values. If you want the last inserted id (and your query is suitable), set lastInsertId to the address of a QVariant. Additionally, methods are provided for prepared statements.

**6.123.3.21 execUpsertDBAction()**

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::execUpsertDBAction (
    const DbEngineAction & action,
    const QVariant & id,
    const QStringList & fieldNames,
    const QList< QVariant > & values )
```

Performs a special DBAction that is usually needed to "INSERT or UPDATE" entries in a table. The corresponding DBAction must contain exactly the named parameters :id, :fieldValueList, :fieldList and :valueList. You pass the value to be bound to the ":id" field, then two lists of the same size: The first containing the field names, the second one containing the values as QVariants ready for binding.

**6.123.3.22 getDBAction()**

```
DbEngineAction Digikam::BdEngineBackend::getDBAction (
    const QString & actionName ) const
```

Returns a database action with name, specified in actionName, for the current database.

**6.123.3.23 getQuery()**

```
DbEngineSqlQuery Digikam::BdEngineBackend::getQuery ( )
```

Creates an empty query object waiting for the statement

**6.123.3.24 handleQueryResult()**

```
BdEngineBackend::QueryState Digikam::BdEngineBackend::handleQueryResult (
    DbEngineSqlQuery & query,
    QList< QVariant > *const values,
    QVariant *const lastInsertId )
```

Checks if there was a connection error. If so [BdEngineBackend::ConnectionError](#) is returned. If not, the values are extracted from the query and inserted in the values list, the last insertion id is taken from the query and [BdEngineBackend::NoErrors](#) is returned.

### 6.123.3.25 isCompatible()

```
bool Digikam::BdEngineBackend::isCompatible (
    const DbEngineParameters & parameters )
```

Checks if the parameters can be used for this database backend.

### 6.123.3.26 isInTransaction()

```
bool Digikam::BdEngineBackend::isInTransaction ( ) const
```

Returns if the database is in a different thread in a transaction. Note that a transaction does not require holding [CoreDbAccess](#). Note that this does not give information about other processes locking the database.

### 6.123.3.27 lastError()

```
QString Digikam::BdEngineBackend::lastError ( )
```

Returns a description of the last error that occurred on this database. Use [CoreDbAccess::lastError](#) for errors presented to the user. This error will be included in that message. It may be empty.

### 6.123.3.28 lastSQLError()

```
QString Digikam::BdEngineBackend::lastSQLError ( )
```

Returns the last error that occurred on this database. Use [CoreDbAccess::lastError](#) for errors presented to the user. It may be empty.

### 6.123.3.29 maximumBoundValues()

```
int Digikam::BdEngineBackend::maximumBoundValues ( ) const
```

Returns the maximum number of bound parameters allowed per query. This value depends on the database engine.

### 6.123.3.30 open()

```
bool Digikam::BdEngineBackend::open (
    const DbEngineParameters & parameters )
```

Open the database connection.

#### Returns

true on success

### 6.123.3.31 prepareQuery()

```
DbEngineSqlQuery Digikam::BdEngineBackend::prepareQuery (
    const QString & sql )
```

Creates a query object prepared with the statement, waiting for bound values

### 6.123.3.32 queryErrorHandling()

```
bool Digikam::BdEngineBackend::queryErrorHandling (
    DbEngineSqlQuery & query,
    int retries )
```

Called with a failed query. Handles certain known errors and debug output. If it returns true, reexecute the query; if it returns false, return it as failed. Pass the number of retries already done for this query to help with some decisions.

### 6.123.3.33 readToList()

```
QList< QVariant > Digikam::BdEngineBackend::readToList (
    DbEngineSqlQuery & query )
```

Reads data of returned result set into a list which is returned. The read process is column wise, which means all data elements of a row is read, then the resultset is switched to the next row.

### 6.123.3.34 rollbackTransaction()

```
void Digikam::BdEngineBackend::rollbackTransaction ( )
```

Rollback the current database transaction

### 6.123.3.35 setDbEngineErrorHandler()

```
void Digikam::BdEngineBackend::setDbEngineErrorHandler (
    DbEngineErrorHandler *const handler )
```

Add a [DbEngineErrorHandler](#). This object must be created in the main thread. If a database error occurs, this object can handle problem solving and user interaction.

### 6.123.3.36 setForeignKeyChecks()

```
void Digikam::BdEngineBackend::setForeignKeyChecks (
    bool check )
```

Enables or disables FOREIGN\_KEY\_CHECKS for the database. This function depends on the database engine.



### 6.123.3.37 status()

```
BdEngineBackend::Status Digikam::BdEngineBackend::status ( ) const
```

Returns the current status of the database backend

### 6.123.3.38 tables()

```
QStringList Digikam::BdEngineBackend::tables ( )
```

Returns a list with the names of tables in the database.

## 6.124 Digikam::BdEngineBackend::QueryState Class Reference

### Public Member Functions

- **operator bool** ( ) const
- **operator QueryStateEnum** ( ) const
- **QueryState** (const [QueryStateEnum](#) value)

## 6.125 Digikam::BdEngineBackendPrivate Class Reference

Inheritance diagram for Digikam::BdEngineBackendPrivate:



### Classes

- class [AbstractUnlocker](#)
- class [AbstractWaitingUnlocker](#)
- class [BusyWaiter](#)
- class [ErrorLocker](#)

## Public Member Functions

- **BdEngineBackendPrivate** ([BdEngineBackend](#) \*const backend)
- bool **checkOperationStatus** ()
- bool **checkRetrySQLiteLockError** (int retries)
- void **closeDatabaseForThread** ()
- void **connectionErrorAbortQueries** () override
- void **connectionErrorContinueQueries** () override
- QString **connectionName** ()
- QSqlDatabase **createDatabaseConnection** ()
- QSqlError **databaseErrorForThread** ()
- QSqlDatabase **databaseForThread** ()
- void **debugOutputFailedQuery** (const QSqlQuery &query) const
- void **debugOutputFailedTransaction** (const QSqlError &error) const
- bool **decrementTransactionCount** ()
- bool **handleWithErrorHandler** (const [DbEngineSqlQuery](#) \*const query)
- bool **incrementTransactionCount** ()
- void **init** (const QString &connectionName, [DbEngineLocking](#) \*const locking)
- bool **isConnectionError** (const [DbEngineSqlQuery](#) &query) const
- bool **isInMainThread** () const
- bool **isInUIThread** () const
- bool **isSQLiteLockError** (const [DbEngineSqlQuery](#) &query) const
- bool **isSQLiteLockTransactionError** (const QSqlError &lastError) const
- bool **needToConsultUserForError** (const [DbEngineSqlQuery](#) &query) const
- bool **needToHandleWithErrorHandler** (const [DbEngineSqlQuery](#) &query) const
- void **queryOperationWakeAll** ([BdEngineBackend::QueryOperationStatus](#) status)
- bool **reconnectOnError** () const
- bool **resetDatabaseForThread** ()
- void **setDatabaseErrorForThread** (const QSqlError &lastError)
- void **setQueryOperationFlag** ([BdEngineBackend::QueryOperationStatus](#) status)
- virtual void **transactionFinished** ()

## Public Attributes

- QString **backendName**
- QWaitCondition **busyWaitCondVar**
- QMutex **busyWaitMutex**
- int **currentValidity** = 0
- [DbEngineErrorHandler](#) \* **errorHandler** = nullptr
- QWaitCondition **errorLockCondVar**
- QMutex **errorLockMutex**
- [BdEngineBackend::QueryOperationStatus](#) **errorLockOperationStatus** = [BdEngineBackend::ExecuteNormal](#)
- bool **isInTransaction** = false
- [DbEngineLocking](#) \* **lock** = nullptr
- [BdEngineBackend::QueryOperationStatus](#) **operationStatus** = [BdEngineBackend::ExecuteNormal](#)
- [DbEngineParameters](#) **parameters**
- [BdEngineBackend](#) \*const **q** = nullptr
- [BdEngineBackend::Status](#) **status** = [BdEngineBackend::Unavailable](#)
- [QThreadStorage](#)< [DbEngineThreadData](#) \* > **threadDataStorage**

## Friends

- class **AbstractUnlocker**

## 6.125.1 Member Function Documentation

### 6.125.1.1 connectionErrorAbortQueries()

```
void Digikam::BdEngineBackendPrivate::connectionErrorAbortQueries ( ) [override], [virtual]
```

Implements [Digikam::DbEngineErrorAnswer](#).

### 6.125.1.2 connectionErrorContinueQueries()

```
void Digikam::BdEngineBackendPrivate::connectionErrorContinueQueries ( ) [override], [virtual]
```

called by [DbEngineErrorHandler](#), implementing [DbEngineErrorAnswer](#).

Implements [Digikam::DbEngineErrorAnswer](#).

### 6.125.1.3 databaseForThread()

```
QSqlDatabase Digikam::BdEngineBackendPrivate::databaseForThread ( )
```

"A connection can only be used from within the thread that created it. Moving connections between threads or creating queries from a different thread is not supported." => one QSqlDatabase object per thread. The main class' open/close methods only interact with the "firstDatabase" object. When another thread requests a DB, a new connection is opened and closed at finishing of the thread.

### 6.125.1.4 handleWithErrorHandler()

```
bool Digikam::BdEngineBackendPrivate::handleWithErrorHandler (
    const DbEngineSqlQuery *const query )
```

Returns true if the query shall be retried.

### 6.125.1.5 queryOperationWakeAll()

```
void Digikam::BdEngineBackendPrivate::queryOperationWakeAll (
    BdEngineBackend::QueryOperationStatus status )
```

Set the wait flag to queryStatus and wake all waiting threads. Typically, call wakeAll with status ExecuteNormal or AbortQueries.

### 6.125.1.6 setQueryOperationFlag()

```
void Digikam::BdEngineBackendPrivate::setQueryOperationFlag (
    BdEngineBackend::QueryOperationStatus status )
```

Set the wait flag to queryStatus. Typically, call this with Wait.

## 6.125.2 Member Data Documentation

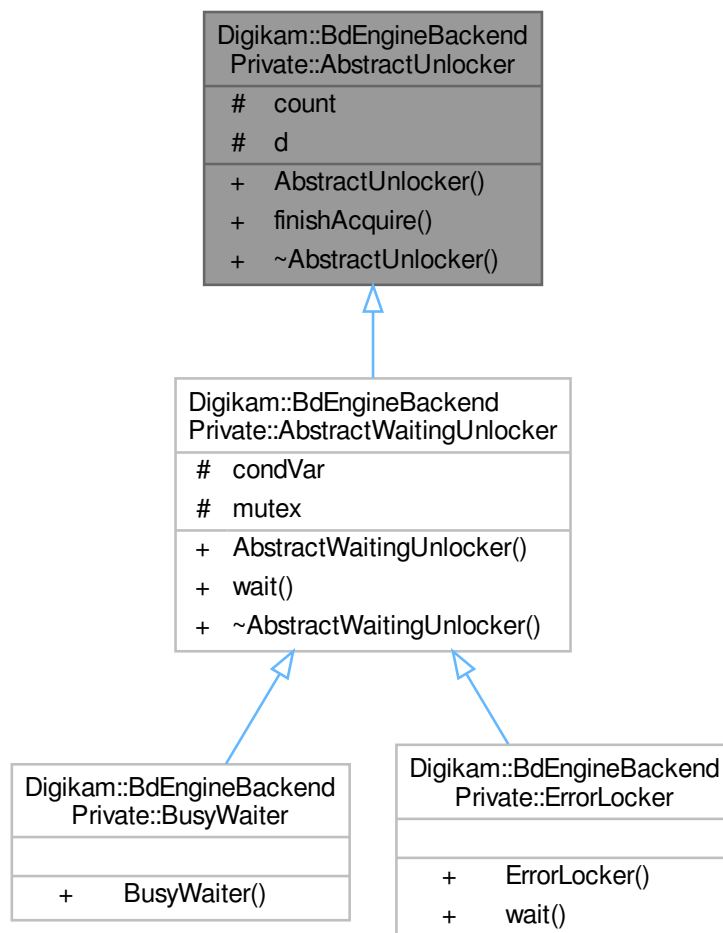
### 6.125.2.1 currentValidity

```
int Digikam::BdEngineBackendPrivate::currentValidity = 0
```

This compares to [DbEngineThreadData](#)'s valid. If currentValidity is increased and > valid, the db is marked as invalid

## 6.126 Digikam::BdEngineBackendPrivate::AbstractUnlocker Class Reference

Inheritance diagram for Digikam::BdEngineBackendPrivate::AbstractUnlocker:



### Public Member Functions

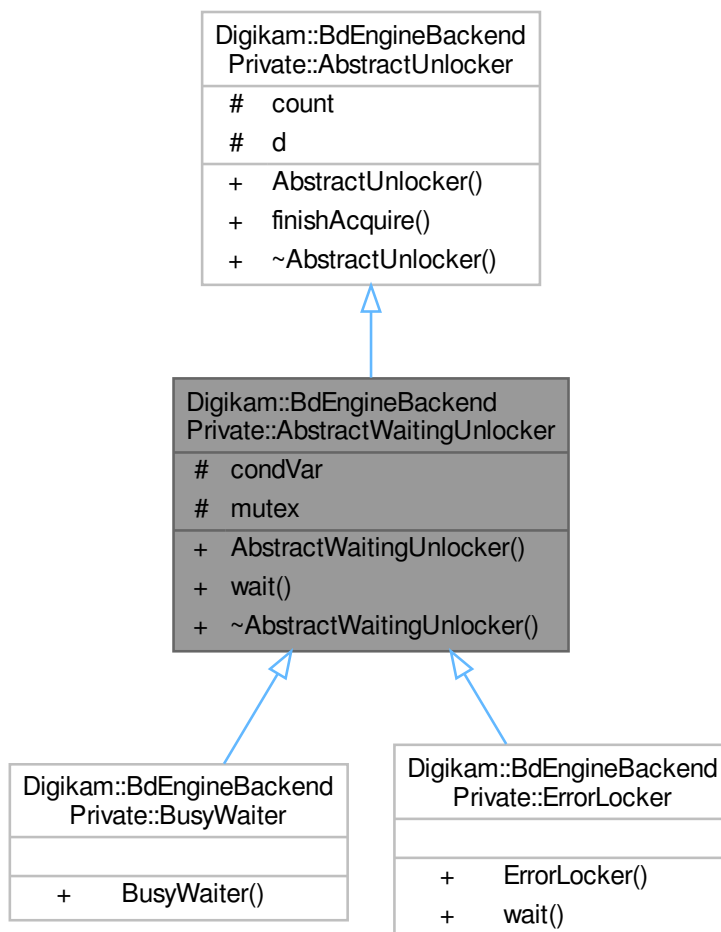
- **AbstractUnlocker** ([BdEngineBackendPrivate](#) \*const dd)
- void **finishAcquire** ()

### Protected Attributes

- int **count** = 0
- [BdEngineBackendPrivate](#) \*const **d** = nullptr

## 6.127 Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker Class Reference

Inheritance diagram for Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker:



### Public Member Functions

- **AbstractWaitingUnlocker** ([BdEngineBackendPrivate](#) \*const d, QMutex \*const mutex, QWaitCondition \*const condVar)
- bool **wait** (unsigned long time=ULONG\_MAX)

### Public Member Functions inherited from Digikam::BdEngineBackendPrivate::AbstractUnlocker

- **AbstractUnlocker** (BdEngineBackendPrivate \*const dd)
- void **finishAcquire** ()

### Protected Attributes

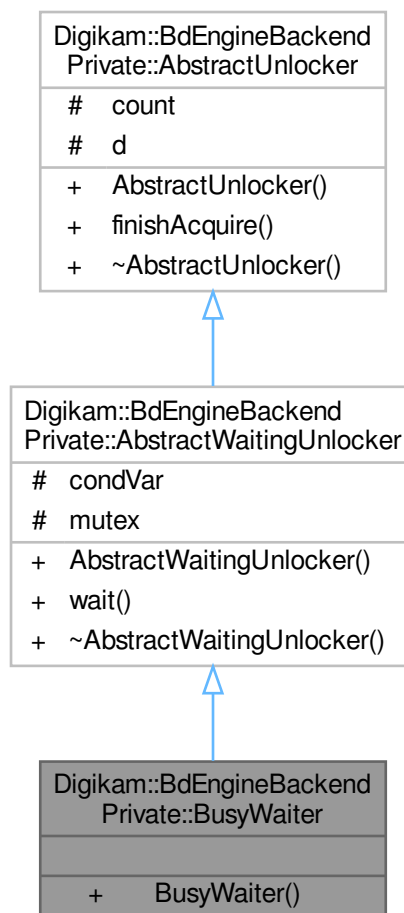
- QWaitCondition \*const **condVar** = nullptr
- QMutex \*const **mutex** = nullptr

### Protected Attributes inherited from Digikam::BdEngineBackendPrivate::AbstractUnlocker

- int **count** = 0
- BdEngineBackendPrivate \*const **d** = nullptr

## 6.128 Digikam::BdEngineBackendPrivate::BusyWaiter Class Reference

Inheritance diagram for Digikam::BdEngineBackendPrivate::BusyWaiter:



## Public Member Functions

- **BusyWaiter** ([BdEngineBackendPrivate](#) \*const d)

## Public Member Functions inherited from

### [Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker](#)

- **AbstractWaitingUnlocker** ([BdEngineBackendPrivate](#) \*const d, QMutex \*const mutex, QWaitCondition \*const condVar)
- bool **wait** (unsigned long time=ULONG\_MAX)

## Public Member Functions inherited from

### [Digikam::BdEngineBackendPrivate::AbstractUnlocker](#)

- **AbstractUnlocker** ([BdEngineBackendPrivate](#) \*const dd)
- void **finishAcquire** ()

## Additional Inherited Members

## Protected Attributes inherited from

### [Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker](#)

- QWaitCondition \*const **condVar** = nullptr
- QMutex \*const **mutex** = nullptr

## Protected Attributes inherited from

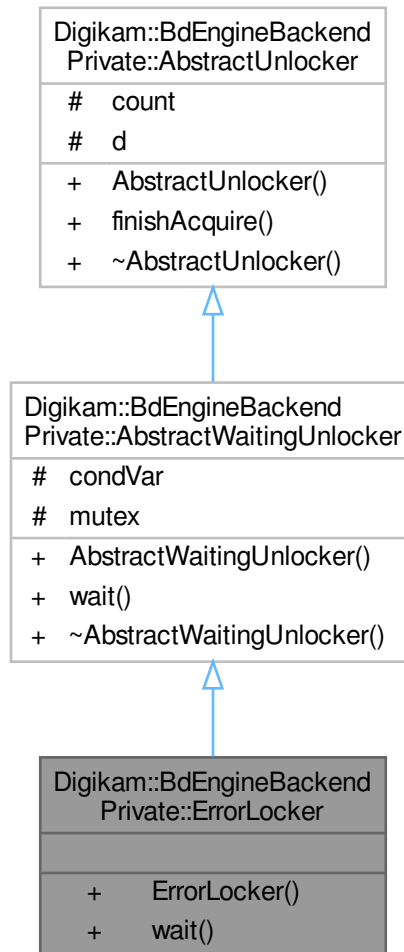
### [Digikam::BdEngineBackendPrivate::AbstractUnlocker](#)

- int **count** = 0
- [BdEngineBackendPrivate](#) \*const **d** = nullptr



## 6.129 Digikam::BdEngineBackendPrivate::ErrorLocker Class Reference

Inheritance diagram for Digikam::BdEngineBackendPrivate::ErrorLocker:



### Public Member Functions

- **ErrorLocker** ([BdEngineBackendPrivate](#) \*const d)
- void [wait](#) ()

### Public Member Functions inherited from [Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker](#)

- **AbstractWaitingUnlocker** ([BdEngineBackendPrivate](#) \*const d, `QMutex` \*const mutex, `QWaitCondition` \*const condVar)
- bool **wait** (unsigned long time=ULONG\_MAX)

### Public Member Functions inherited from [Digikam::BdEngineBackendPrivate::AbstractUnlocker](#)

- **AbstractUnlocker** ([BdEngineBackendPrivate](#) \*const dd)
- void **finishAcquire** ()

### Additional Inherited Members

### Protected Attributes inherited from [Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker](#)

- QWaitCondition \*const **condVar** = nullptr
- QMutex \*const **mutex** = nullptr

### Protected Attributes inherited from [Digikam::BdEngineBackendPrivate::AbstractUnlocker](#)

- int **count** = 0
- [BdEngineBackendPrivate](#) \*const **d** = nullptr

## 6.129.1 Member Function Documentation

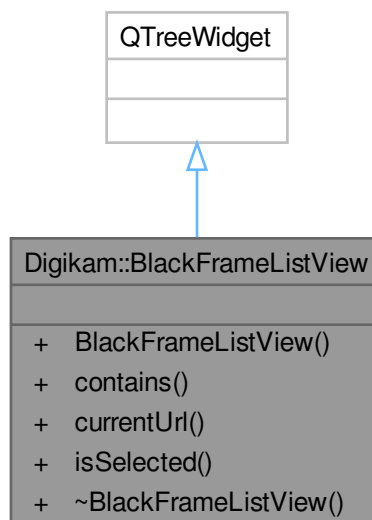
### 6.129.1.1 wait()

```
void Digikam::BdEngineBackendPrivate::ErrorLocker::wait ( )
```

This suspends the current thread if the query status as set by setFlag() is Wait and until the thread is woken with wakeAll(). The [CoreDbAccess](#) mutex will be unlocked while waiting.

## 6.130 Digikam::BlackFrameListView Class Reference

Inheritance diagram for Digikam::BlackFrameListView:



## Signals

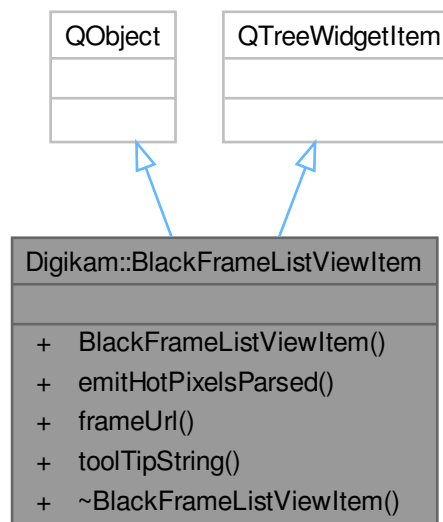
- void **signalBlackFrameRemoved** (const QUrl &)
- void **signalBlackFrameSelected** (const QList< [HotPixelProps](#) > &, const QUrl &)
- void **signalClearBlackFrameList** ()

## Public Member Functions

- **BlackFrameListView** (QWidget \*const parent=nullptr)
- bool **contains** (const QUrl &url)
- QUrl **currentUrl** ()
- bool **isSelected** (const QUrl &url)

## 6.131 Digikam::BlackFrameListViewItem Class Reference

Inheritance diagram for Digikam::BlackFrameListViewItem:



## Public Types

- enum **BlackFrameConst** { **PREVIEW** = 0 , **SIZE** = 1 , **HOTPIXELS** = 2 , **THUMB\_WIDTH** = 150 }

## Signals

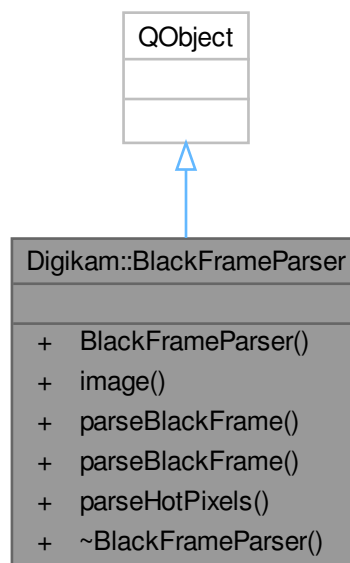
- void **signalHotPixelsParsed** (const QList< [HotPixelProps](#) > &, const QUrl &)

### Public Member Functions

- **BlackFrameListItem** (QWidget \*const parent, const QUrl &url)
- void **emitHotPixelsParsed** ()
- QUrl **frameUrl** () const
- QString **toolTipString** () const

## 6.132 Digikam::BlackFrameParser Class Reference

Inheritance diagram for Digikam::BlackFrameParser:



### Signals

- void **signalHotPixelsParsed** (const QList< [HotPixelProps](#) > &)
- void **signalLoadingComplete** ()
- void **signalLoadingProgress** (float)

### Public Member Functions

- **BlackFrameParser** (QObject \*const parent)
- [DImg](#) **image** () const
- void **parseBlackFrame** (const [DImg](#) &img)
- void **parseBlackFrame** (const QUrl &url)
- void **parseHotPixels** (const QString &file)

## 6.133 Digikam::BlackFrameToolTip Class Reference

Inheritance diagram for Digikam::BlackFrameToolTip:



### Public Member Functions

- **BlackFrameToolTip** (`QTreeWidgetItem *const view`)
- void **setItem** (`QTreeWidgetItem *const item`)
- void **setToolTipString** (`const QString &tip`)
- void **show** ()

## Public Member Functions inherited from [Digikam::DItemToolTip](#)

- **DItemToolTip** (QWidget \*const parent=nullptr)

## Protected Member Functions

- QRect [repositionRect](#) () override
- QString [tipContents](#) () override

## Protected Member Functions inherited from [Digikam::DItemToolTip](#)

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- void **resizeEvent** (QResizeEvent \*) override
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()

## 6.133.1 Member Function Documentation

### 6.133.1.1 [repositionRect\(\)](#)

QRect Digikam::BlackFrameToolTip::repositionRect ( ) [override], [protected], [virtual]

Implements [Digikam::DItemToolTip](#).

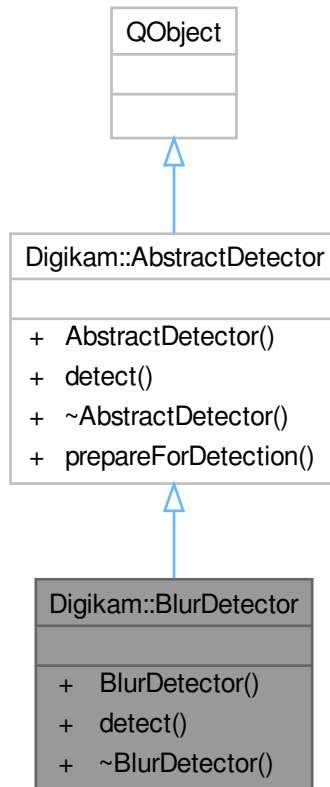
### 6.133.1.2 [tipContents\(\)](#)

QString Digikam::BlackFrameToolTip::tipContents ( ) [override], [protected], [virtual]

Implements [Digikam::DItemToolTip](#).

## 6.134 Digikam::BlurDetector Class Reference

Inheritance diagram for Digikam::BlurDetector:



### Public Member Functions

- `BlurDetector` (const [DImg](#) &image)
- float `detect` (const cv::Mat &image) const override

### Public Member Functions inherited from [Digikam::AbstractDetector](#)

- `AbstractDetector` (QObject \*const parent=nullptr)

### Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::AbstractDetector](#)

- static cv::Mat `prepareForDetection` (const [DImg](#) &inputImage)

## 6.134.1 Member Function Documentation

### 6.134.1.1 detect()

```
float Digikam::BlurDetector::detect (  
    const cv::Mat & image ) const [override], [virtual]
```

Implements [Digikam::AbstractDetector](#).



## 6.135 Digikam::BlurFilter Class Reference

Inheritance diagram for Digikam::BlurFilter:



### Public Member Functions

- **BlurFilter** (`DImg *const orgImage`, `QObject *const parent=nullptr`, `int radius=3`)
- **BlurFilter** (`DimgThreadedFilter *const parentFilter`, `const DImg &orgImage`, `const DImg &destImage`, `int progressBegin=0`, `int progressEnd=100`, `int radius=3`)

- **BlurFilter** (QObject \*const parent=nullptr)
- **FilterAction** filterAction () override
- QString filterIdentifier () const override
- void readParameters (const FilterAction &action) override

### Public Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void cancelFilter ()
- DImgThreadedFilter (DImg \*const orgImage, QObject \*const parent, const QString &name=QString())
- DImgThreadedFilter (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & filterName ()
- int filterVersion () const
- DImg getTargetImage ()
- QList< int > multithreadedSteps (int stop, int start=0) const
- virtual bool parametersSuccessfullyRead () const
- virtual QString readParametersError (const FilterAction &actionThatFailed) const
- void setFilterName (const QString &name)
- void setFilterVersion (int version)
- void setOriginalImage (const DImg &orgImage)
- void setupAndStartDirectly (const DImg &orgImage, DImgThreadedFilter \*const master, int progressBegin=0, int progressEnd=100)
- void setupFilter (const DImg &orgImage)
- virtual void startFilter ()
- virtual void startFilterDirectly ()
- virtual QList< int > supportedVersions () const

### Public Member Functions inherited from Digikam::DynamicThread

- DynamicThread (QObject \*const parent=nullptr)
- bool isFinished () const
- bool isRunning () const
- QThread::Priority priority () const
- void setEmitSignals (bool emitThem)
- void setPriority (QThread::Priority priority)
- State state () const
- ~DynamicThread () override

### Static Public Member Functions

- static int CurrentVersion ()
- static QString DisplayableName ()
- static QString FilterIdentifier ()
- static QList< int > SupportedVersions ()

### Additional Inherited Members

### Public Types inherited from Digikam::DynamicThread

- enum State { Inactive , Scheduled , Running , Deactivating }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.135.1 Constructor & Destructor Documentation

### 6.135.1.1 BlurFilter()

```
Digikam::BlurFilter::BlurFilter (
    DImgThreadedFilter *const parentFilter,
    const DImg & orgImage,
    const DImg & destImage,
    int progressBegin = 0,
    int progressEnd = 100,
    int radius = 3 ) [explicit]
```

Constructor for slave mode: execute immediately in current thread with specified master filter

## 6.135.2 Member Function Documentation

### 6.135.2.1 filterAction()

```
FilterAction Digikam::BlurFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.135.2.2 filterIdentifier()

```
QString Digikam::BlurFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

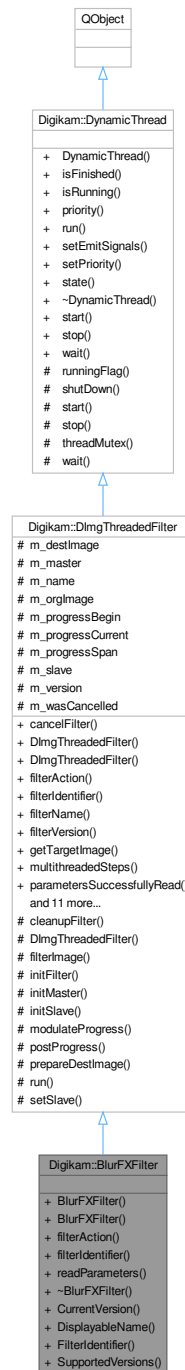
### 6.135.2.3 readParameters()

```
void Digikam::BlurFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.136 Digikam::BlurFXFilter Class Reference

Inheritance diagram for Digikam::BlurFXFilter:



### Public Types

- enum **BlurFXFilterTypes** {
  - ZoomBlur** = 0 , **RadialBlur** , **FarBlur** , **MotionBlur** ,
  - SoftenerBlur** , **ShakeBlur** , **FocusBlur** , **SmartBlur** ,
  - FrostGlass** , **Mosaic** }

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Member Functions

- **BlurFXFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, int blurFXType=ZoomBlur, int distance=100, int level=45)
- **BlurFXFilter** (QObject \*const parent=nullptr)
- [FilterAction](#) filterAction () override
- QString filterIdentifier () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > **multithreadedSteps** (int stop, int start=0) const
- virtual bool **parametersSuccessfullyRead** () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void **setFilterVersion** (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void **setupAndStartDirectly** (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void **setupFilter** (const [DImg](#) &orgImage)
- virtual void **startFilter** ()
- virtual void **startFilterDirectly** ()
- virtual QList< int > **supportedVersions** () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- **~DynamicThread** () override

## Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

**Additional Inherited Members****Public Slots inherited from [Digikam::DynamicThread](#)**

- void **start** ()
- void **stop** ()
- void **wait** ()

**Signals inherited from [Digikam::DImgThreadedFilter](#)**

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

**Signals inherited from [Digikam::DynamicThread](#)**

- void **finished** ()
- void **starting** ()

**Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)**

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

**Protected Member Functions inherited from [Digikam::DynamicThread](#)**

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

**Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)**

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0  
*To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.136.1 Member Function Documentation

### 6.136.1.1 filterAction()

```
FilterAction Digikam::BlurFXFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.136.1.2 filterIdentifier()

```
QString Digikam::BlurFXFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.136.1.3 readParameters()

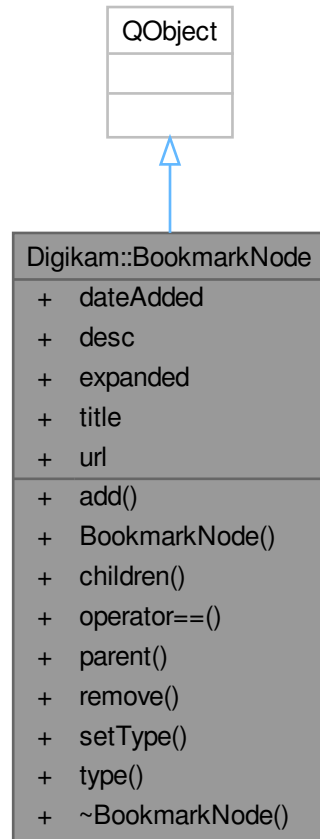
```
void Digikam::BlurFXFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).



## 6.137 Digikam::BookmarkNode Class Reference

Inheritance diagram for Digikam::BookmarkNode:



### Public Types

- enum **Type** {  
**Root** , **Folder** , **Bookmark** , **Separator** ,  
**RootFolder** }

### Public Member Functions

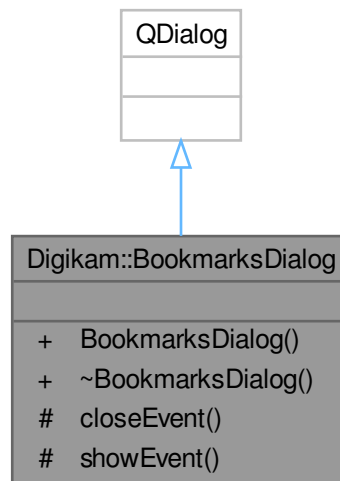
- void **add** ([BookmarkNode](#) \*const child, int offset=-1)
- [BookmarkNode](#) ([Type](#) type=Root, [BookmarkNode](#) \*const parent=nullptr)
- [QList](#)< [BookmarkNode](#) \* > **children** () const
- bool **operator==** (const [BookmarkNode](#) &other) const
- [BookmarkNode](#) \* **parent** () const
- void **remove** ([BookmarkNode](#) \*const child)
- void **setType** ([Type](#) type)
- [Type](#) **type** () const

### Public Attributes

- QDateTime **dateAdded**
- QString **desc**
- bool **expanded**
- QString **title**
- QString **url**

## 6.138 Digikam::BookmarksDialog Class Reference

Inheritance diagram for Digikam::BookmarksDialog:



### Public Member Functions

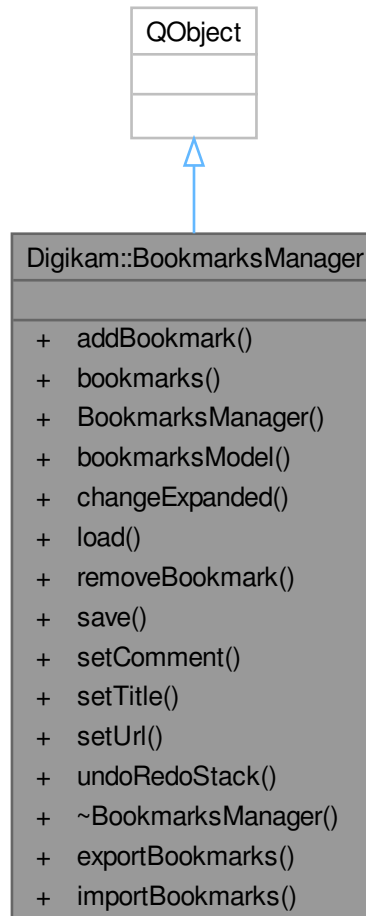
- **BookmarksDialog** (QWidget \*const parent=nullptr, [BookmarksManager](#) \*const mngr=nullptr)

### Protected Member Functions

- void **closeEvent** (QCloseEvent \*) override
- void **showEvent** (QShowEvent \*) override

## 6.139 Digikam::BookmarksManager Class Reference

Inheritance diagram for Digikam::BookmarksManager:



### Public Slots

- void `exportBookmarks` ()
- void `importBookmarks` ()

### Signals

- void `entryAdded` ([BookmarkNode](#) \*item)
- void `entryChanged` ([BookmarkNode](#) \*item)
- void `entryRemoved` ([BookmarkNode](#) \*parent, int row, [BookmarkNode](#) \*item)

## Public Member Functions

- void **addBookmark** ([BookmarkNode](#) \*const parent, [BookmarkNode](#) \*const node, int row=-1)
- [BookmarkNode](#) \* **bookmarks** ()
- **BookmarksManager** (const QString &bookmarksFile, QObject \*const parent=nullptr)
- [BookmarksModel](#) \* **bookmarksModel** ()
- void **changeExpanded** ()
- void **load** ()
- void **removeBookmark** ([BookmarkNode](#) \*const node)
- void **save** ()
- void **setComment** ([BookmarkNode](#) \*const node, const QString &newDesc)
- void **setTitle** ([BookmarkNode](#) \*const node, const QString &newTitle)
- void **setUrl** ([BookmarkNode](#) \*const node, const QString &newUrl)
- QUndoStack \* **undoRedoStack** () const

## Friends

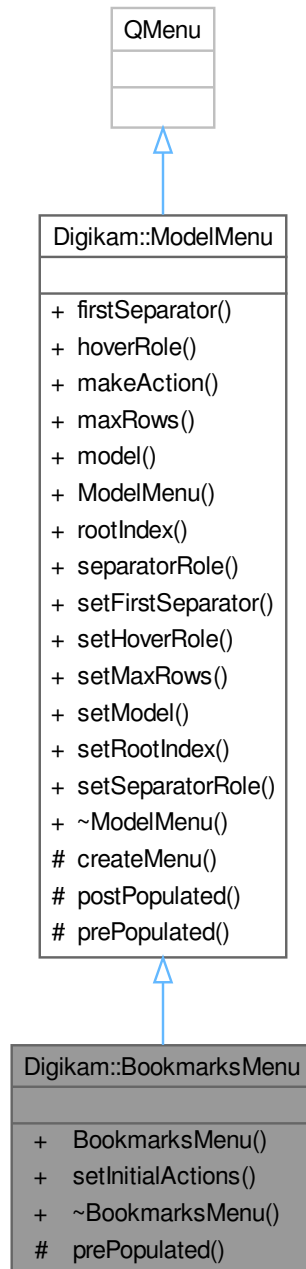
- class **ChangeBookmarkCommand**
- class **RemoveBookmarksCommand**

### 6.139.1 Detailed Description

Bookmark manager, owner of the bookmarks, loads, saves and basic tasks

## 6.140 Digikam::BookmarksMenu Class Reference

Inheritance diagram for Digikam::BookmarksMenu:



### Signals

- void **openUrl** (const QUrl &url)

## Signals inherited from [Digikam::ModelMenu](#)

- void **activated** (const QModelIndex &index)
- void **hovered** (const QString &text)

## Public Member Functions

- **BookmarksMenu** ([BookmarksManager](#) \*const mngr, QWidget \*const parent=nullptr)
- void **setInitialActions** (const QList< QAction \* > &actions)

## Public Member Functions inherited from [Digikam::ModelMenu](#)

- int **firstSeparator** () const
- int **hoverRole** () const
- QAction \* **makeAction** (const QIcon &icon, const QString &text, QObject \*const parent)
- int **maxRows** () const
- QAbstractItemModel \* **model** () const
- **ModelMenu** (QWidget \*const parent=nullptr)
- QModelIndex **rootIndex** () const
- int **separatorRole** () const
- void **setFirstSeparator** (int offset)
- void **setHoverRole** (int role)
- void **setMaxRows** (int max)
- void **setModel** (QAbstractItemModel \*model)
- void **setRootIndex** (const QModelIndex &index)
- void **setSeparatorRole** (int role)

## Protected Member Functions

- bool **prePopulated** () override  
*add any actions before the tree, return true if any actions are added.*

## Protected Member Functions inherited from [Digikam::ModelMenu](#)

- void **createMenu** (const QModelIndex &parent, int max, QMenu \*parentMenu=nullptr, QMenu \*menu=nullptr)  
*put all of the children of parent into menu up to max*
- virtual void **postPopulated** ()  
*add any actions after the tree*

### 6.140.1 Detailed Description

Menu that is dynamically populated from the bookmarks

### 6.140.2 Member Function Documentation

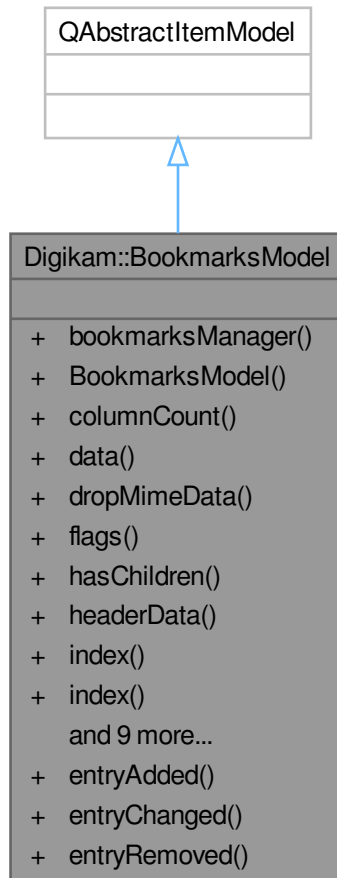
#### 6.140.2.1 prePopulated()

```
bool Digikam::BookmarksMenu::prePopulated ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ModelMenu](#).

## 6.141 Digikam::BookmarksModel Class Reference

Inheritance diagram for Digikam::BookmarksModel:



### Public Types

- enum **Roles** {  
**TypeRole** = Qt::UserRole + 1 , **UrlRole** = Qt::UserRole + 2 , **UrlStringRole** = Qt::UserRole + 3 , **Separator**↔  
**Role** = Qt::UserRole + 4 ,  
**DateAddedRole** = Qt::UserRole + 5 }

### Public Slots

- void **entryAdded** ([BookmarkNode](#) \*item)
- void **entryChanged** ([BookmarkNode](#) \*item)
- void **entryRemoved** ([BookmarkNode](#) \*parent, int row, [BookmarkNode](#) \*item)

## Public Member Functions

- [BookmarksManager](#) \* **bookmarksManager** () const
- **BookmarksModel** ([BookmarksManager](#) \*const mngr, QObject \*const parent=nullptr)
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- bool **dropMimeData** (const QMimeData \*data, Qt::DropAction action, int row, int column, const QModelIndex &parent) override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **hasChildren** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- QModelIndex **index** ([BookmarkNode](#) \*node) const
- QModelIndex **index** (int, int, const QModelIndex &=QModelIndex()) const override
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const override
- QStringList **mimeTypes** () const override
- [BookmarkNode](#) \* **node** (const QModelIndex &index) const
- QModelIndex **parent** (const QModelIndex &index=QModelIndex()) const override
- bool **removeRows** (int row, int count, const QModelIndex &parent=QModelIndex()) override
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override
- Qt::DropActions **supportedDropActions** () const override

### 6.141.1 Detailed Description

[BookmarksModel](#) is a QAbstractItemModel wrapper around the [BookmarkManager](#)

## 6.142 Digikam::BorderContainer Class Reference

### Public Types

- enum **BorderTypes** {  
**SolidBorder** = 0 , **NiepceBorder** , **BeveledBorder** , **PineBorder** ,  
**WoodBorder** , **PaperBorder** , **ParqueBorder** , **IceBorder** ,  
**LeafBorder** , **MarbleBorder** , **RainBorder** , **CratersBorder** ,  
**DriedBorder** , **PinkBorder** , **StoneBorder** , **ChalkBorder** ,  
**GraniteBorder** , **RockBorder** , **WallBorder** }

### Static Public Member Functions

- static QString **getBorderPath** (int border)



### Public Attributes

- QColor **bevelLowerRightColor** = QColor(128, 128, 128)
- QColor **bevelUpperLeftColor** = QColor(192, 192, 192)
- QString **borderPath**
- double **borderPercent** = 0.1
- int **borderType** = 0
- int **borderWidth1** = 0
- int **borderWidth2** = 0
- int **borderWidth3** = 0
- int **borderWidth4** = 0
- QColor **decorativeFirstColor** = QColor(0, 0, 0)
- QColor **decorativeSecondColor** = QColor(0, 0, 0)
- QColor **niepceBorderColor** = QColor(255, 255, 255)
- QColor **niepceLineColor** = QColor(0, 0, 0)
- int **orgHeight** = 0
- int **orgWidth** = 0
- bool **preserveAspectRatio** = true
- QColor **solidColor** = QColor(0, 0, 0)

## 6.143 Digikam::BorderFilter Class Reference

Inheritance diagram for Digikam::BorderFilter:



### Public Member Functions

- **BorderFilter** (*DImg* \*orgImage, *QObject* \*const parent=nullptr, const *BorderContainer* &settings=*BorderContainer*())
- **BorderFilter** (*QObject* \*const parent=nullptr)

- [FilterAction filterAction](#) () override
- [QString filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.143.1 Constructor & Destructor Documentation

### 6.143.1.1 BorderFilter()

```
Digikam::BorderFilter::BorderFilter (
    QObject *const parent = nullptr ) [explicit]
```

Constructor using settings to preserve aspect ratio of image.

## 6.143.2 Member Function Documentation

### 6.143.2.1 filterAction()

```
FilterAction Digikam::BorderFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.143.2.2 filterIdentifier()

```
QString Digikam::BorderFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

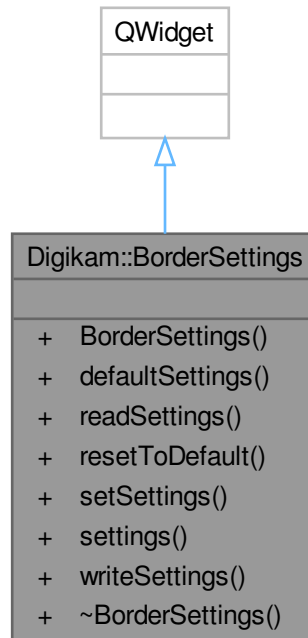
### 6.143.2.3 readParameters()

```
void Digikam::BorderFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.144 Digikam::BorderSettings Class Reference

Inheritance diagram for Digikam::BorderSettings:



### Signals

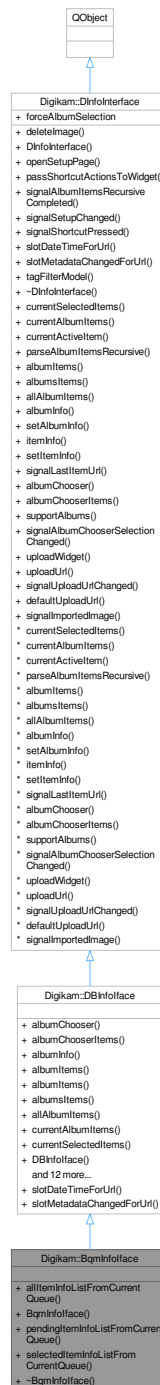
- void `signalSettingsChanged` ()

### Public Member Functions

- `BorderSettings` (`QWidget *const parent`)
- `BorderContainer defaultSettings` () const
- void `readSettings` (`const KConfigGroup &group`)
- void `resetToDefault` ()
- void `setSettings` (`const BorderContainer &settings`)
- `BorderContainer settings` () const
- void `writeSettings` (`KConfigGroup &group`)

## 6.145 Digikam::BqmlInfoface Class Reference

Inheritance diagram for Digikam::BqmlInfoface:



### Public Member Functions

- [QueuePoolItemsList allItemInfoListFromCurrentQueue \(\)](#) const
- **BqmlInfoface** (QObject \*const parent)
- [QueuePoolItemsList pendingItemInfoListFromCurrentQueue \(\)](#) const
- [QueuePoolItemsList selectedItemInfoListFromCurrentQueue \(\)](#) const

## Public Member Functions inherited from [Digikam::DBInfoface](#)

- `QWidget * albumChooser` (`QWidget *const parent`) const override  
*Albums chooser view methods (to use items from albums before to process).*
- `DAlbumIDs albumChooserItems` () const override
- `DInfoMap albumInfo` (int) const override
- `QList< QUrl > albumItems` (`Album *const album`) const
- `QList< QUrl > albumItems` (int id) const override
- `QList< QUrl > albumsItems` (const `DAlbumIDs &`) const override
- `QList< QUrl > allAlbumItems` () const override
- `QList< QUrl > currentAlbumItems` () const override
- `QList< QUrl > currentSelectedItems` () const override  
*Low level items and albums methods.*
- `DBInfoface` (`QObject *const parent`, const `QList< QUrl > &lst=QList< QUrl >()`, const `OperationType type=UnspecifiedOps`)
- `QUrl defaultUploadUrl` () const override  
*Url to upload new items without to use album selector.*
- void `deleteImage` (const `QUrl &url`) override  
*Manipulate with item.*
- `DInfoMap itemInfo` (const `QUrl &`) const override
- void `openSetupPage` (`SetupPage page`) override  
*Open configuration dialog page.*
- void `parseAlbumItemsRecursive` () override
- `QMap< QString, QString > passShortcutActionsToWidget` (`QWidget *const wdg`) const override  
*Pass extra shortcut actions to widget and return prefixes of shortcuts.*
- void `setItemInfo` (const `QUrl &`, const `DInfoMap &`) override
- bool `supportAlbums` () const override
- `QAbstractItemModel * tagFilterModel` () override  
*Return an instance of tag filter model if host application support this feature, else null pointer.*
- `QUrl uploadUrl` () const override
- `QWidget * uploadWidget` (`QWidget *const parent`) const override  
*Album selector view methods (to upload items from an external place).*

## Public Member Functions inherited from [Digikam::DInfoInterface](#)

- `DInfoInterface` (`QObject *const parent`)
- `Q_SIGNAL` void `signalAlbumItemsRecursiveCompleted` (const `QList< QUrl > &imageList`)
- `Q_SIGNAL` void `signalSetupChanged` ()
- `Q_SIGNAL` void `signalShortcutPressed` (const `QString &shortcut`, int val)
- virtual `Q_SLOT` void `slotDateTimeForUrl` (const `QUrl &url`, const `QDateTime &dt`, bool updModDate)  
*Slot to call when date time stamp from item is changed.*
- virtual `Q_SLOT` void `slotMetadataChangedForUrl` (const `QUrl &url`)  
*Slot to call when something in metadata from item is changed.*
  
- virtual `QUrl currentActiveItem` () const
- virtual void `setAlbumInfo` (int, const `DInfoMap &`) const
- `Q_SIGNAL` void `signalLastItemUrl` (const `QUrl &`)
  
- `Q_SIGNAL` void `signalAlbumChooserSelectionChanged` ()
  
- `Q_SIGNAL` void `signalUploadUrlChanged` ()
- `Q_SIGNAL` void `signalImportedImage` (const `QUrl &`)



## Additional Inherited Members

### Public Types inherited from [Digikam::DInfoInterface](#)

- typedef QList< int > **DAAlbumIDs**  
*List of [Album](#) ids.*
- typedef QMap< QString, QVariant > **DInfoMap**  
*Map of properties name and value.*
- enum **SetupPage** { **ExifToolPage** = 0 , **ImageQualityPage** }

### Public Slots inherited from [Digikam::DBInfoIface](#)

- void **slotDateTimeForUrl** (const QUrl &url, const QDateTime &dt, bool updModDate) override
- void **slotMetadataChangedForUrl** (const QUrl &url) override

### Public Attributes inherited from [Digikam::DInfoInterface](#)

- bool **forceAlbumSelection** = false

## 6.145.1 Member Function Documentation

### 6.145.1.1 allItemInfoListFromCurrentQueue()

`QueuePoolItemsList` Digikam::BqmInfoIface::allItemInfoListFromCurrentQueue ( ) const

Return all item info list from the current queue.

### 6.145.1.2 pendingItemInfoListFromCurrentQueue()

`QueuePoolItemsList` Digikam::BqmInfoIface::pendingItemInfoListFromCurrentQueue ( ) const

Return pending item info list from the current queue.

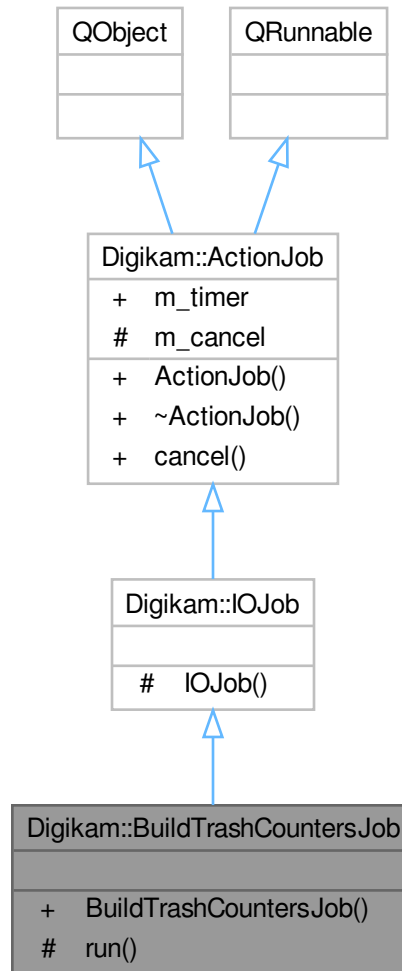
### 6.145.1.3 selectedItemInfoListFromCurrentQueue()

`QueuePoolItemsList` Digikam::BqmInfoIface::selectedItemInfoListFromCurrentQueue ( ) const

Return selected item info list from the current queue.

## 6.146 Digikam::BuildTrashCountersJob Class Reference

Inheritance diagram for Digikam::BuildTrashCountersJob:



### Signals

- void **signalTrashCountersMap** (const QMap< QString, int > &counterMap)

### Signals inherited from [Digikam::IOJob](#)

- void **signalError** (const QString &errMsg)
- void **signalOneProcessed** (const QUrl &url)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

**Protected Member Functions**

- void `run ()` override

**Additional Inherited Members****Public Slots inherited from [Digikam::ActionJob](#)**

- void `cancel ()`

**Public Member Functions inherited from [Digikam::ActionJob](#)**

- `ActionJob` (QObject \*const parent=nullptr)
- `~ActionJob ()` override

**Public Attributes inherited from [Digikam::ActionJob](#)**

- QElapsedTimer `m_timer`

**Protected Attributes inherited from [Digikam::ActionJob](#)**

- bool `m_cancel` = false

## 6.147 Digikam::BWSepiaContainer Class Reference

**Public Types**

- enum `BlackWhiteConversionType` {  
`BWNoFilter` = 0 , `BWGreenFilter` , `BWOrangeFilter` , `BWRedFilter` ,  
`BWYellowFilter` , `BWYellowGreenFilter` , `BWBlueFilter` , `BWGeneric` ,  
`BWAgfa200X` , `BWAgfapan25` , `BWAgfapan100` , `BWAgfapan400` ,  
`BWIlfordDelta100` , `BWIlfordDelta400` , `BWIlfordDelta400Pro3200` , `BWIlfordFP4` ,  
`BWIlfordHP5` , `BWIlfordPanF` , `BWIlfordXP2Super` , `BWKodakTmax100` ,  
`BWKodakTmax400` , `BWKodakTriX` , `BWIlfordSFX200` , `BWIlfordSFX400` ,  
`BWIlfordSFX800` , `BWNoTone` , `BWSepiaTone` , `BWBrownTone` ,  
`BWColdTone` , `BWSeleniumTone` , `BWPlatinumTone` , `BWGreenTone` ,  
`BWKodakHIE` }

**Public Member Functions**

- `BWSepiaContainer` (int ptype)
- `BWSepiaContainer` (int ptype, const [CurvesContainer](#) &container)

## Public Attributes

- [BCGContainer](#) **bcgPrm**
- [CurvesContainer](#) **curvesPrm**
- int **filmType** = [BWGeneric](#)
- int **filterType** = [BWNoFilter](#)
- bool **preview** = false
- int **previewType** = [BWGeneric](#)
- double **strength** = 1.0
- int **toneType** = [BWNoTone](#)

## 6.147.1 Member Enumeration Documentation

### 6.147.1.1 BlackWhiteConversionType

enum [Digikam::BWSepiaContainer::BlackWhiteConversionType](#)

#### Enumerator

BWNoFilter	B&W filter to the front of lens.
BWGeneric	B&W film simulation.
BWIlfordSFX200	Infrared film simulation.
BWNoTone	Chemical color tone filter.
BWKodakHIE	Infrared film simulation.

## 6.148 Digikam::BWSepiaFilter Class Reference

Inheritance diagram for Digikam::BWSepiaFilter:



### Public Member Functions

- **BWSepiaFilter** (`DImg *orgImage`, `QObject *const parent=nullptr`, `const BWSepiaContainer &settings=BWSepiaContainer()`)
- **BWSepiaFilter** (`QObject *const parent=nullptr`)

- [FilterAction filterAction](#) () override
- [QString filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.148.1 Member Function Documentation

### 6.148.1.1 filterAction()

```
FilterAction Digikam::BWSepiaFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.148.1.2 filterIdentifier()

```
QString Digikam::BWSepiaFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.148.1.3 readParameters()

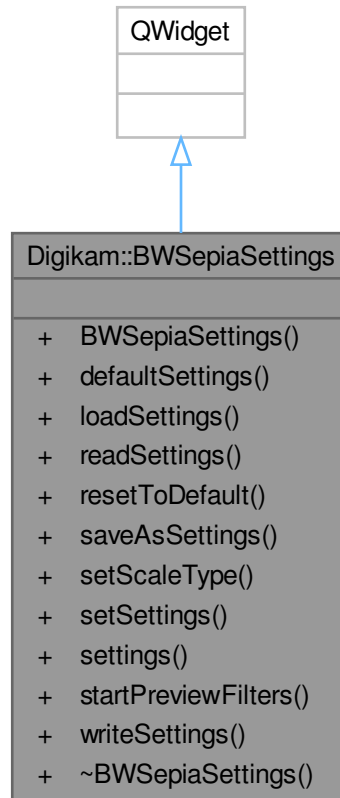
```
void Digikam::BWSepiaFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).



## 6.149 Digikam::BWSepiaSettings Class Reference

Inheritance diagram for Digikam::BWSepiaSettings:



### Signals

- void `signalSettingsChanged ()`

### Public Member Functions

- `BWSepiaSettings` (`QWidget *const parent`, `DImg *const img`)
- `BWSepiaContainer defaultSettings ()` const
- void `loadSettings ()`
- void `readSettings` (`KConfigGroup &group`)
- void `resetToDefault ()`
- void `saveAsSettings ()`
- void `setScaleType` (`HistogramScale scale`)
- void `setSettings` (const `BWSepiaContainer &settings`)
- `BWSepiaContainer settings ()` const
- void `startPreviewFilters ()`
- void `writeSettings` (`KConfigGroup &group`)

## 6.150 Digikam::CachedPixmapKey Class Reference

### Public Attributes

- QPixmapCache::Key **key**
- QRect **region**

## 6.151 Digikam::CachedPixmaps Class Reference

### Public Member Functions

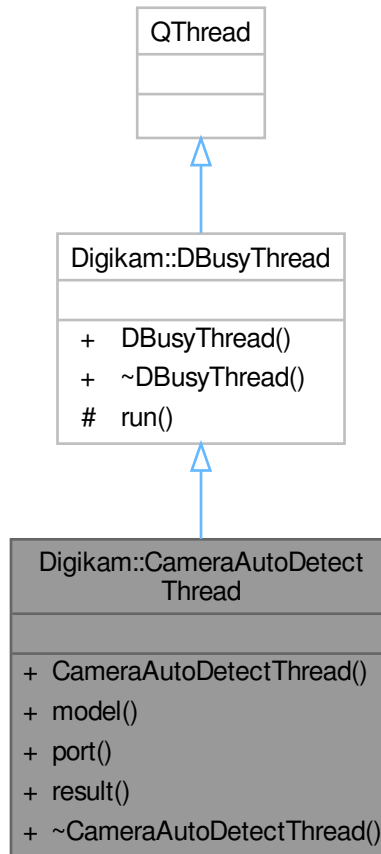
- **CachedPixmaps** (int maxCount=2)
- void **clear** ()
- bool **find** (const QRect &region, QPixmap \*const pix, QRect \*const source)
- void **insert** (const QRect &region, const QPixmap &pixmap)
- void **setMaxCount** (int)

### Protected Attributes

- QQueue< [CachedPixmapKey](#) > **keys**
- int **maxCount** = 2

## 6.152 Digikam::CameraAutoDetectThread Class Reference

Inheritance diagram for Digikam::CameraAutoDetectThread:



### Public Member Functions

- `CameraAutoDetectThread` (`QObject *const parent`)
- `QString model () const`
- `QString port () const`
- `int result () const`

### Public Member Functions inherited from [Digikam::DBusThread](#)

- `DBusThread` (`QObject *const parent`)

### Additional Inherited Members

### Signals inherited from [Digikam::DBusThread](#)

- `void signalComplete ()`

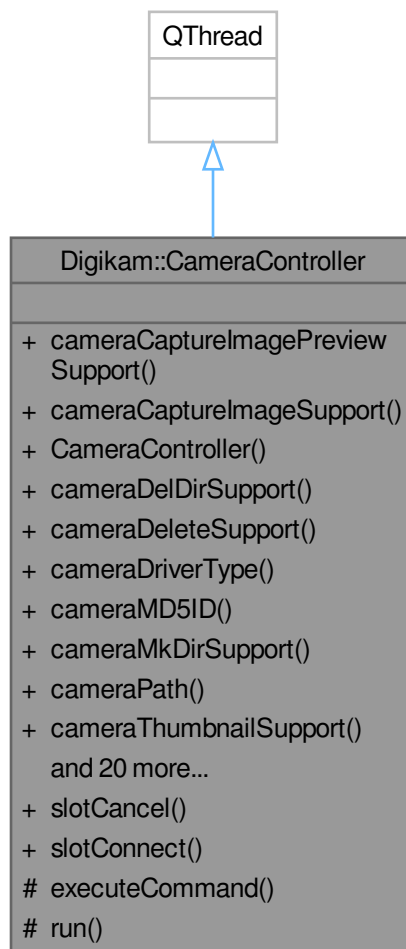
## Protected Member Functions inherited from [Digikam::DBusyThread](#)

- void **run** () override

*Reimplement this method with your code to run in a separate thread.*

## 6.153 Digikam::CameraController Class Reference

Inheritance diagram for Digikam::CameraController:



### Public Slots

- void **slotCancel** ()
- void **slotConnect** ()

## Signals

- void **signalBusy** (bool val)
- void **signalCameraInformation** (const QString &summary, const QString &>manual, const QString &about)
- void **signalConnected** (bool val)
- void **signalDeleted** (const QString &folder, const QString &file, bool status)
- void **signalDownloaded** (const QString &folder, const QString &file, const QString &temp, int status)
- void **signalFileList** (const CamItemInfoList &infoList)
- void **signalFolderList** (const QStringList &folderList)
- void **signalFreeSpace** (qint64 bytesSize, qint64 bytesAvail)
- void **signalInternalDeleteFailed** (const QString &folder, const QString &file)
- void **signalInternalDownloadFailed** (const QString &folder, const QString &file)
- void **signalInternalLockFailed** (const QString &folder, const QString &file)
- void **signalInternalUploadFailed** (const QString &folder, const QString &file, const QString &src)
- void **signalLocked** (const QString &folder, const QString &file, bool status)
- void **signalLogMsg** (const QString &msg, DHistoryView::EntryType type, const QString &folder, const QString &file)
- void **signalMetadata** (const QString &folder, const QString &file, const [MetaEngineData](#) &exifData)
- void **signalPreview** (const QImage &preview)
- void **signalThumbInfo** (const QString &folder, const QString &file, const [CamItemInfo](#) &itemInfo, const QImage &thumb)
- void **signalThumbInfoFailed** (const QString &folder, const QString &file, const [CamItemInfo](#) &itemInfo)
- void **signalUploaded** (const [CamItemInfo](#) &itemInfo)

## Public Member Functions

- bool **cameraCaptureImagePreviewSupport** () const
- bool **cameraCaptureImageSupport** () const
- **CameraController** (QWidget \*const parent, const QString &title, const QString &model, const QString &port, const QString &path)
- bool **cameraDelDirSupport** () const
- bool **cameraDeleteSupport** () const
- DKCamera::CameraDriverType **cameraDriverType** () const
- QByteArray **cameraMD5ID** () const
- bool **cameraMkDirSupport** () const
- QString **cameraPath** () const
- bool **cameraThumbnailSupport** () const
- QString **cameraTitle** () const
- bool **cameraUploadSupport** () const
- void **capture** ()
- void **deleteFile** (const QString &folder, const QString &file)
- void **download** (const [DownloadSettings](#) &downloadSettings)
- void **download** (const DownloadSettingsList &list)
- void **getCameraInformation** ()
- void **getFreeSpace** ()
- void **getMetadata** (const QString &folder, const QString &file)
- void **getPreview** ()
- CameraCommand \* **getThumbsInfo** (const CamItemInfoList &infoList, int thumbSize)
- void **listFiles** (const QString &folder, bool useMetadata)
- void **listFolders** (const QString &folder=QString())
- void **listRootFolder** (bool useMetadata)
- void **lockFile** (const QString &folder, const QString &file, bool lock)
- QIcon **mimeThumbnail** (const QString &itemName) const
- void **moveThumbsInfo** (CameraCommand \*const cmd)
- void **openFile** (const QString &folder, const QString &file)
- void **upload** (const QFileInfo &srcFileInfo, const QString &destFile, const QString &destFolder)

### Protected Member Functions

- void **executeCommand** (CameraCommand \*const cmd)
- void **run** () override

## 6.153.1 Member Function Documentation

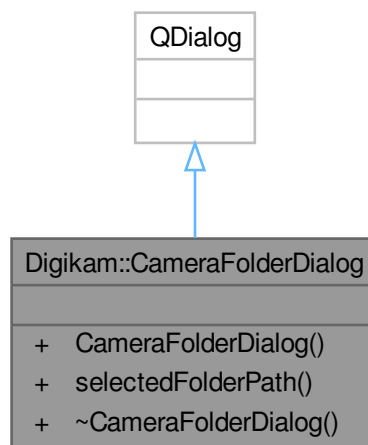
### 6.153.1.1 getThumbsInfo()

```
CameraCommand * Digikam::CameraController::getThumbsInfo (
    const CamItemInfoList & infoList,
    int thumbSize )
```

Get thumbnails for a list of camera items plus advanced information from metadata.

## 6.154 Digikam::CameraFolderDialog Class Reference

Inheritance diagram for Digikam::CameraFolderDialog:

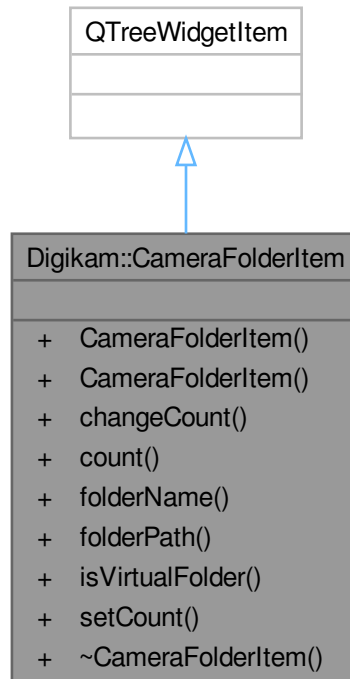


### Public Member Functions

- **CameraFolderDialog** (QWidget \*const parent, const QMap< QString, int > &map, const QString &cameraName, const QString &rootPath)
- QString **selectedFolderPath** () const

## 6.155 Digikam::CameraFolderItem Class Reference

Inheritance diagram for Digikam::CameraFolderItem:

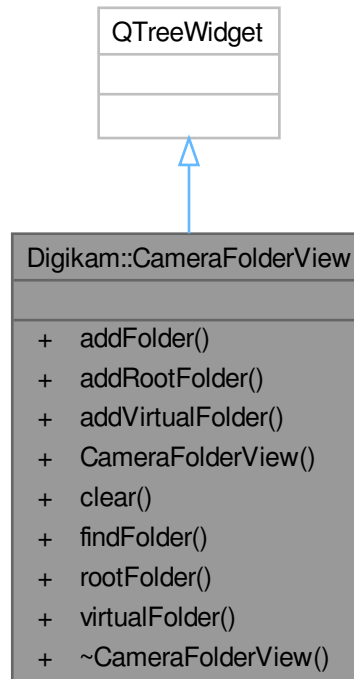


### Public Member Functions

- **CameraFolderItem** (QTreeWidgetItem \*const parent, const QString &name, const QIcon &icon=QIcon::fromTheme(QLatin1String("folder")))
- **CameraFolderItem** (QTreeWidgetItem \*const parent, const QString &folderName, const QString &folderPath, const QIcon &icon=QIcon::fromTheme(QLatin1String("folder")))
- void **changeCount** (int val)
- int **count** () const
- QString **folderName** () const
- QString **folderPath** () const
- bool **isVirtualFolder** () const
- void **setCount** (int val)

## 6.156 Digikam::CameraFolderView Class Reference

Inheritance diagram for Digikam::CameraFolderView:



### Signals

- void **signalCleared** ()
- void **signalFolderChanged** ([CameraFolderItem](#) \*)

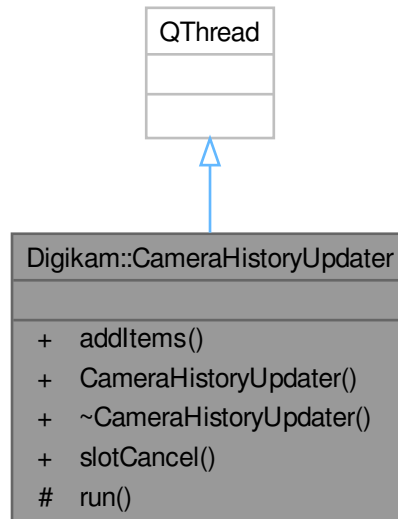
### Public Member Functions

- [CameraFolderItem](#) \* **addFolder** (const QString &folder, const QString &subFolder, int nbltems, const QIcon &icon=QIcon::fromTheme(QLatin1String("folder")))
- void **addRootFolder** (const QString &folder, int nbltems=-1, const QIcon &icon=QIcon::fromTheme(QLatin1String("folder")))
- void **addVirtualFolder** (const QString &name, const QIcon &icon=QIcon::fromTheme(QLatin1String("camera-photo")))
- **CameraFolderView** (QWidget \*const parent)
- virtual void **clear** ()
- [CameraFolderItem](#) \* **findFolder** (const QString &folderPath)
- [CameraFolderItem](#) \* **rootFolder** () const
- [CameraFolderItem](#) \* **virtualFolder** () const



## 6.157 Digikam::CameraHistoryUpdater Class Reference

Inheritance diagram for Digikam::CameraHistoryUpdater:



### Public Slots

- void **slotCancel** ()

### Signals

- void **signalBusy** (bool val)
- void **signalHistoryMap** (const CHUpdateItemMap &)

### Public Member Functions

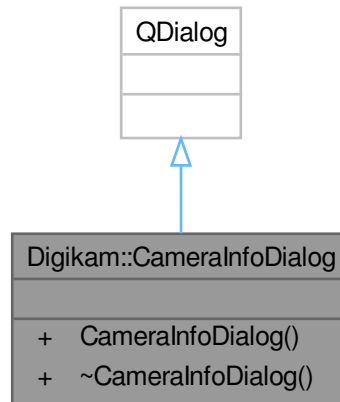
- void **addItems** (const QByteArray &id, CHUpdateItemMap &map)
- **CameraHistoryUpdater** (QWidget \*const parent)

### Protected Member Functions

- void **run** ()

## 6.158 Digikam::CameraInfoDialog Class Reference

Inheritance diagram for Digikam::CameraInfoDialog:

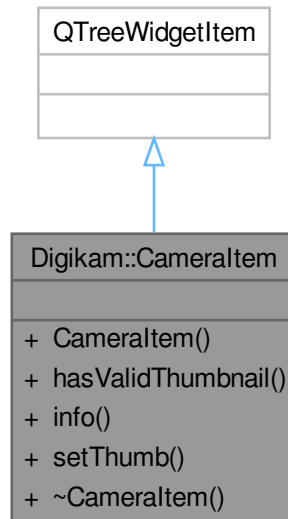


### Public Member Functions

- **CameraInfoDialog** (`QWidget *const parent, const QString &summary, const QString &>manual, const QString &about`)

## 6.159 Digikam::CamerItem Class Reference

Inheritance diagram for Digikam::CamerItem:

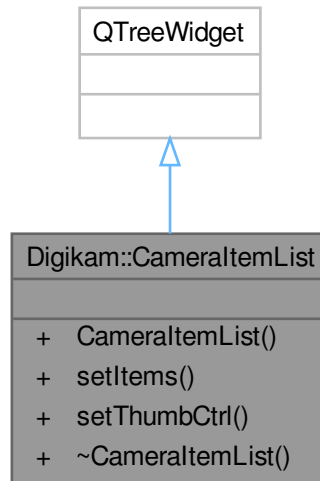


### Public Member Functions

- **CamerItem** (`QTreeWidgetItem *const parent`, `const CamItemInfo &info`)
- `bool hasValidThumbnail () const`
- `CamItemInfo info () const`
- `void setThumb (const QPixmap &pix, bool hasThumb=true)`

## 6.160 Digikam::CameralemList Class Reference

Inheritance diagram for Digikam::CameralemList:

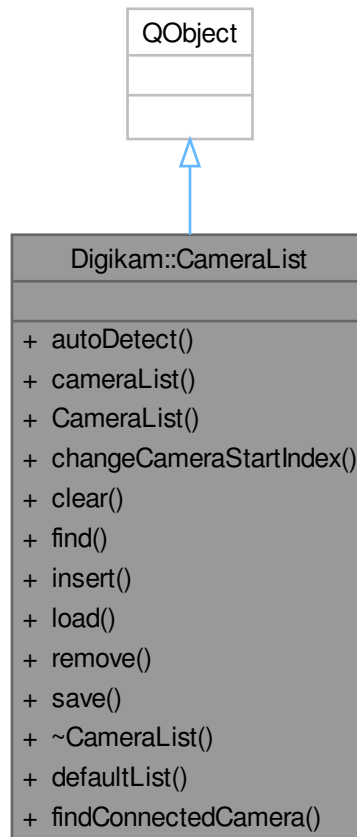


### Public Member Functions

- **CameralemList** (`QWidget *const parent=nullptr`)
- void **setItems** (`const CamItemInfoList &items`)
- void **setThumbCtrl** (`CameraThumbsCtrl *const ctrl`)

## 6.161 Digikam::CameraList Class Reference

Inheritance diagram for Digikam::CameraList:



### Signals

- void **signalCameraAdded** ([CameraType](#) \*)
- void **signalCameraRemoved** ([QAction](#) \*)

### Public Member Functions

- [CameraType](#) \* **autoDetect** (bool &retry)
- [QList](#)< [CameraType](#) \* > \* **cameraList** () const
- **CameraList** ([QObject](#) \*const parent, const [QString](#) &file)
- bool **changeCameraStartIndex** (const [QString](#) &cameraTitle, int startIndex)
- void **clear** ()
- [CameraType](#) \* **find** (const [QString](#) &title) const
- void **insert** ([CameraType](#) \*const ctype)
- bool **load** ()
- void **remove** ([CameraType](#) \*const ctype)
- bool **save** ()

### Static Public Member Functions

- static [CameraList](#) \* **defaultList** ()
- static bool **findConnectedCamera** (int vendorId, int productId, QString &model, QString &port)

## 6.162 Digikam::CameraMessageBox Class Reference

### Static Public Member Functions

- static void **informationList** ([CameraThumbsCtrl](#) \*const ctrl, QWidget \*const parent, const QString &caption, const QString &text, const CamItemInfoList &items, const QString &dontShowAgainName=QString())
- static int **warningContinueCancelList** ([CameraThumbsCtrl](#) \*const ctrl, QWidget \*const parent, const QString &caption, const QString &text, const CamItemInfoList &items, const QString &dontAskAgainName=QString())

### 6.162.1 Member Function Documentation

#### 6.162.1.1 informationList()

```
void Digikam::CameraMessageBox::informationList (
    CameraThumbsCtrl *const ctrl,
    QWidget *const parent,
    const QString & caption,
    const QString & text,
    const CamItemInfoList & items,
    const QString & dontShowAgainName = QString() ) [static]
```

Show List of camera items into an informative message box.

#### 6.162.1.2 warningContinueCancelList()

```
int Digikam::CameraMessageBox::warningContinueCancelList (
    CameraThumbsCtrl *const ctrl,
    QWidget *const parent,
    const QString & caption,
    const QString & text,
    const CamItemInfoList & items,
    const QString & dontAskAgainName = QString() ) [static]
```

Show List of camera items to process into a message box and wait user feedback. Return QMessageBox::Yes or QMessageBox::Cancel

## 6.163 Digikam::CameraNameHelper Class Reference

### Static Public Member Functions

- static QString **cameraName** (const QString &name)
- static QString **cameraNameAutoDetected** (const QString &name)
- static QString **createCameraName** (const QString &vendor, const QString &product=QString(), const QString &mode=QString(), bool autoDetected=false)
- static bool **sameDevices** (const QString &deviceA, const QString &deviceB)

## 6.164 Digikam::CameraNameOption Class Reference

Inheritance diagram for Digikam::CameraNameOption:



### Protected Member Functions

- QString `parseOperation` (`ParseSettings &settings`, const `QRegularExpressionMatch &match`) override

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void [setDescription](#) (const QString &desc)
- void [setIcon](#) (const QString &pixmap)
- void [setRegExp](#) (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum [IconType](#) { [Action](#) = 0 , [Dialog](#) }

## Signals inherited from [Digikam::Rule](#)

- void [signalTokenTriggered](#) (const QString &)

## Public Member Functions inherited from [Digikam::Option](#)

- [Option](#) (const QString &name, const QString &description)
- [Option](#) (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString [description](#) () const
- QPixmap [icon](#) (Rule::IconType type=Rule::Action) const
- bool [isValid](#) () const
- [ParseResults](#) [parse](#) ([ParseSettings](#) &settings)
- QRegularExpression & [regExp](#) () const
- QPushButton \* [registerButton](#) (QWidget \*parent)
- QAction \* [registerMenu](#) (QMenu \*parent)
- virtual void [reset](#) ()
- [Rule](#) (const QString &name)
- [Rule](#) (const QString &name, const QString &icon)
- TokenList & [tokens](#) () const
- bool [useTokenMenu](#) () const

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString [escapeToken](#) (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void [slotTokenTriggered](#) (const QString &)

### 6.164.1 Member Function Documentation

#### 6.164.1.1 [parseOperation\(\)](#)

```
QString Digikam::CameraNameOption::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [virtual]
```

TODO: describe me



## Parameters

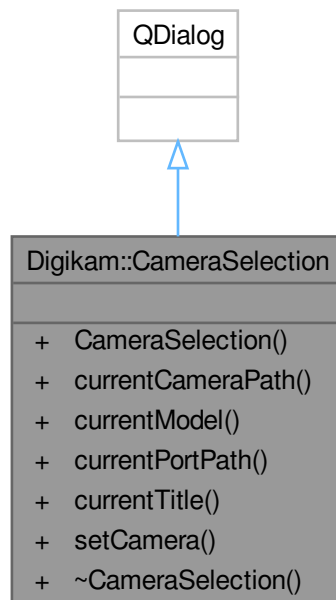
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in Option::parse()

## Returns

Implements [Digikam::Option](#).

## 6.165 Digikam::CameraSelection Class Reference

Inheritance diagram for Digikam::CameraSelection:



## Signals

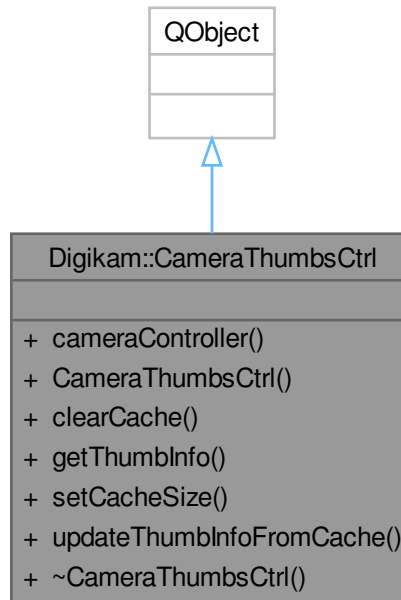
- void **signalOkClicked** (const QString &title, const QString &model, const QString &port, const QString &path)

## Public Member Functions

- **CameraSelection** (QWidget \*const parent=nullptr)
- QString **currentCameraPath** () const
- QString **currentModel** () const
- QString **currentPortPath** () const
- QString **currentTitle** () const
- void **setCamera** (const QString &title, const QString &model, const QString &port, const QString &path)

## 6.166 Digikam::CameraThumbsCtrl Class Reference

Inheritance diagram for Digikam::CameraThumbsCtrl:



### Signals

- void **signalThumbInfoReady** (const [CamItemInfo](#) &)

### Public Member Functions

- [CameraController](#) \* **cameraController** () const
- **CameraThumbsCtrl** ([CameraController](#) \*const ctrl, QWidget \*const parent)
- void **clearCache** ()
- bool **getThumbInfo** (const [CamItemInfo](#) &info, CachedItem &item) const
- void **setCacheSize** (int numberOfItems)
- void **updateThumbInfoFromCache** (const [CamItemInfo](#) &info)

## 6.166.1 Member Function Documentation

### 6.166.1.1 cameraController()

```
CameraController * Digikam::CameraThumbsCtrl::cameraController ( ) const
```

Return camera controller instance.

### 6.166.1.2 getThumbInfo()

```
bool Digikam::CameraThumbsCtrl::getThumbInfo (
    const CamItemInfo & info,
    CachedItem & item ) const
```

Fill item with relevant information. if item is not in cache, return false and information will be dispatched later through signalThumbInfoReady(), else return true and information is available immediately.

### 6.166.1.3 updateThumbInfoFromCache()

```
void Digikam::CameraThumbsCtrl::updateThumbInfoFromCache (
    const CamItemInfo & info )
```

Force controller to update info from device in cache.

## 6.167 Digikam::CameraType Class Reference

### Public Member Functions

- QAction \* **action** () const
- **CameraType** (const [CameraType](#) &ctype)
- **CameraType** (const QString &title, const QString &model, const QString &port, const QString &path, int startingNumber, QAction \*const action=nullptr)
- [ImportUI](#) \* **currentImportUI** () const
- QString **model** () const
- [CameraType](#) & **operator=** (const [CameraType](#) &type)
- QString **path** () const
- QString **port** () const
- void **setAction** (QAction \*const action)
- void **setCurrentImportUI** ([ImportUI](#) \*const importui)
- void **setModel** (const QString &model)
- void **setPath** (const QString &path)
- void **setPort** (const QString &port)
- void **setStartingNumber** (int sn)
- void **setTitle** (const QString &title)
- void **setValid** (bool valid)
- int **startingNumber** () const
- QString **title** () const
- bool **valid** () const

## 6.168 Digikam::CamItemInfo Class Reference

### Public Types

- enum [DownloadStatus](#) {
  - [DownloadUnknown](#) = -1 , [DownloadedNo](#) = 0 , [DownloadedYes](#) = 1 , [DownloadFailed](#) = 2 ,
  - [DownloadStarted](#) = 3 , [NewPicture](#) = 4 }

## Public Member Functions

- bool `isNull ()` const
- bool `operator!= (const CamItemInfo &info)` const
- bool `operator== (const CamItemInfo &info)` const
- `QUrl url ()` const

## Public Attributes

- int **colorLabel** = NoColorLabel  
*Pre-picklabel value of camera file.*
- `QDateTime` **ctime**  
*Created time stamp of camera file.*
- int **downloaded** = DownloadUnknown  
*Variable values depending of user actions.*
- `QString` **downloadName**  
*Preview of the file-name to use during download from camera.*
- `QString` **folder**  
*Folder path to access to file in camera.*
- int **height** = -1  
*Image height in pixels.*
- `qulonglong` **id** = -1  
*Unique image id.*
- `QString` **mime**  
*Type mime of camera file.*
- `QString` **name**  
*File name in camera file-system.*
- `PhotoInfoContainer` **photoInfo**  
*Photo Info from camera file (get from file metadata)*
- int **pickLabel** = NoPickLabel  
*Pre-picklabel value of camera file.*
- bool **previewPossible** = false
- int **rating** = NoRating  
*Pre-rating value of camera file.*
- int **readPermissions** = -1  
*Read permission of camera file.*
- `qint64` **size** = -1  
*Static values taken from camera.*
- `QList< int >` **tagIds**  
*Pre-tags ids of camera file.*
- int **width** = -1  
*Image width in pixels.*
- int **writePermissions** = -1  
*Write permission of camera file.*

## 6.168.1 Member Enumeration Documentation

### 6.168.1.1 DownloadStatus

enum `Digikam::CamItemInfo::DownloadStatus`

## Enumerator

DownloadUnknown	Download state is unknown.
DownloadedNo	Is not yet downloaded on computer.
DownloadedYes	Is already downloaded on computer.
DownloadFailed	Download is failed or have been aborted by user.
DownloadStarted	Download is under progress.
NewPicture	This is a new item from camera.

## 6.168.2 Member Function Documentation

### 6.168.2.1 isNull()

```
bool Digikam::CamItemInfo::isNull ( ) const
```

Return true if all member in this container are null.

### 6.168.2.2 operator!=(())

```
bool Digikam::CamItemInfo::operator!= (
    const CamItemInfo & info ) const
```

Compare for camera information un-equality, not including variable values.

### 6.168.2.3 operator==(())

```
bool Digikam::CamItemInfo::operator== (
    const CamItemInfo & info ) const
```

Compare for camera information equality, not including variable values.

### 6.168.2.4 url()

```
QUrl Digikam::CamItemInfo::url ( ) const
```

Return the local file system (mounted on computer) url to the camera file.

## 6.168.3 Member Data Documentation

### 6.168.3.1 downloaded

```
int Digikam::CamItemInfo::downloaded = DownloadUnknown
```

Download status of camera file. See DownloadStatus enum for details

### 6.168.3.2 size

```
qint64 Digikam::CamItemInfo::size = -1
```

Camera file size in bytes.

## 6.169 Digikam::CamItemSortSettings Class Reference

### Public Types

- enum **CategorizationMode** { **NoCategories** , **CategoryByFolder** , **CategoryByFormat** , **CategoryByDate** }
- enum **SortOrder** { **AscendingOrder** = Qt::AscendingOrder , **DescendingOrder** = Qt::DescendingOrder , **DefaultOrder** }
- enum **SortRole** { **SortByFileName** , **SortByFilePath** , **SortByCreationDate** , **SortByFileSize** , **SortByDownloadState** , **SortByRating** }

### Public Member Functions

- int **compare** (const [CamItemInfo](#) &left, const [CamItemInfo](#) &right) const
- int **compare** (const [CamItemInfo](#) &left, const [CamItemInfo](#) &right, SortRole sortRole) const
- int **compareCategories** (const [CamItemInfo](#) &left, const [CamItemInfo](#) &right) const
- bool **isCategorized** () const
- bool **lessThan** (const [CamItemInfo](#) &left, const [CamItemInfo](#) &right) const
- bool **lessThan** (const QVariant &left, const QVariant &right) const
- bool **operator==** (const [CamItemSortSettings](#) &other) const
- void **setCategorizationMode** (CategorizationMode mode)
- void **setCategorizationSortOrder** ([SortOrder](#) order)
- void **setSortOrder** ([SortOrder](#) order)
- void **setSortRole** (SortRole role)
- void **setStringTypeNatural** (bool natural)

### Static Public Member Functions

- template<typename T >  
static int **compareByOrder** (const T &a, const T &b, Qt::SortOrder [sortOrder](#))
- static int **compareByOrder** (int compareResult, Qt::SortOrder [sortOrder](#))
- template<typename T >  
static int **compareValue** (const T &a, const T &b)
- static Qt::SortOrder **defaultSortOrderForCategorizationMode** (CategorizationMode mode)
- static Qt::SortOrder **defaultSortOrderForSortRole** (SortRole role)
- template<typename T >  
static bool **lessThanByOrder** (const T &a, const T &b, Qt::SortOrder [sortOrder](#))
- static int **naturalCompare** (const QString &a, const QString &b, Qt::SortOrder [sortOrder](#), Qt::CaseSensitivity caseSensitive=Qt::CaseSensitive, bool natural=true)

## Public Attributes

- Qt::CaseSensitivity **categorizationCaseSensitivity** = Qt::CaseSensitive
- CategorizationMode **categorizationMode** = NoCategories
- [SortOrder](#) **categorizationSortOrder** = [DefaultOrder](#)
- Qt::SortOrder **currentCategorizationSortOrder** = Qt::AscendingOrder  
*Only Ascending or Descending, never be DefaultOrder.*
- Qt::SortOrder **currentSortOrder** = Qt::AscendingOrder
- Qt::CaseSensitivity **sortCaseSensitivity** = Qt::CaseSensitive
- [SortOrder](#) **sortOrder** = [DefaultOrder](#)  
*Camera Items Sorting.*
- SortRole **sortRole** = SortByFileName
- bool **strTypeNatural** = true

## 6.169.1 Member Enumeration Documentation

### 6.169.1.1 SortOrder

```
enum Digikam::CamItemSortSettings::SortOrder
```

#### Enumerator

DefaultOrder	sort order depends on the chosen sort role
--------------	--

## 6.169.2 Member Function Documentation

### 6.169.2.1 compare()

```
int Digikam::CamItemSortSettings::compare (
    const CamItemInfo & left,
    const CamItemInfo & right ) const
```

Compares the camItemInfos left and right. Return -1 if left is less than right, 1 if left is greater than right, and 0 if left equals right comparing the current sort role's value. Adheres to set sort role and sort order.

### 6.169.2.2 compareByOrder()

```
static int Digikam::CamItemSortSettings::compareByOrder (
    int compareResult,
    Qt::SortOrder sortOrder ) [inline], [static]
```

Takes a typical result from a compare method (0 is equal, -1 is less than, 1 is greater than) and applies the given sort order to it.

### 6.169.2.3 compareCategories()

```
int Digikam::CamItemSortSettings::compareCategories (
    const CamItemInfo & left,
    const CamItemInfo & right ) const
```

Compares the categories of left and right camItemInfos. It returns -1 if the left camItemInfo is less than right, and 0 if both fall in the same category, and 1 if the left camItemInfo is greater than right. Adheres to set categorization mode and current category sort order.

### 6.169.2.4 compareValue()

```
template<typename T >
static int Digikam::CamItemSortSettings::compareValue (
    const T & a,
    const T & b ) [inline], [static]
```

Returns the usual compare result of -1, 0, or 1 for lessThan, equals and greaterThan.

### 6.169.2.5 lessThan() [1/2]

```
bool Digikam::CamItemSortSettings::lessThan (
    const CamItemInfo & left,
    const CamItemInfo & right ) const
```

Returns true if left is less than right. Adheres to current sort role and sort order.

### 6.169.2.6 lessThan() [2/2]

```
bool Digikam::CamItemSortSettings::lessThan (
    const QVariant & left,
    const QVariant & right ) const
```

Returns true if left QVariant is less than right. Adheres to current sort role and sort order.

### 6.169.2.7 lessThanByOrder()

```
template<typename T >
static bool Digikam::CamItemSortSettings::lessThanByOrder (
    const T & a,
    const T & b,
    Qt::SortOrder sortOrder ) [inline], [static]
```

Returns  $a < b$  if sortOrder is Ascending, or  $b < a$  if order is descending



### 6.169.2.8 naturalCompare()

```
static int Digikam::CamItemSortSettings::naturalCompare (
    const QString & a,
    const QString & b,
    Qt::SortOrder sortOrder,
    Qt::CaseSensitivity caseSensitive = Qt::CaseSensitive,
    bool natural = true ) [inline], [static]
```

Compares the two string by natural comparison and adheres to given sort order

## 6.170 Digikam::Canvas Class Reference

Inheritance diagram for Digikam::Canvas:



### Public Slots

- void **slotCopy** ()
- void **slotCrop** ()

- void **slotFlipHoriz** ()
- void **slotFlipVert** ()
- void **slotRedo** (int steps=1)
- void **slotRestore** ()
- void **slotRotate180** ()
- void **slotRotate270** ()
- void **slotRotate90** ()
- *image modifiers*
- void **slotSelectAll** ()
- void **slotSelected** ()
- void **slotSelectionMoved** ()
- void **slotSelectNone** ()
- void **slotUndo** (int steps=1)

## Signals

- void **signalAddedDroppedItems** (QDropEvent \*)
- void **signalChanged** ()
- void **signalLoadingFinished** (const QString &filename, bool success)
- void **signalLoadingProgress** (const QString &filePath, float progress)
- void **signalLoadingStarted** (const QString &filename)
- void **signalPrepareToLoad** ()
- void **signalRedoSteps** (int)
- void **signalRightButtonClicked** ()
- void **signalSavingFinished** (const QString &filename, bool success)
- void **signalSavingProgress** (const QString &filePath, float progress)
- void **signalSavingStarted** (const QString &filename)
- void **signalSelected** (bool)
- void **signalSelectionChanged** (const QRect &)
- void **signalSelectionSetText** (const QRect &)
- void **signalShowNextImage** ()
- void **signalShowPrevImage** ()
- void **signalToggleOffFitToWindow** ()
- void **signalUndoSteps** (int)
- void **signalZoomChanged** (double)

## Signals inherited from [Digikam::GraphicsDImgView](#)

- void **activated** ()
- void **contentsMoved** (bool panningFinished)
- void **contentsMoving** (int, int)
- void **leftButtonClicked** ()
- void **leftButtonDoubleClicked** ()
- void **resized** ()
- void **rightButtonClicked** ()
- void **toNextImage** ()
- void **toPreviousImage** ()
- void **viewportRectChanged** (const QRectF &viewportRect)

## Public Member Functions

- void **abortSaving** ()
- void **applyTransform** (const [IccTransform](#) &transform)
- **Canvas** (QWidget \*const parent=nullptr)
- [DImg](#) **currentImage** () const
- QString **currentImageFileFormat** () const
- QString **currentImageFilePath** () const
- QString **ensureHasCurrentUuid** () const
- bool **exifRotated** () const
- void **fitToSelect** ()
- QRect **getSelectedArea** () const
- int **imageHeight** () const
- int **imageWidth** () const
- [EditorCore](#) \* **interface** () const
- bool **isReadOnly** () const
- void **load** (const QString &filename, [IOFileSettings](#) \*const [IOFileSettings](#))
- void **makeDefaultEditingCanvas** ()
- void **preload** (const QString &filename)
- void **resetImage** ()
- void **setExifOrient** (bool exifOrient)
- void **setExposureSettings** ([ExposureSettingsContainer](#) \*const expoSettings)
- void **setICCSettings** (const [ICCSettingsContainer](#) &cmSettings)
- void **setModified** ()
- void **setSoftProofingEnabled** (bool enable)

## Public Member Functions inherited from [Digikam::GraphicsDImgView](#)

- int **contentsX** () const
- int **contentsY** () const
- void **drawText** (QPainter \*p, const QRectF &rect, const QString &text)
- void **fitToWindow** ()
- [GraphicsDImgView](#) (QWidget \*const parent=nullptr)
- [GraphicsDImgItem](#) \* **item** () const
- [SinglePhotoPreviewLayout](#) \* **layout** () const
- [DImgPreviewItem](#) \* **previewItem** () const
- void **scrollPointOnPoint** (const QPointF &scenePos, const QPoint &viewportPos)
- void **setContentPos** (int x, int y)
- void **setItem** ([GraphicsDImgItem](#) \*const item)
- void **toggleFullScreen** (bool set)
- QRect **visibleArea** () const

## Protected Member Functions

- void **addRubber** ()
- void **dragEnterEvent** (QDragEnterEvent \*) override
- void **dragMoveEvent** (QDragMoveEvent \*) override
- void **dropEvent** (QDropEvent \*) override
- void **keyPressEvent** (QKeyEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override

## Protected Member Functions inherited from [Digikam::GraphicsDImgView](#)

- virtual bool **acceptsMouseClicked** (QMouseEvent \*e)
- void **continuePanning** (const QPoint &pos)
- void **drawForeground** (QPainter \*painter, const QRectF &rect) override
- void **finishPanning** ()
- void **installPanIcon** ()
- void **mouseDoubleClickEvent** (QMouseEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override
- void **scrollContentsBy** (int dx, int dy) override
- void **setScaleFitToWindow** (bool value)
- void **setShowText** (bool value)
- void **startPanning** (const QPoint &pos)
- void **wheelEvent** (QWheelEvent \*) override

### Additional Inherited Members

## Protected Slots inherited from [Digikam::GraphicsDImgView](#)

- void **slotContentsMoved** ()
- void **slotCornerButtonPressed** ()
- void **slotPanIconHidden** ()
- virtual void **slotPanIconSelectionMoved** (const QRect &, bool)

## 6.170.1 Member Function Documentation

### 6.170.1.1 applyTransform()

```
void Digikam::Canvas::applyTransform (
    const IccTransform & transform )
```

Apply Color Management transformation to image (typically working color space).

### 6.170.1.2 currentImage()

```
DImg Digikam::Canvas::currentImage ( ) const
```

Return a copy of current image loaded in editor.

### 6.170.1.3 currentImageFileFormat()

```
QString Digikam::Canvas::currentImageFileFormat ( ) const
```

Return the type mime of current image loaded in editor.

#### 6.170.1.4 currentImagePath()

```
QString Digikam::Canvas::currentImagePath ( ) const
```

Return the file path of current image loaded in editor.

#### 6.170.1.5 exifRotated()

```
bool Digikam::Canvas::exifRotated ( ) const
```

Return true if image have been rotated following Exif information.

#### 6.170.1.6 fitToSelect()

```
void Digikam::Canvas::fitToSelect ( )
```

Change zoom level to fit current selection on canvas size.

#### 6.170.1.7 getSelectedArea()

```
QRect Digikam::Canvas::getSelectedArea ( ) const
```

Return the rectangle information of current canvas selection.

#### 6.170.1.8 imageHeight()

```
int Digikam::Canvas::imageHeight ( ) const
```

Return the height of current image loaded in editor.

#### 6.170.1.9 imageWidth()

```
int Digikam::Canvas::imageWidth ( ) const
```

Return the width of current image loaded in editor.

#### 6.170.1.10 interface()

```
EditorCore * Digikam::Canvas::interface ( ) const
```

Return the core interface instance of editor.

#### 6.170.1.11 isReadOnly()

```
bool Digikam::Canvas::isReadOnly ( ) const
```

If current image file format is only available in read only, typically all RAW image file formats.

**6.170.1.12 setExifOrient()**

```
void Digikam::Canvas::setExifOrient (
    bool exifOrient )
```

Rotate image following Exif information.

**6.170.1.13 setExposureSettings()**

```
void Digikam::Canvas::setExposureSettings (
    ExposureSettingsContainer *const expoSettings )
```

Apply under.over exposure indicator settings.

**6.170.1.14 setICCSettings()**

```
void Digikam::Canvas::setICCSettings (
    const ICCSettingsContainer & cmSettings )
```

Apply Color management settings (typically screen profile).

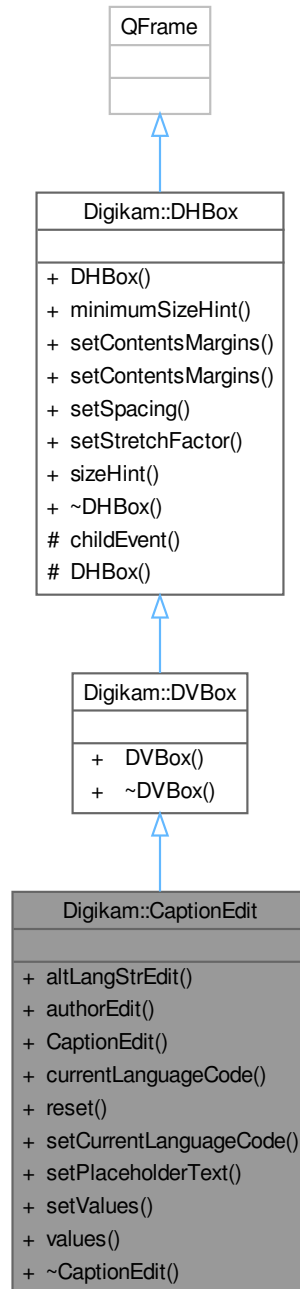
**6.170.1.15 setSoftProofingEnabled()**

```
void Digikam::Canvas::setSoftProofingEnabled (
    bool enable )
```

Turn on/off Color Management Soft proofing mode.

## 6.171 Digikam::CaptionEdit Class Reference

Inheritance diagram for Digikam::CaptionEdit:



### Signals

- void **signalModified** ()



## Public Member Functions

- [AltLangStrEdit](#) \* **altLangStrEdit** () const
- [QLineEdit](#) \* **authorEdit** () const
- **CaptionEdit** ([QWidget](#) \*const parent)
- [QString](#) **currentLanguageCode** () const
- void **reset** ()
- void **setCurrentLanguageCode** (const [QString](#) &lang)
- void **setPlaceholderText** (const [QString](#) &msg)
- void **setValues** (const [CaptionsMap](#) &values)
- [CaptionsMap](#) & **values** () const

## Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** ([QWidget](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** ([QWidget](#) \*const parent=nullptr)
- [QSize](#) **minimumSizeHint** () const override
- void **setContentsMargins** (const [QMargins](#) &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** ([QWidget](#) \*const widget, int stretch)
- [QSize](#) **sizeHint** () const override

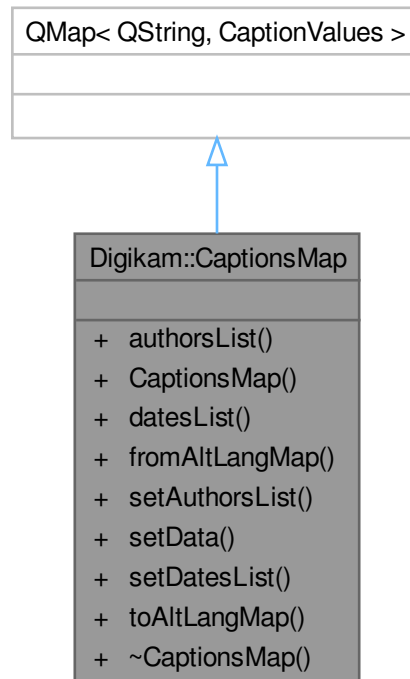
## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** ([QChildEvent](#) \*e) override
- **DHBox** (bool vertical, [QWidget](#) \*const parent)

## 6.172 Digikam::CaptionsMap Class Reference

Inheritance diagram for Digikam::CaptionsMap:



### Public Member Functions

- [MetaEngine::AltLangMap](#) **authorsList** () const
- [MetaEngine::AltLangMap](#) **datesList** () const
- void **fromAltLangMap** (const [MetaEngine::AltLangMap](#) &map)
- void **setAuthorsList** (const [MetaEngine::AltLangMap](#) &map, const QString &commonAuthor=QString())
- void **setData** (const [MetaEngine::AltLangMap](#) &comments, const [MetaEngine::AltLangMap](#) &authors, const QString &commonAuthor, const [MetaEngine::AltLangMap](#) &dates)
- void **setDatesList** (const [MetaEngine::AltLangMap](#) &map)
- [MetaEngine::AltLangMap](#) **toAltLangMap** () const

### 6.172.1 Detailed Description

A map used to store a list of Alternative Language values + author and date properties The map key is the language code following RFC3066 notation (like "fr-FR" for French), and the [CaptionsMap](#) value all caption properties.

## 6.172.2 Member Function Documentation

### 6.172.2.1 setAuthorsList()

```
void Digikam::CaptionsMap::setAuthorsList (
    const MetaEngine::AltLangMap & map,
    const QString & commonAuthor = QString() )
```

Sets the author for the comments in the specified languages. If commonAuthor is not null, it will be used to set the author of all comments for which the author is not specified in the map.

## 6.173 Digikam::CaptionValues Class Reference

### Public Member Functions

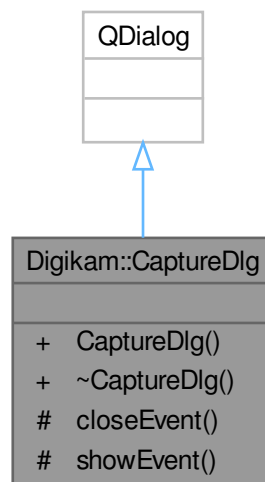
- bool **operator==** (const [CaptionValues](#) &val) const

### Public Attributes

- QString **author**
- QString **caption**
- QDateTime **date**

## 6.174 Digikam::CaptureDlg Class Reference

Inheritance diagram for Digikam::CaptureDlg:



### Public Member Functions

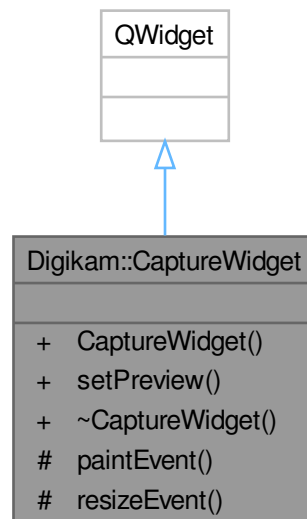
- **CaptureDlg** (QWidget \*const parent, [CameraController](#) \*const controller, const QString &cameraTitle)

### Protected Member Functions

- void **closeEvent** (QCloseEvent \*e) override
- void **showEvent** (QShowEvent \*e) override

## 6.175 Digikam::CaptureWidget Class Reference

Inheritance diagram for Digikam::CaptureWidget:



### Public Member Functions

- **CaptureWidget** (QWidget \*const parent=nullptr)
- void **setPreview** (const QImage &preview)

### Protected Member Functions

- void **paintEvent** (QPaintEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override

## 6.176 Digikam::CaseModifier Class Reference

Inheritance diagram for Digikam::CaseModifier:



### Public Member Functions

- QString [parseOperation](#) (ParseSettings &settings, const QRegularExpressionMatch &match) override

## Public Member Functions inherited from [Digikam::Modifier](#)

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- [ParseResults](#) **parse** ([ParseSettings](#) &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from [Digikam::Rule](#)

- void **signalTokenTriggered** (const QString &)

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool **addToken** (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void **setUseTokenMenu** (bool value)

## 6.176.1 Member Function Documentation

### 6.176.1.1 parseOperation()

```
QString Digikam::CaseModifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [virtual]
```

TODO: describe me

## Parameters

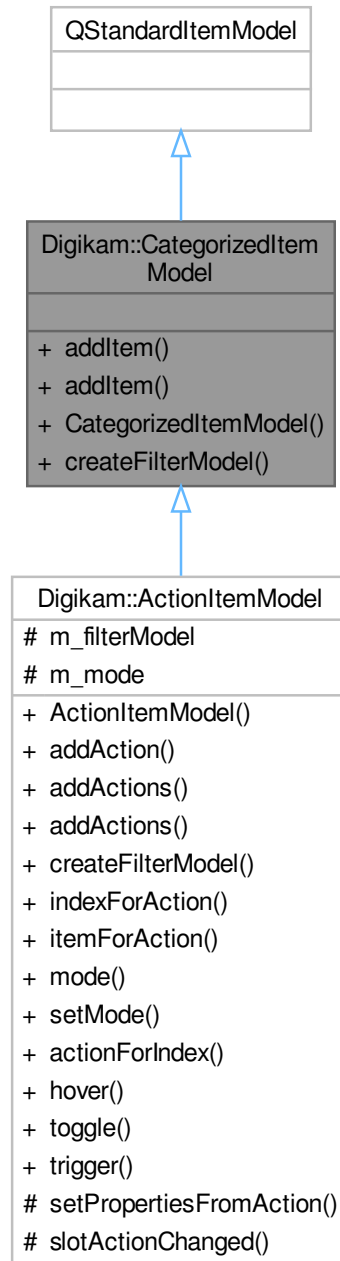
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in Option::parse()

## Returns

Implements [Digikam::Modifier](#).

## 6.177 Digikam::CategorizedItemModel Class Reference

Inheritance diagram for Digikam::CategorizedItemModel:



### Public Types

- enum `ExtraRoles` { `ItemOrderRole` = `Qt::UserRole + 1` }



## Public Member Functions

- `QStandardItem * addItem` (`const QString &text`, `const QIcon &decoration`, `const QVariant &category`, `const QVariant &categorySorting=QVariant()`)
- `QStandardItem * addItem` (`const QString &text`, `const QVariant &category`, `const QVariant &categorySorting=QVariant()`)
- `CategorizedItemModel` (`QObject *const parent=nullptr`)
- virtual `DCategorizedSortFilterProxyModel * createFilterModel ()`

## 6.177.1 Member Enumeration Documentation

### 6.177.1.1 ExtraRoles

enum `Digikam::CategorizedItemModel::ExtraRoles`

#### Enumerator

<code>ItemOrderRole</code>	This role, per default, reflects the order in which items are added.
----------------------------	--

## 6.178 Digikam::CBContainer Class Reference

### Public Attributes

- double `alpha` = 1.0
- double `blue` = 1.0
- double `gamma` = 1.0
- double `green` = 1.0
- double `red` = 1.0

## 6.179 Digikam::CBFilter Class Reference

Inheritance diagram for Digikam::CBFilter:



### Public Member Functions

- **CBFilter** (const [CBContainer](#) &settings, [DimgThreadedFilter](#) \*const master, const [Dimg](#) &orgImage, [Dimg](#) &destImage, int progressBegin=0, int progressEnd=100)

- **CBFilter** (*DImg* \*const orgImage, *QObject* \*const parent=nullptr, const [CBCContainer](#) &settings=[CBCContainer](#)())
- **CBFilter** (*QObject* \*const parent=nullptr)
- [FilterAction](#) filterAction () override
- *QString* filterIdentifier () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) (*DImg* \*const orgImage, *QObject* \*const parent, const *QString* &name=*QString*())
- [DImgThreadedFilter](#) (*QObject* \*const parent=nullptr, const *QString* &name=*QString*())
- const *QString* & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- *QList*< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual *QString* **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const *QString* &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual *QList*< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (*QObject* \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- *QThread*::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (*QThread*::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static *QString* **DisplayName** ()
- static *QString* **FilterIdentifier** ()
- static *QList*< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.179.1 Member Function Documentation

### 6.179.1.1 filterAction()

`FilterAction` Digikam::CBFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.179.1.2 filterIdentifier()

`QString` Digikam::CBFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

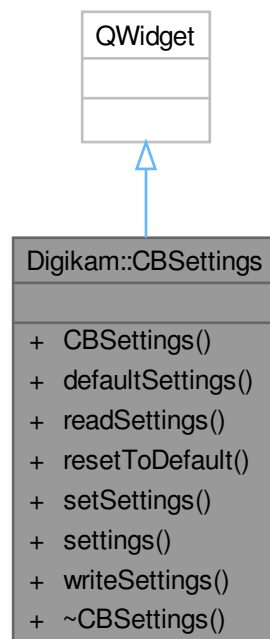
### 6.179.1.3 readParameters()

```
void Digikam::CBFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.180 Digikam::CBSettings Class Reference

Inheritance diagram for Digikam::CBSettings:



## Signals

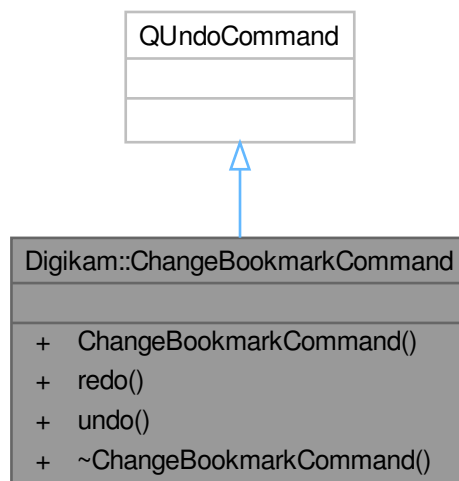
- void **signalSettingsChanged** ()

## Public Member Functions

- **CBSettings** (QWidget \*const parent)
- **CBContainer defaultSettings** () const
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const **CBContainer** &settings)
- **CBContainer settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.181 Digikam::ChangeBookmarkCommand Class Reference

Inheritance diagram for Digikam::ChangeBookmarkCommand:



## Public Types

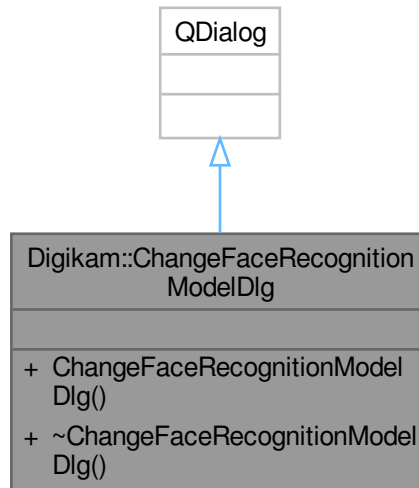
- enum **BookmarkData** { **Url** = 0 , **Title** , **Desc** }

## Public Member Functions

- **ChangeBookmarkCommand** (**BookmarksManager** \*const mngr, **BookmarkNode** \*const node, const QString &newValue, **BookmarkData** type)
- void **redo** () override
- void **undo** () override

## 6.182 Digikam::ChangeFaceRecognitionModelDlg Class Reference

Inheritance diagram for Digikam::ChangeFaceRecognitionModelDlg:



### Public Member Functions

- **ChangeFaceRecognitionModelDlg** (`QWidget *const parent`, [FaceScanSettings::FaceRecognitionModel newModel](#))

## 6.183 Digikam::ChangingDB Class Reference

### Public Member Functions

- **ChangingDB** ([AlbumManager::Private \\*const dd](#))
- **ChangingDB** ([CollectionManager::Private \\*const dd](#))
- **ChangingDB** ([TagsCache::Private \\*const dd](#))

### Public Attributes

- [AlbumManager::Private \\*const d](#) = nullptr
- [CollectionManager::Private \\*const d](#) = nullptr
- [TagsCache::Private \\*const d](#) = nullptr

## 6.184 Digikam::CharcoalFilter Class Reference

Inheritance diagram for Digikam::CharcoalFilter:



### Public Member Functions

- **CharcoalFilter** (`Dimg *const orgImage`, `QObject *const parent=nullptr`, `double pencil=5.0`, `double smooth=10.0`)



- **CharcoalFilter** (QObject \*const parent=nullptr)
- [FilterAction filterAction](#) () override
- QString [filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) (DImg \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static QString [DisplayableName](#) ()
- static QString [FilterIdentifier](#) ()
- static QList< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.184.1 Member Function Documentation

### 6.184.1.1 filterAction()

`FilterAction` Digikam::CharcoalFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.184.1.2 filterIdentifier()

`QString` Digikam::CharcoalFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

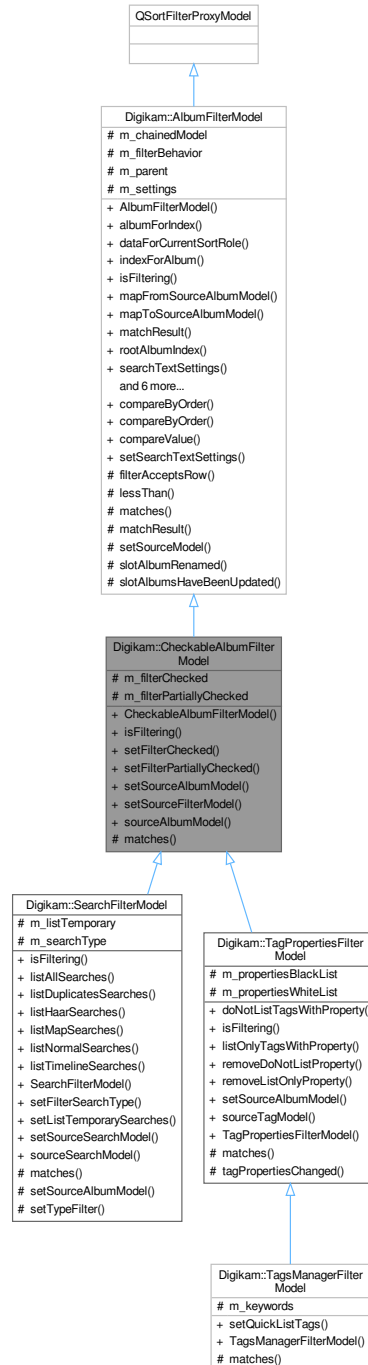
### 6.184.1.3 readParameters()

```
void Digikam::CharcoalFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.185 Digikam::CheckableAlbumFilterModel Class Reference

Inheritance diagram for Digikam::CheckableAlbumFilterModel:



### Public Member Functions

- `CheckableAlbumFilterModel` (`QObject *const parent=nullptr`)
- `bool isFiltering ()` const override

- void **setFilterChecked** (bool filter)
- void **setFilterPartiallyChecked** (bool filter)
- void **setSourceAlbumModel** ([AbstractCheckableAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([CheckableAlbumFilterModel](#) \*const source)
- [AbstractCheckableAlbumModel](#) \* **sourceAlbumModel** () const

## Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- **AlbumFilterModel** (QObject \*const parent=nullptr)
- [Album](#) \* **albumForIndex** (const QModelIndex &index) const  
*Convenience methods.*
- QVariant **dataForCurrentSortRole** ([Album](#) \*album) const
- QModelIndex **indexForAlbum** ([Album](#) \*album) const
- QModelIndex **mapFromSourceAlbumModel** (const QModelIndex &index) const
- QModelIndex **mapToSourceAlbumModel** (const QModelIndex &index) const
- [MatchResult](#) **matchResult** (const QModelIndex &index) const
- QModelIndex **rootAlbumIndex** () const
- [SearchTextSettings](#) **searchTextSettings** () const
- void **setFilterBehavior** ([FilterBehavior](#) behavior)
- void **setSourceAlbumModel** ([AbstractAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([AlbumFilterModel](#) \*const source)
- [AbstractAlbumModel](#) \* **sourceAlbumModel** () const
- [AlbumFilterModel](#) \* **sourceFilterModel** () const
- void **updateFilter** ()

## Protected Member Functions

- bool **matches** ([Album](#) \*album) const override

## Protected Member Functions inherited from [Digikam::AlbumFilterModel](#)

- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- [MatchResult](#) **matchResult** ([Album](#) \*album) const
- void **setSourceModel** (QAbstractItemModel \*const model) override

## Protected Attributes

- bool **m\_filterChecked** = false
- bool **m\_filterPartiallyChecked** = false

## Protected Attributes inherited from [Digikam::AlbumFilterModel](#)

- QPointer< [AlbumFilterModel](#) > **m\_chainedModel** = nullptr
- [FilterBehavior](#) **m\_filterBehavior** = [FullFiltering](#)
- QObject \* **m\_parent** = nullptr
- [SearchTextSettings](#) **m\_settings**

## Additional Inherited Members

### Public Types inherited from [Digikam::AlbumFilterModel](#)

- enum [FilterBehavior](#) { [SimpleFiltering](#) , [FullFiltering](#) , [StrictFiltering](#) }
- enum [MatchResult](#) { [NoMatch](#) = 0 , [DirectMatch](#) , [ParentMatch](#) , [ChildMatch](#) , [SpecialMatch](#) }

### Public Slots inherited from [Digikam::AlbumFilterModel](#)

- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)

### Signals inherited from [Digikam::AlbumFilterModel](#)

- void [hasSearchResult](#) (bool hasResult)
- void [searchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [searchTextSettingsChanged](#) (bool wasSearching, bool searched)
- void [signalFilterChanged](#) ()

### Static Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- template<typename T >  
static int [compareByOrder](#) (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int [compareByOrder](#) (int compareResult, Qt::SortOrder sortOrder)
- template<typename T >  
static int [compareValue](#) (const T &a, const T &b)

### Protected Slots inherited from [Digikam::AlbumFilterModel](#)

- void [slotAlbumRenamed](#) ([Album](#) \*album)
- void [slotAlbumsHaveBeenUpdated](#) (int type)

## 6.185.1 Detailed Description

[Filter](#) model for checkable album models that allows more filtering options based on check state.

## 6.185.2 Member Function Documentation

### 6.185.2.1 [isFiltering\(\)](#)

```
bool Digikam::CheckableAlbumFilterModel::isFiltering ( ) const [override], [virtual]
```

Returns if the currently applied filters will result in any filtering.

#### Returns

`true` if the current selected filter could result in any filtering without checking if this really happens.

Reimplemented from [Digikam::AlbumFilterModel](#).

Reimplemented in [Digikam::SearchFilterModel](#), and [Digikam::TagPropertiesFilterModel](#).

### 6.185.2.2 matches()

```
bool Digikam::CheckableAlbumFilterModel::matches (  
    Album * album ) const [override], [protected], [virtual]
```

This method provides the basic match checking algorithm. Return true if this single album matches the current criteria. This method can be overridden to provide custom filtering.

**Parameters**

<i>album</i>	the album to tell if it matches the filter criteria or not.
--------------	---

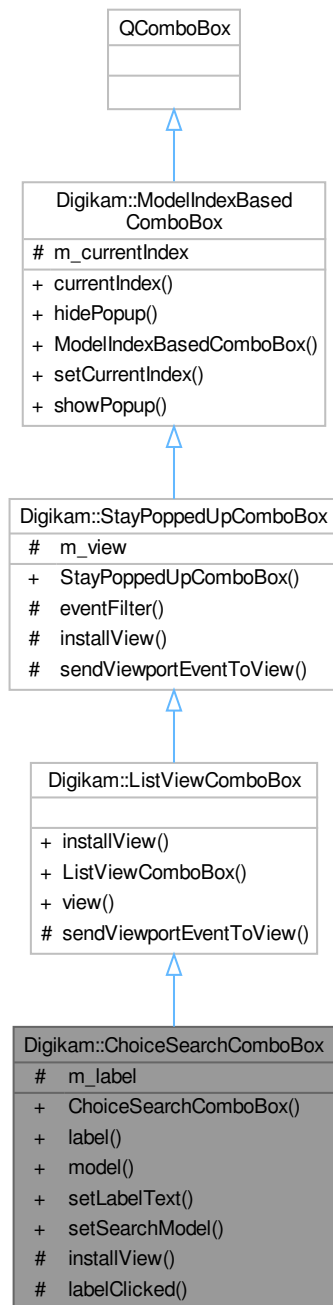
Reimplemented from [Digikam::AlbumFilterModel](#).

Reimplemented in [Digikam::SearchFilterModel](#), [Digikam::TagPropertiesFilterModel](#), and [Digikam::TagsManagerFilterModel](#).



## 6.186 Digikam::ChoiceSearchComboBox Class Reference

Inheritance diagram for Digikam::ChoiceSearchComboBox:



### Signals

- void **checkStateChanged** ()

### Public Member Functions

- [ChoiceSearchComboBox](#) (QWidget \*const parent=nullptr)
- [DSqueezedClickLabel](#) \* **label** () const
- [ChoiceSearchModel](#) \* **model** () const
- void [setLabelText](#) (const QString &text)
- void [setsearchModel](#) ([ChoiceSearchModel](#) \*model)

### Public Member Functions inherited from [Digikam::ListViewComboBox](#)

- [ListViewComboBox](#) (QWidget \*parent=nullptr)
- QListView \* [view](#) () const

### Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex **currentIndex** () const
- void **hidePopup** () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

### Protected Slots

- void **labelClicked** ()

### Protected Member Functions

- void [installView](#) (QAbstractItemView \*view=nullptr) override

### Protected Member Functions inherited from [Digikam::ListViewComboBox](#)

- void [sendViewportEventToView](#) (QEvent \*e) override

### Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- void [installView](#) (QAbstractItemView \*view)

### Protected Attributes

- [DSqueezedClickLabel](#) \* **m\_label** = nullptr

## Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

- `QAbstractItemView * m_view = nullptr`

## Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- `QPersistentModelIndex m_currentIndex`

## 6.186.1 Constructor & Destructor Documentation

### 6.186.1.1 ChoiceSearchComboBox()

```
Digikam::ChoiceSearchComboBox::ChoiceSearchComboBox (
    QWidget *const parent = nullptr ) [explicit]
```

A combo box for entering a choice of values. Operates on a [ChoiceSearchModel](#). After constructing the object, call `setModel` with your model.

## 6.186.2 Member Function Documentation

### 6.186.2.1 installView()

```
void Digikam::ChoiceSearchComboBox::installView (
    QAbstractItemView * view = nullptr ) [override], [protected], [virtual]
```

Replace the standard combo box list view with a `QTreeView`. Call this after installing an appropriate model.

Reimplemented from [Digikam::ListViewComboBox](#).

### 6.186.2.2 setLabelText()

```
void Digikam::ChoiceSearchComboBox::setLabelText (
    const QString & text )
```

Updates the text on the line edit area.

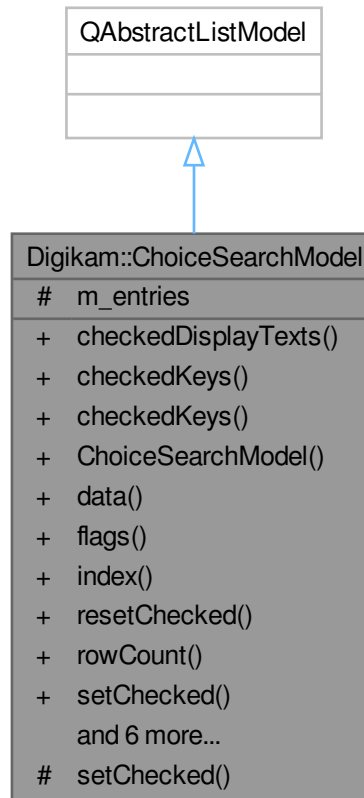
### 6.186.2.3 setSearchModel()

```
void Digikam::ChoiceSearchComboBox::setSearchModel (
    ChoiceSearchModel * model )
```

Sets the model and initializes the widget. Can only be called once for a widget.

## 6.187 Digikam::ChoiceSearchModel Class Reference

Inheritance diagram for Digikam::ChoiceSearchModel:



### Classes

- class [Entry](#)

### Public Types

- enum `CustomRoles` { `IdRole = Qt::UserRole` }

### Signals

- void `checkStateChanged` (const `QVariant` &key, bool isChecked)

**Public Member Functions**

- QStringList [checkedDisplayTexts](#) () const
- QVariantList [checkedKeys](#) () const
- template<typename T >  
QList< T > [checkedKeys](#) () const
- **ChoicesearchModel** (QObject \*const parent=nullptr)
- QVariant **data** (const QModelIndex &index, int role) const override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- QModelIndex **index** (int row, int column=0, const QModelIndex &parent=QModelIndex()) const override
- void [resetChecked](#) ()
- int **rowCount** (const QModelIndex &parent) const override
- template<typename T >  
void [setChecked](#) (const QList< T > &keys, bool checked=true)
- template<typename T >  
void [setChecked](#) (const T &key, bool checked=true)
- template<typename T >  
void [setChecked](#) (const T &value, SearchXml::Relation relation)
- void [setChoice](#) (const QMap< int, QString > &data)
- void [setChoice](#) (const QStringList &data)
- void [setChoice](#) (const QVariantList &data)
- bool **setData** (const QModelIndex &index, const QVariant &value, int role) override

**Protected Member Functions**

- void **setChecked** (int index, bool checked)

**Protected Attributes**

- QList< [Entry](#) > **m\_entries**

**6.187.1 Member Function Documentation****6.187.1.1 checkedDisplayTexts()**

```
QStringList Digikam::ChoicesearchModel::checkedDisplayTexts ( ) const
```

Returns the display text of all entries that are selected.

**6.187.1.2 checkedKeys() [1/2]**

```
QVariantList Digikam::ChoicesearchModel::checkedKeys ( ) const
```

Returns the keys of all entries that are selected (checked).

**6.187.1.3 checkedKeys() [2/2]**

```
template<typename T >  
QList< T > Digikam::ChoicesearchModel::checkedKeys ( ) const
```

Returns the keys of all entries that are selected (checked), converted to a list of the template type. Supported for Int and QString types.

#### 6.187.1.4 resetChecked()

```
void Digikam::ChoiceSearchModel::resetChecked ( )
```

Sets all entries to unchecked.

#### 6.187.1.5 setChecked() [1/3]

```
template<typename T >
void Digikam::ChoiceSearchModel::setChecked (
    const QList< T > & keys,
    bool checked = true )
```

Sets the check state of all the entries whose key is found in the list to checked.

#### 6.187.1.6 setChecked() [2/3]

```
template<typename T >
void Digikam::ChoiceSearchModel::setChecked (
    const T & key,
    bool checked = true )
```

Sets the check state of the entry with given key.

#### 6.187.1.7 setChecked() [3/3]

```
template<typename T >
void Digikam::ChoiceSearchModel::setChecked (
    const T & value,
    SearchXml::Relation relation )
```

Sets the check state of all entries. The check state is determined by the key of an entry, the relation, and a constant value. Think of "Set to checked if key is less than 5". Supported for Int and QString types.

#### 6.187.1.8 setChoice() [1/3]

```
void Digikam::ChoiceSearchModel::setChoice (
    const QMap< int, QString > & data )
```

Sets the data from the given map, with integer keys and QString user displayable value.

#### 6.187.1.9 setChoice() [2/3]

```
void Digikam::ChoiceSearchModel::setChoice (
    const QStringList & data )
```

Sets the data from the given list, taking every first entry as the key, every second as the user displayable value.

### 6.187.1.10 setChoice() [3/3]

```
void Digikam::ChoiceSearchModel::setChoice (
    const QVariantList & data )
```

Sets the data from the given list, taking every first entry as the key, every second as the user displayable value. Ensure that the QVariants' type is correct (identical for all even entries, QString for all odd entries).

## 6.188 Digikam::ChoiceSearchModel::Entry Class Reference

### Public Member Functions

- **Entry** (const QVariant &key, const QString &userDisplay)
- bool **operator==** (const [Entry](#) &other) const
- bool **operator==** (const QVariant &other) const

### Public Attributes

- bool **m\_checkState** = false
- QString **m\_display**
- QVariant **m\_key**

## 6.188.1 Member Function Documentation

### 6.188.1.1 operator==( )

```
bool Digikam::ChoiceSearchModel::Entry::operator== (
    const QVariant & other ) const
```

## 6.189 Digikam::CIETongueWidget Class Reference

Inheritance diagram for Digikam::CIETongueWidget:



### Public Member Functions

- **CIETongueWidget** (int w, int h, QWidget \*const parent=nullptr, cmsHPROFILE hMonitor=nullptr)
- void **loadingFailed** ()
- void **loadingStarted** ()
- bool **setProfileData** (const QByteArray &profileData=QByteArray())
- bool **setProfileFromFile** (const QUrl &file=QUrl())
- void **uncalibratedColor** ()

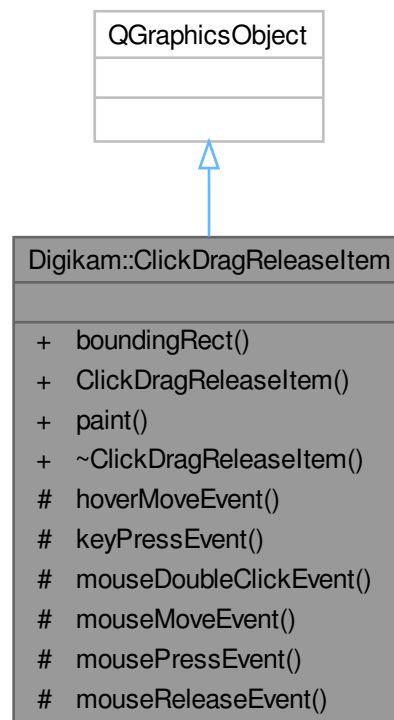


### Protected Member Functions

- QRgb **colorByCoord** (double x, double y)
- void **drawLabels** ()
- void **drawSmallEllipse** (LPcmsCIExyY xyY, BYTE r, BYTE g, BYTE b, int sz)
- void **drawTongueAxis** ()
- void **drawTongueGrid** ()
- void **fillTongue** ()
- int **grids** (double val) const
- void **outlineTongue** ()
- void **paintEvent** (QPaintEvent \*) override
- void **resizeEvent** (QResizeEvent \*event) override

## 6.190 Digikam::ClickDragReleaseltem Class Reference

Inheritance diagram for Digikam::ClickDragReleaseltem:



### Signals

- void **cancelled** ()
- void **finished** (const QRectF &rect)
- void **moving** (const QRectF &rect)
- void **started** (const QPointF &pos)

## Public Member Functions

- QRectF **boundingRect** () const override
- **ClickDragReleaseItem** (QGraphicsItem \*const parent)
- void **paint** (QPainter \*, const QStyleOptionGraphicsItem \*, QWidget \*) override

## Protected Member Functions

- void **hoverMoveEvent** (QGraphicsSceneHoverEvent \*) override
- void **keyPressEvent** (QKeyEvent \*) override
- void **mouseDoubleClickEvent** (QGraphicsSceneMouseEvent \*) override
- void **mouseMoveEvent** (QGraphicsSceneMouseEvent \*) override
- void **mousePressEvent** (QGraphicsSceneMouseEvent \*) override
- void **mouseReleaseEvent** (QGraphicsSceneMouseEvent \*) override

## 6.190.1 Member Function Documentation

### 6.190.1.1 mousePressEvent()

```
void Digikam::ClickDragReleaseItem::mousePressEvent (
    QGraphicsSceneMouseEvent * e ) [override], [protected]
```

1) Press - Drag - Release: mousePress, PressedState -> mouseMoveEvent over threshold, PressDragState -> mouseReleaseEvent, finished  
 2) Click - Move - Click: mousePressEvent, PressedState -> mouseReleaseEvent, ClickedMoveState -> hoverMoveEvent -> mouseReleaseEvent, finished

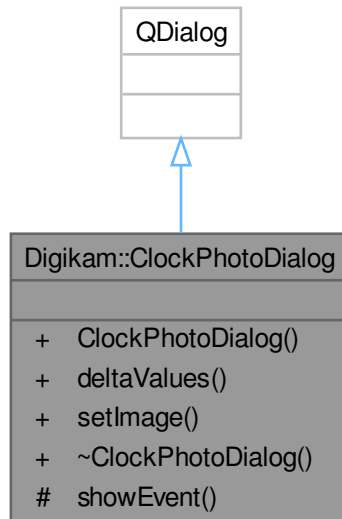
### 6.190.1.2 started

```
void Digikam::ClickDragReleaseItem::started (
    const QPointF & pos ) [signal]
```

Signals are emitted at click, drag and release event. Reported positions are in scene coordinates. A drag is reported only if the mouse was moved a certain threshold. A release is reported after every press.

## 6.191 Digikam::ClockPhotoDialog Class Reference

Inheritance diagram for Digikam::ClockPhotoDialog:



### Public Member Functions

- **ClockPhotoDialog** (`QWidget *const parent, const QUrl &defaultUrl`)
- **DeltaTime deltaValues** () const
- bool **setImage** (`const QUrl &`)

### Protected Member Functions

- void **showEvent** (`QShowEvent *`) override

### 6.191.1 Member Function Documentation

#### 6.191.1.1 setImage()

```
bool Digikam::ClockPhotoDialog::setImage (
    const QUrl & imageFile )
```

Try to load the photo specified by the QUrl, and set the datetime widget to the photo time. Return true on success, or false if either the photo can't be read or the datetime information can't be read.

## 6.192 Digikam::CMat Struct Reference

### Public Attributes

- double \* [center](#)
- double \* [data](#)
- int [radius](#)
- int [row\\_stride](#)

### 6.192.1 Detailed Description

[CMat](#):

Centered matrix. This is a square matrix where the indices range from  $[-radius, radius]$ . The matrix contains  $(2 * radius + 1) ** 2$  elements.

### 6.192.2 Member Data Documentation

#### 6.192.2.1 center

```
double* Digikam::CMat::center
```

Points to element with index (0, 0)

#### 6.192.2.2 data

```
double* Digikam::CMat::data
```

Contents of matrix

#### 6.192.2.3 radius

```
int Digikam::CMat::radius
```

Radius of the matrix.

#### 6.192.2.4 row\_stride

```
int Digikam::CMat::row_stride
```

Size of one row =  $2 * radius + 1$

## 6.193 Digikam::CollectionImageChangeset Class Reference

### Public Types

- enum [Operation](#) {  
**Unknown** , [Added](#) , [Removed](#) , [RemovedAll](#) ,  
[Deleted](#) , [RemovedDeleted](#) , [Moved](#) , [Copied](#) }

### Public Member Functions

- QList< int > **albums** () const
- [CollectionImageChangeset](#) ()=default
- CollectionImageChangeset** (const QList< qlonglong > &[ids](#), const QList< int > &[albums](#), [Operation](#) operation)
- CollectionImageChangeset** (const QList< qlonglong > &[ids](#), int album, [Operation](#) operation)
- CollectionImageChangeset** (qlonglong id, int album, [Operation](#) operation)
- bool **containsAlbum** (int id) const
- bool **containsImage** (qlonglong id) const
- QList< qlonglong > **ids** () const
- [Operation](#) **operation** () const
- [CollectionImageChangeset](#) & **operator<<** (const [CollectionImageChangeset](#) &other)

### 6.193.1 Member Enumeration Documentation

#### 6.193.1.1 Operation

enum [Digikam::CollectionImageChangeset::Operation](#)

#### Enumerator

Added	"Added" indicates that images have been added to albums.
Removed	"Removed" indicates that an image has been removed from the given album, and has possibly set a status of Removed and a null <a href="#">Album</a> (though this can already have changed to valid values), but the image-specific tables have not been removed.
RemovedAll	"RemovedAll" indicates that for all entries in the specified album, the "Removed" operation has been carried out. This is equivalent to a "Removed" changesets with all image ids in the list, but for RemovedAll, the list may not be explicitly given (may be empty).
Deleted	"Deleted" indicates that the image-specific tables have been removed from the database. While "Removed" means all data is still there, though possibly not accessible from an album, this means all data has been irreversibly deleted.
RemovedDeleted	Special combination: Images which has the "Removed" status have now been "Deleted". A changeset with Removed or RemovedAll is guaranteed to have been sent anytime before. Image ids nor albums ids may or may be not available in any combination.
Moved	Images have been moved. This is extra information; a Removed and then an Added changeset are guaranteed to be sent subsequently. <a href="#">Album</a> is the source album.
Copied	Images have been copied. This is extra information; an Added changeset is guaranteed to be sent subsequently. <a href="#">Album</a> is the source album.

## 6.193.2 Constructor & Destructor Documentation

### 6.193.2.1 CollectionImageChangeset()

```
Digikam::CollectionImageChangeset::CollectionImageChangeset ( ) [default]
```

An [CollectionImageChangeset](#) covers adding and removing an image to/from the collection. It is described by a list of affected image ids, a list of affected albums, and an operation. Special Case "RemovedAll": If all images have been removed from an album, operation is RemovedAll, the album list contains the (now empty) albums, [ids\(\)](#) is empty, but [containsImage\(\)](#) always returns true. Special Case "RemovedDeleted": Images with the "Removed" status are now irreversibly deleted. [ids\(\)](#) and/or [albums\(\)](#) may be empty (this means information is not available).

## 6.193.3 Member Function Documentation

### 6.193.3.1 ids()

```
QList< qlonglong > Digikam::CollectionImageChangeset::ids ( ) const
```

Specification of this changeset. All special cases where the returned list may be empty are noted above. The lists are valid unless such a case is explicitly mentioned.

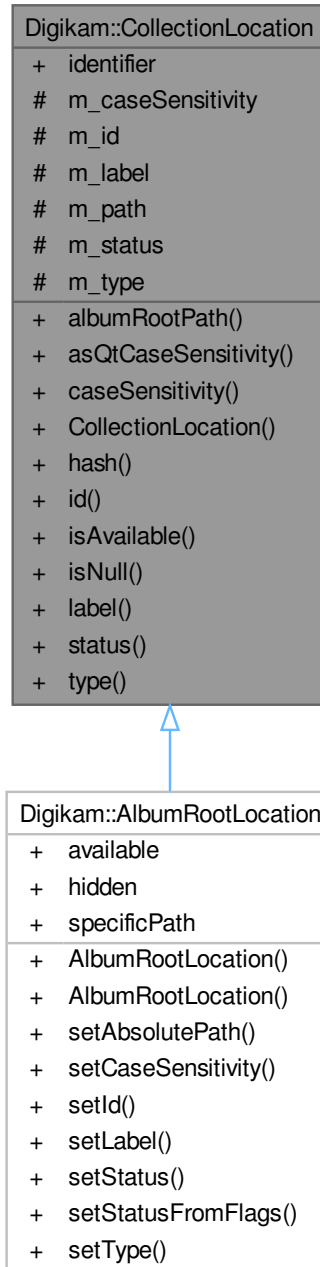
### 6.193.3.2 operator<<()

```
CollectionImageChangeset & Digikam::CollectionImageChangeset::operator<< (
    const CollectionImageChangeset & other )
```

Combines two CollectionImageChangesets. The operations shall not differ between the two sets; the operation is set to Unknown if it differs. This is especially not suitable for RemovedAll changesets.

## 6.194 Digikam::CollectionLocation Class Reference

Inheritance diagram for Digikam::CollectionLocation:



### Public Types

- enum `CaseSensitivity` { `UnknownCaseSensitivity`, `CaseInsensitive`, `CaseSensitive` }

- enum `Status` { `LocationNull` , `LocationAvailable` , `LocationHidden` , `LocationUnavailable` , `LocationDeleted` }
- enum `Type` { `Undefined` = 0 , `VolumeHardWired` = 1 , `VolumeRemovable` = 2 , `Network` = 3 }

### Public Member Functions

- `QString albumRootPath () const`
- `Qt::CaseSensitivity asQtCaseSensitivity () const`
- `CaseSensitivity caseSensitivity () const`
- `size_t hash () const`
- `int id () const`
- `bool isAvailable () const`
- `bool isNull () const`
- `QString label () const`
- `Status status () const`
- `Type type () const`

### Public Attributes

- `QString identifier`

### Protected Attributes

- `CaseSensitivity m_caseSensitivity = UnknownCaseSensitivity`
- `int m_id = -1`
- `QString m_label`
- `QString m_path`
- `Status m_status = LocationNull`
- `Type m_type = VolumeHardWired`

## 6.194.1 Member Enumeration Documentation

### 6.194.1.1 CaseSensitivity

```
enum Digikam::CollectionLocation::CaseSensitivity
```

#### Enumerator

<code>UnknownCaseSensitivity</code>	The location has an unknown case sensitivity.
<code>CaseInsensitive</code>	The location is case insensitive.
<code>CaseSensitive</code>	The location is case sensitive.

### 6.194.1.2 Status

```
enum Digikam::CollectionLocation::Status
```



## Enumerator

LocationNull	An invalid status. A location has this status if it is not valid, and it had this status before its creation (for oldStatus information)
LocationAvailable	The location if available. This is the most common status.
LocationHidden	The location is explicitly hidden. This gives no information if the location was available were it not hidden.
LocationUnavailable	The location is currently not available. (Harddisk unplugged, CD not in drive, network fs not mounted etc.) It may become available any time.
LocationDeleted	An invalid status. A location object acquires this status if it has been deleted. The object then does no longer point to an existing location.

## 6.194.1.3 Type

```
enum Digikam::CollectionLocation::Type
```

## Enumerator

Undefined	The location is undefined. Keep values constant.
VolumeHardWired	The location is located on a storage device that is built-in without frequent removal: Hard-disk inside the machine.
VolumeRemovable	The location is located on a storage device that can be removed from the local machine, and is expected to be removed. USB stick, USB hard-disk, CD, DVD
Network	The location is available via a network file system. The availability depends on the network connection.

## 6.194.2 Member Function Documentation

## 6.194.2.1 albumRootPath()

```
QString Digikam::CollectionLocation::albumRootPath ( ) const
```

The current file system path leading to this album root. Only guaranteed to be valid for location with status Available.

## 6.194.2.2 asQtCaseSensitivity()

```
Qt::CaseSensitivity Digikam::CollectionLocation::asQtCaseSensitivity ( ) const
```

Return as Qt case sensitivity enum of location. For unknown, it is assumed to be Qt::CaseSensitive.

## 6.194.2.3 caseSensitivity()

```
CollectionLocation::CaseSensitivity Digikam::CollectionLocation::caseSensitivity ( ) const
```

The case sensitivity of location. See above for possible values.

**6.194.2.4 id()**

```
int Digikam::CollectionLocation::id ( ) const
```

The id uniquely identifying this collection

**6.194.2.5 label()**

```
QString Digikam::CollectionLocation::label ( ) const
```

A user-visible, optional label.

**6.194.2.6 status()**

```
CollectionLocation::Status Digikam::CollectionLocation::status ( ) const
```

The current status. See above for possible values.

**6.194.2.7 type()**

```
CollectionLocation::Type Digikam::CollectionLocation::type ( ) const
```

The type of location. See above for possible values.

## 6.195 Digikam::CollectionManager Class Reference

Inheritance diagram for Digikam::CollectionManager:



### Classes

- class [Private](#)

## Public Types

- enum [LocationCheckResult](#) { [LocationInvalidCheck](#) , [LocationAllRight](#) , [LocationHasProblems](#) , [LocationNotAllowed](#) }

## Signals

- void [triggerUpdateVolumesList](#) ()

## Public Member Functions

- void [refresh](#) ()
- void [setWatchDisabled](#) ()

## Static Public Member Functions

- static void [cleanUp](#) ()
- static [CollectionManager](#) \* [instance](#) ()

## Operations on Albums

- class **Private**
- class **CoreDbWatch**
- class **CoreDbAccess**
- QStringList [allAvailableAlbumRootPaths](#) ()
- QString [albumRootPath](#) (int id)
- QString [albumRootLabel](#) (int id)
- QUrl [albumRoot](#) (const QUrl &fileUrl)
- QString [albumRootPath](#) (const QUrl &fileUrl)
- QString [albumRootPath](#) (const QString &filePath)
- bool [isAlbumRoot](#) (const QUrl &fileUrl)
- bool [isAlbumRoot](#) (const QString &filePath)
- QString [album](#) (const QUrl &fileUrl)
- QString [album](#) (const QString &filePath)
- QString [album](#) (const [CollectionLocation](#) &location, const QUrl &fileUrl)
- QString [album](#) (const [CollectionLocation](#) &location, const QString &filePath)
- QUrl [oneAlbumRoot](#) ()
- QString [oneAlbumRootPath](#) ()

## Operations on Collection Location

- [CollectionLocation addLocation](#) (const [QUrl](#) &fileUrl, const [QString](#) &label=QString())
- [CollectionLocation addNetworkLocation](#) (const [QUrl](#) &fileUrl, const [QString](#) &label=QString())
- [CollectionLocation refreshLocation](#) (const [CollectionLocation](#) &location, int newType, const [QStringList](#) &pathList, const [QString](#) &label=QString())
- [LocationCheckResult checkLocation](#) (const [QUrl](#) &fileUrl, [QList](#)< [CollectionLocation](#) > &assumeDeleted, [QString](#) \*message=nullptr, [QString](#) \*suggestedMessageIconName=nullptr)
- [LocationCheckResult checkNetworkLocation](#) (const [QUrl](#) &fileUrl, [QList](#)< [CollectionLocation](#) > &assumeDeleted, [QString](#) \*message=nullptr, [QString](#) \*suggestedMessageIconName=nullptr)
- void [removeLocation](#) (const [CollectionLocation](#) &location)
- void [setLabel](#) (const [CollectionLocation](#) &location, const [QString](#) &label)
- void [changeType](#) (const [CollectionLocation](#) &location, int type)
- [QList](#)< [CollectionLocation](#) > [checkHardWiredLocations](#) ()
- void [migrationCandidates](#) (const [CollectionLocation](#) &disappearedLocation, [QString](#) \*const technicalDescription, [QStringList](#) \*const candidateIdentifiers, [QStringList](#) \*const candidateDescriptions)
- void [migrateToVolume](#) (const [CollectionLocation](#) &location, const [QString](#) &identifier)
- [QList](#)< [CollectionLocation](#) > [allLocations](#) ()
- [QList](#)< [CollectionLocation](#) > [allAvailableLocations](#) ()
- [CollectionLocation](#) [locationForAlbumRootId](#) (int id)
- [CollectionLocation](#) [locationForAlbumRoot](#) (const [QUrl](#) &fileUrl)
- [CollectionLocation](#) [locationForAlbumRootPath](#) (const [QString](#) &albumRootPath)
- [CollectionLocation](#) [locationForUrl](#) (const [QUrl](#) &fileUrl)
- [CollectionLocation](#) [locationForPath](#) (const [QString](#) &filePath)
- void [locationStatusChanged](#) (const [CollectionLocation](#) &location, int oldStatus)
- void [locationPropertiesChanged](#) (const [CollectionLocation](#) &location)

## 6.195.1 Member Enumeration Documentation

### 6.195.1.1 LocationCheckResult

```
enum Digikam::CollectionManager::LocationCheckResult
```

#### Enumerator

LocationInvalidCheck	The check did not succeed, status unknown.
LocationAllRight	All right. The accompanying message may be empty.
LocationHasProblems	Location can be added, but the user should be aware of a problem.
LocationNotAllowed	Adding the location will fail (e.g. there is already a location for the path)

## 6.195.2 Member Function Documentation

### 6.195.2.1 addLocation()

```
CollectionLocation Digikam::CollectionManager::addLocation (
    const QUrl & fileUrl,
    const QString & label = QString() )
```

Add the given file system location as new collection location. Type and availability will be detected. On failure returns null. This would be the case if the given url is already contained in another collection location. You may

pass an optional user-visible label that will be stored in the database. The label has no further meaning and can be freely chosen.

[CollectionLocation](#) objects returned are simple data containers. If the corresponding location is returned, the data is still safe to access, but does not represent anything. Therefore, do not store returned objects, but prefer to retrieve them freshly.

#### 6.195.2.2 album()

```
QString Digikam::CollectionManager::album (
    const QUrl & fileUrl )
```

Returns the album part of the given file path, i.e. the album root path at the beginning is removed and the second part, starting with "/", ending without a slash, is returned. Example: "/media/fotos/Paris 2007" gives "/Paris 2007" Returns a null QString if the file path is not located in an album root. Returns "/" if the file path is an album root. Note that trailing slashes are removed in the return value, regardless if there was one or not. Note that you have to feed a path/url pointing to a directory. File names cannot be recognized as such by this method, and will be treated as a directory.

#### 6.195.2.3 albumRoot()

```
QUrl Digikam::CollectionManager::albumRoot (
    const QUrl & fileUrl )
```

For a given path, the part of the path that forms the album root is returned, ending without a slash. Example: "/media/fotos/Paris 2007" gives "/media/fotos". Only available (or hidden, but available) album roots are guaranteed to be found.

#### 6.195.2.4 albumRootLabel()

```
QString Digikam::CollectionManager::albumRootLabel (
    int id )
```

Returns the album root label with the given id. Returns a null QString if the root path does not exist or is not available.

#### 6.195.2.5 albumRootPath()

```
QString Digikam::CollectionManager::albumRootPath (
    int id )
```

Returns the album root path with the given id. Returns a null QString if the root path does not exist or is not available.

#### 6.195.2.6 allAvailableAlbumRootPaths()

```
QStringList Digikam::CollectionManager::allAvailableAlbumRootPaths ( )
```

Returns a list of the paths of all currently available root paths

### 6.195.2.7 allAvailableLocations()

```
QList< CollectionLocation > Digikam::CollectionManager::allAvailableLocations ( )
```

Returns a list of all currently available CollectionLocations

### 6.195.2.8 allLocations()

```
QList< CollectionLocation > Digikam::CollectionManager::allLocations ( )
```

Returns a list of all CollectionLocations stored in the database

### 6.195.2.9 changeType()

```
void Digikam::CollectionManager::changeType (
    const CollectionLocation & location,
    int type )
```

Changes the [CollectionLocation::Type](#) of the given location

### 6.195.2.10 checkHardWiredLocations()

```
QList< CollectionLocation > Digikam::CollectionManager::checkHardWiredLocations ( )
```

Checks the locations of type HardWired. If one of these is not available currently, it is added to the list of disappeared locations. This case may happen if a file system is changed, a backup restored or other actions taken that change the UUID, although the data may still be available and mounted. If there are hard-wired volumes available which are candidates for a newly appeared volume (in fact those that do not contain any collections currently), they are added to the map, identifier -> i18n'ed user presentable description. The identifier can be used for changeVolume.

### 6.195.2.11 checkLocation()

```
CollectionManager::LocationCheckResult Digikam::CollectionManager::checkLocation (
    const QUrl & filePath,
    QList< CollectionLocation > & assumeDeleted,
    QString * message = nullptr,
    QString * suggestedMessageIconName = nullptr )
```

Analyzes the given file path. Creates an info message describing the result of identification or possible problems. The text is i18n'ed and can be presented to the user. The returned result enum describes the test result.

### 6.195.2.12 isAlbumRoot() [1/2]

```
bool Digikam::CollectionManager::isAlbumRoot (
    const QString & filePath )
```

The file path should not end with the directory slash. Using [CoreDbUrl](#)'s method is fine.

**6.195.2.13 isAlbumRoot()** [2/2]

```
bool Digikam::CollectionManager::isAlbumRoot (
    const QUrl & fileUrl )
```

Returns true if the given path forms an album root. It will return false if the path is a path below an album root, or if the path does not belong to an album root. Example: `"/media/fotos/Paris 2007"` is an album with album root `"/media/fotos"`. `"/media/fotos"` returns true, `"/media/fotos/Paris 2007"` and `"/media"` return false. Only available (or hidden, but available) album roots are guaranteed to be found.

**6.195.2.14 locationForAlbumRoot()**

```
CollectionLocation Digikam::CollectionManager::locationForAlbumRoot (
    const QUrl & fileUrl )
```

Returns the [CollectionLocation](#) that contains the given album root. The path must be an album root with [isAlbumRoot\(\)](#) == true. Returns 0 if no collection location matches. Only available (or hidden, but available) locations are guaranteed to be found.

**6.195.2.15 locationForAlbumRootId()**

```
CollectionLocation Digikam::CollectionManager::locationForAlbumRootId (
    int id )
```

Returns the location for the given album root id

**6.195.2.16 locationForUrl()**

```
CollectionLocation Digikam::CollectionManager::locationForUrl (
    const QUrl & fileUrl )
```

Returns the [CollectionLocation](#) that contains the given path. Equivalent to calling `locationForAlbumRoot(albumRoot(fileUrl))`. Only available (or hidden, but available) locations are guaranteed to be found.

**6.195.2.17 locationPropertiesChanged**

```
void Digikam::CollectionManager::locationPropertiesChanged (
    const CollectionLocation & location ) [signal]
```

Emitted when the label of a collection location is changed

**6.195.2.18 locationStatusChanged**

```
void Digikam::CollectionManager::locationStatusChanged (
    const CollectionLocation & location,
    int oldStatus ) [signal]
```

Emitted when the status of a collection location changed. This means that the location became available, hidden or unavailable.

An added location will change its status after addition, from Null to Available, Hidden or Unavailable.

A removed location will change its status to Deleted during the removal; in this case, you shall not use the object passed with this signal with any method of [CollectionManager](#).

The second signal argument is of type [CollectionLocation::Status](#) and describes the status before the state change occurred



### 6.195.2.19 migrateToVolume()

```
void Digikam::CollectionManager::migrateToVolume (
    const CollectionLocation & location,
    const QString & identifier )
```

Migrates the existing collection to a new volume, identified by an internal identifier as returned by [checkHardWiredLocations\(\)](#). Use this *only* to react to changes like those detailed for [checkHardWiredLocations](#); the actual data pointed to shall be unchanged.

### 6.195.2.20 migrationCandidates()

```
void Digikam::CollectionManager::migrationCandidates (
    const CollectionLocation & disappearedLocation,
    QString *const technicalDescription,
    QStringList *const candidateIdentifiers,
    QStringList *const candidateDescriptions )
```

For a given disappeared location (retrieved from [checkHardWiredLocations\(\)](#)) retrieve a user-presentable technical description (excluding the [CollectionLocation](#)'s label) and a list of identifiers and corresponding user presentable strings of candidates to where the given location may have been moved.

### 6.195.2.21 oneAlbumRoot()

```
QUrl Digikam::CollectionManager::oneAlbumRoot ( )
```

Returns just one album root, out of the list of available location, the one that is most suitable to serve as a default, e.g. to suggest as default place when the user wants to add files.

### 6.195.2.22 refresh()

```
void Digikam::CollectionManager::refresh ( )
```

Clears all locations and re-reads the lists of collection locations. Enables the watch.

### 6.195.2.23 removeLocation()

```
void Digikam::CollectionManager::removeLocation (
    const CollectionLocation & location )
```

Removes the given location. This means that all images contained on the location will be removed from the database, all tags will be lost.

### 6.195.2.24 setLabel()

```
void Digikam::CollectionManager::setLabel (
    const CollectionLocation & location,
    const QString & label )
```

Sets the label of the given location

### 6.195.2.25 setWatchDisabled()

```
void Digikam::CollectionManager::setWatchDisabled ( )
```

Disables the collection watch. It will be reenabled as soon as [refresh\(\)](#) is called or any other action triggered.

## 6.196 Digikam::CollectionManager::Private Class Reference

### Public Member Functions

- `QList< SolidVolumeInfo > actuallyListVolumes ( )`  
*hack for Solid's threading problems*
- `bool checkCollectionUUID (AlbumRootLocation *const location, const QString &path)`  
*Check or create the file UUID in the collection root if possible.*
- `bool checkIfExists (const QString &path, QList< CollectionLocation > assumeDeleted)`  
*Check if a location for specified path exists, assuming the given list of locations was deleted.*
- `SolidVolumeInfo findVolumeForLocation (const AlbumRootLocation *location, const QList< SolidVolumeInfo > &volumes)`
- `SolidVolumeInfo findVolumeForUrl (const QUrl &fileUrl, const QList< SolidVolumeInfo > &volumes)`
- `QString getCollectionUUID (const QString &path)`  
*Get the file UUID in the collection root if possible.*
- `QList< SolidVolumeInfo > listVolumes ( )`  
*Access Solid and return a list of storage volumes.*
- `QString networkShareIdentifier (const QStringList &paths)`  
*Create a network share identifier based on the mountpaths.*
- `QStringList networkShareMountPathsFromIdentifier (const AlbumRootLocation *location)`  
*Return the path, if location has a path-only identifier. Else returns a null string.*
- `QString pathFromIdentifier (const AlbumRootLocation *location)`  
*Return the path, if location has a path-only identifier. Else returns a null string.*
- `Private (CollectionManager *const ss)`
- `void slotTriggerUpdateVolumesList ( )`
- `QString technicalDescription (const AlbumRootLocation *location)`  
*Make a user presentable description, regardless of current location status.*
- `QString volumeIdentifier (const QString &path)`  
*Create a volume identifier based on the path only.*
- `QString volumeIdentifier (const SolidVolumeInfo &info)`  
*Create the volume identifier for the given volume info.*

### Static Public Member Functions

- `static QString directoryHash (const QString &path)`  
*Create an MD5 hash of the top-level entries (file names, not file content) of the given path.*

### Public Attributes

- `bool changingDB = false`
- `QMap< int, AlbumRootLocation * > locations`
- `QReadWriteLock lock`
- `CollectionManager * s = nullptr`
- `QStringList udisToWatch`
- `QList< SolidVolumeInfo > volumesListCache`
- `bool watchEnabled = false`

## 6.196.1 Member Function Documentation

### 6.196.1.1 findVolumeForLocation()

```
SolidVolumeInfo Digikam::CollectionManager::Private::findVolumeForLocation (
    const AlbumRootLocation * location,
    const QList< SolidVolumeInfo > & volumes )
```

Find from a given list (usually the result of listVolumes) the volume corresponding to the location

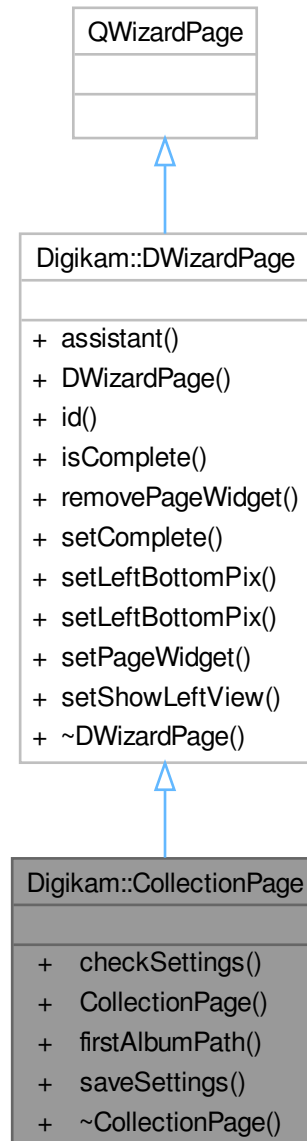
### 6.196.1.2 findVolumeForUrl()

```
SolidVolumeInfo Digikam::CollectionManager::Private::findVolumeForUrl (
    const QUrl & fileUrl,
    const QList< SolidVolumeInfo > & volumes )
```

Find from a given list (usually the result of listVolumes) the volume on which the file path specified by the url is located.

## 6.197 Digikam::CollectionPage Class Reference

Inheritance diagram for Digikam::CollectionPage:



### Public Member Functions

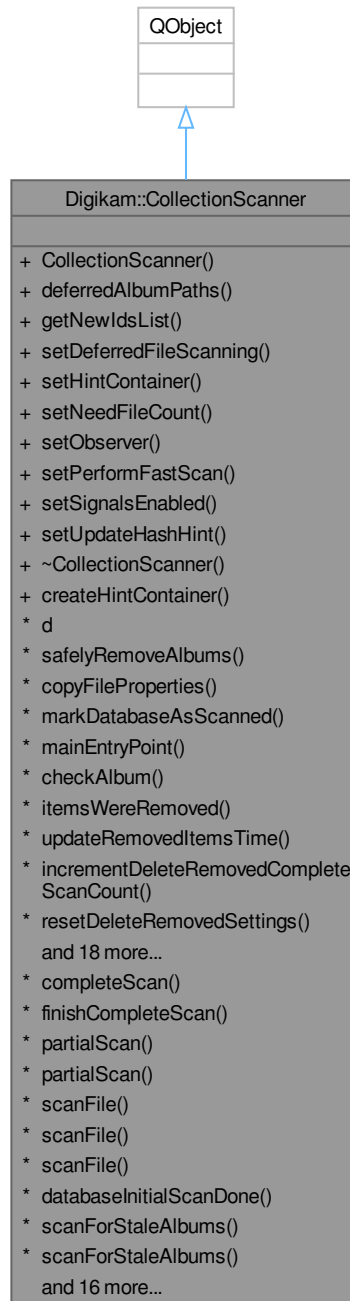
- bool **checkSettings** ()
- **CollectionPage** (QWizard \*const dlg)
- QString **firstAlbumPath** () const
- void **saveSettings** ()

## Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)
- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.198 Digikam::CollectionScanner Class Reference

Inheritance diagram for Digikam::CollectionScanner:



### Classes

- class [Private](#)

## Public Types

- enum [FileScanMode](#) { [NormalScan](#) , [ModifiedScan](#) , [Rescan](#) , [CleanScan](#) }

## Public Member Functions

- QStringList **deferredAlbumPaths** () const
- QList< qlonglong > [getNewIdsList](#) () const
- void **setDeferredFileScanning** (bool defer)
- void **setHintContainer** ([CollectionScannerHintContainer](#) \*const container)
- void [setNeedFileCount](#) (bool on)
- void [setObserver](#) ([CollectionScannerObserver](#) \*const observer)
- void [setPerformFastScan](#) (bool on)
- void [setSignalsEnabled](#) (bool on)
- void **setUpdateHashHint** (bool hint=true)

## Static Public Member Functions

- static [CollectionScannerHintContainer](#) \* [createHintContainer](#) ()

## Scan utilities

- void [safelyRemoveAlbums](#) (const QList< int > &albumIds)
- static void [copyFileProperties](#) (const [ItemInfo](#) &source, const [ItemInfo](#) &dest)
- void **markDatabaseAsScanned** ()
- void **mainEntryPoint** (bool complete)
- int **checkAlbum** (const [CollectionLocation](#) &location, const QString &album)
- void **itemsWereRemoved** (const QList< qlonglong > &removedIds)
- void **updateRemovedItemsTime** ()
- void **incrementDeleteRemovedCompleteScanCount** ()
- void **resetDeleteRemovedSettings** ()
- bool **checkDeleteRemoved** ()
- void **loadNameFilters** ()
- int **countItemsInFolder** (const QString &path)
- DatabasesItem::Category **category** (const QFileInfo &info)
- void [totalFilesToScan](#) (int count)
- void [startScanningAlbumRoot](#) (const QString &albumRoot)
- void **startScanningAlbum** (const QString &albumRoot, const QString &album)
- void **startScanningForStaleAlbums** ()
- void **startScanningAlbumRoots** ()
- void **startCompleteScan** ()
- void **signalScannedNewImage** (const QFileInfo &info)
- void [finishedScanningAlbumRoot](#) (const QString &albumRoot)
- void **finishedScanningAlbum** (const QString &albumRoot, const QString &album, int filesScanned)
- void **finishedScanningForStaleAlbums** ()
- void **finishedCompleteScan** ()
- void [scannedFiles](#) (int filesScanned)
- void [cancelled](#) ()

## Scan operations

- void `completeScan` ()
- void `finishCompleteScan` (const QStringList &albumPaths)
- void `partialScan` (const QString &filePath)
- void `partialScan` (const QString &albumRoot, const QString &album)
- qlonglong `scanFile` (const QString &filePath, [FileScanMode](#) mode=[ModifiedScan](#))
- qlonglong `scanFile` (const QString &albumRoot, const QString &album, const QString &fileName, [FileScanMode](#) mode=[ModifiedScan](#))
- void `scanFile` (const [ItemInfo](#) &info, [FileScanMode](#) mode=[ModifiedScan](#))
- static bool `databaseInitialScanDone` ()
- void `scanForStaleAlbums` (const QList< [CollectionLocation](#) > &locations)
- void `scanForStaleAlbums` (const QList< int > &locationIdsToScan)
- void `scanAlbumRoot` (const [CollectionLocation](#) &location)
- void `scanAlbum` (const [CollectionLocation](#) &location, const QString &album, bool checkDate=false)
- void `scanExistingFile` (const QFileInfo &fi, qlonglong id)
- void `scanFileNormal` (const QFileInfo &info, const [ItemScanInfo](#) &scanInfo, bool checkSidecar=true, const QFileInfo \*const sidecarInfo=nullptr)
- void `scanModifiedFile` (const QFileInfo &info, const [ItemScanInfo](#) &scanInfo)
- void `scanFileUpdateHashReuseThumbnail` (const QFileInfo &fi, const [ItemScanInfo](#) &scanInfo, bool file←WasEdited)
- void `cleanScanFile` (const QFileInfo &info, const [ItemScanInfo](#) &scanInfo)
- void `rescanFile` (const QFileInfo &info, const [ItemScanInfo](#) &scanInfo)
- void `completeScanCleanupPart` ()
- void `completeHistoryScanning` ()
- void `finishHistoryScanning` ()
- void `historyScanningStage2` (const QList< qlonglong > &ids)
- void `historyScanningStage3` (const QList< qlonglong > &ids)
- qlonglong `scanFile` (const QFileInfo &fi, int albumId, qlonglong id, [FileScanMode](#) mode)
- qlonglong `scanNewFile` (const QFileInfo &info, int albumId)
- qlonglong `scanNewFileFullScan` (const QFileInfo &info, int albumId)

## 6.198.1 Member Enumeration Documentation

### 6.198.1.1 FileScanMode

enum `Digikam::CollectionScanner::FileScanMode`

#### Enumerator

NormalScan	The file will be scanned like it is done for any usual scan. If it was not modified, no further action is taken. If the file is not known yet, it will be fully scanned, or, if an identical file is found, this data will be copied.
ModifiedScan	The file will scanned like a modified file. Only a selected portion of the metadata will be updated into the database. If the file is not known yet, it will be fully scanned, or, if an identical file is found, this data will be copied.
Rescan	The file will be scanned like a completely new file. The complete metadata is re-read into the database. No search for identical files will be done.
CleanScan	This is the same as Rescan but the database metadata will be cleaned up if the corresponding metadata write option is enabled.



## 6.198.2 Member Function Documentation

### 6.198.2.1 cancelled

```
void Digikam::CollectionScanner::cancelled ( ) [signal]
```

Emitted when the observer told to cancel the scan

### 6.198.2.2 completeScan()

```
void Digikam::CollectionScanner::completeScan ( )
```

Carries out a full scan on all available parts of the collection. Only a full scan can finally remove deleted files from the database, only a full scan will mark the database as scanned. The database will be locked while running (Note: this is not done for partialScans).

### 6.198.2.3 copyFileProperties()

```
void Digikam::CollectionScanner::copyFileProperties (
    const ItemInfo & source,
    const ItemInfo & dest ) [static]
```

When a file is derived from another file, typically through editing, copy all relevant attributes from source file to the new file.

### 6.198.2.4 createHintContainer()

```
CollectionScannerHintContainer * Digikam::CollectionScanner::createHintContainer ( ) [static]
```

Hints give the scanner additional info about things that happened in the past carried out by higher level which the collection scanner cannot know. They allow to carry out optimizations. Record hints in a container, and provide the container to the collection scanner. The Container set in setHintContainer must be one created by createContainer.

### 6.198.2.5 databaseInitialScanDone()

```
bool Digikam::CollectionScanner::databaseInitialScanDone ( ) [static]
```

Returns if the initial scan of the database has been done. This is the first complete scan after creation of a new database file (or update requiring a rescan)

### 6.198.2.6 finishCompleteScan()

```
void Digikam::CollectionScanner::finishCompleteScan (
    const QStringList & albumPaths )
```

If you enable deferred file scanning for a [completeScan\(\)](#), new files will not be scanned. The relevant albums are available from [deferredAlbumPaths\(\)](#) when [completeScan\(\)](#) has finished. You need to call [finishCompleteScan\(\)](#) afterwards with the list to get the same complete scan than undeferred [completeScan\(\)](#).

### 6.198.2.7 finishedScanningAlbumRoot

```
void Digikam::CollectionScanner::finishedScanningAlbumRoot (
    const QString & albumRoot ) [signal]
```

Emitted when the scanning has finished. Note that start/finishScanningAlbum may be emitted recursively.

### 6.198.2.8 getNewIdsList()

```
QList< qlonglong > Digikam::CollectionScanner::getNewIdsList ( ) const
```

Returns item ids from new detected items

### 6.198.2.9 partialScan() [1/2]

```
void Digikam::CollectionScanner::partialScan (
    const QString & albumRoot,
    const QString & album )
```

Same procedure as above, but albumRoot and album is provided.

### 6.198.2.10 partialScan() [2/2]

```
void Digikam::CollectionScanner::partialScan (
    const QString & filePath )
```

Carries out a partial scan on the specified path of the collection. The includes scanning for new files + albums and updating modified file data. Files no longer found in the specified path however are not completely removed, but only marked as removed. They will be removed only after a complete scan.

### 6.198.2.11 safelyRemoveAlbums()

```
void Digikam::CollectionScanner::safelyRemoveAlbums (
    const QList< int > & albumIds )
```

Prepare the given albums to be removed, typically by setting the albums as orphan and removing all entries from the albums

### 6.198.2.12 scanFile() [1/3]

```
void Digikam::CollectionScanner::scanFile (
    const ItemInfo & info,
    FileScanMode mode = ModifiedScan )
```

The given file represented by the [ItemInfo](#) will be scanned according to mode

**6.198.2.13 scanFile() [2/3]**

```
qulonglong Digikam::CollectionScanner::scanFile (
    const QString & albumRoot,
    const QString & album,
    const QString & fileName,
    FileScanMode mode = ModifiedScan )
```

Same procedure as above, but albumRoot and album is provided. If you already have this info it need not be retrieved. Returns the image id of the file, or -1 on failure.

**6.198.2.14 scanFile() [3/3]**

```
qulonglong Digikam::CollectionScanner::scanFile (
    const QString & filePath,
    FileScanMode mode = ModifiedScan )
```

The given file will be scanned according to the given mode. Returns the image id of the file.

**6.198.2.15 scannedFiles**

```
void Digikam::CollectionScanner::scannedFiles (
    int filesScanned ) [signal]
```

Emitted between startScanningAlbum and finishedScanningAlbum. In between these two signals, the sum of files Scanned of all sent signals equals the one reported by finishedScanningAlbum()

**6.198.2.16 setNeedFileCount()**

```
void Digikam::CollectionScanner::setNeedFileCount (
    bool on )
```

Call this to enable emitting the total files to scan (for progress info) before a complete collection scan. Default is off. If on, setSignalEnabled() must be on to take effect.

**6.198.2.17 setObserver()**

```
void Digikam::CollectionScanner::setObserver (
    CollectionScannerObserver *const observer )
```

Set an observer to be able to cancel a running scan

**6.198.2.18 setPerformFastScan()**

```
void Digikam::CollectionScanner::setPerformFastScan (
    bool on )
```

Call this to disable fast scan with album date check. Default is on.

**6.198.2.19 setSignalsEnabled()**

```
void Digikam::CollectionScanner::setSignalsEnabled (
    bool on )
```

Call this to enable the progress info signals. Default is off.

**6.198.2.20 startScanningAlbumRoot**

```
void Digikam::CollectionScanner::startScanningAlbumRoot (
    const QString & albumRoot ) [signal]
```

Notifies the begin of the scanning of the specified album root, album, of stale files, or of the whole collection (after stale files)

**6.198.2.21 totalFilesToScan**

```
void Digikam::CollectionScanner::totalFilesToScan (
    int count ) [signal]
```

Emitted once in scanAlbums(), the scan() methods, and updateItemsWithoutDate(). Gives the number of the files that need to be scanned.

**6.199 Digikam::CollectionScanner::Private Class Reference****Public Member Functions**

- bool **checkDeferred** (const QFileInfo &info)
- bool **checkObserver** ()
- void **finishScanner** (ItemScanner &scanner)
- void **removedItems** ()
- void **resetRemovedItemsTime** ()

**Public Attributes**

- QHash< QString, QDateTime > **albumDateCache**
- QSet< QString > **audioFilterSet**
- QSet< QString > **deferredAlbumPaths**
- bool **deferredFileScanning** = false
- QHash< int, int > **establishedSourceAlbums**
- [CollectionScannerHintContainerImplementation](#) \* **hints** = nullptr
- QSet< QString > **ignoreDirectory**
- QSet< QString > **imageFilterSet**
- QSet< QString > **nameFilters**
- QSet< qlonglong > **needResolveHistorySet**
- QSet< qlonglong > **needTaggingHistorySet**
- bool **needTotalFiles** = false
- QList< qlonglong > **newIdsList**
- [CollectionScannerObserver](#) \* **observer** = nullptr
- bool **performFastScan** = true
- bool **recordHistoryIds** = false
- QDateTime **removedItemsTime**
- QList< int > **scannedAlbums**
- bool **updatingHashHint** = false
- QSet< QString > **videoFilterSet**
- bool **wantSignals** = false

## 6.199.1 Member Function Documentation

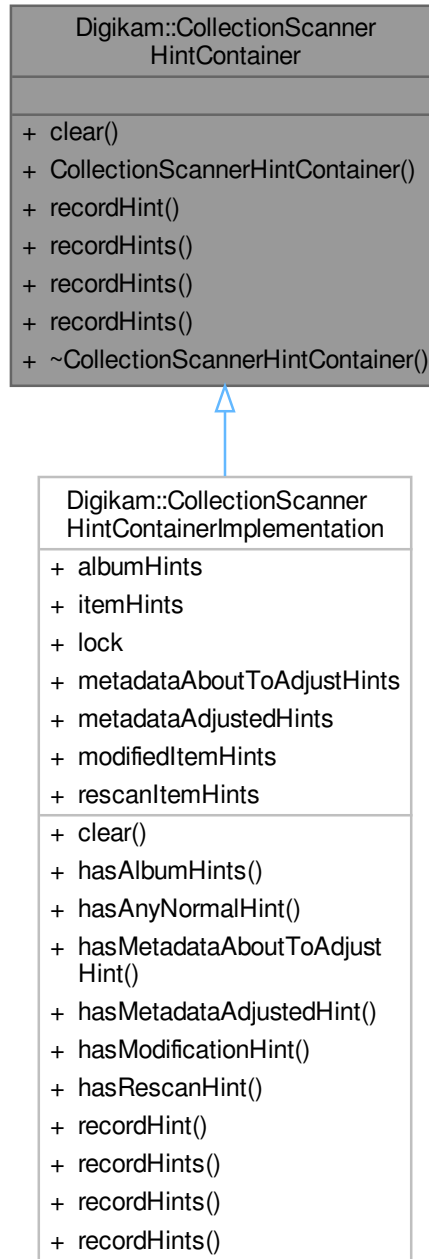
### 6.199.1.1 finishScanner()

```
void Digikam::CollectionScanner::Private::finishScanner (
    ItemScanner & scanner )
```

Perform the actual write operation to the database

## 6.200 Digikam::CollectionScannerHintContainer Class Reference

Inheritance diagram for Digikam::CollectionScannerHintContainer:



### Public Member Functions

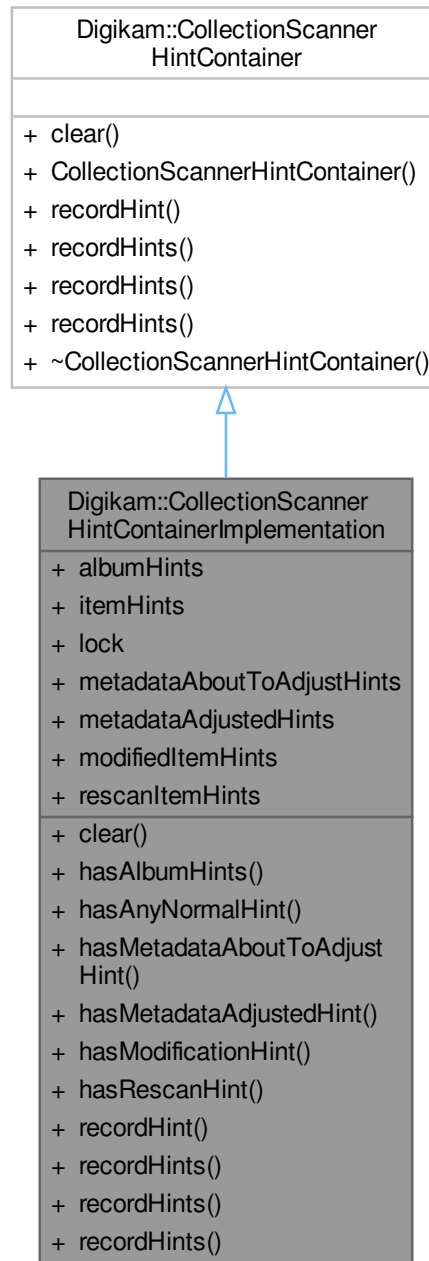
- virtual void **clear** ()=0
- **CollectionScannerHintContainer** ()=default

*Note: All methods of this class must be thread-safe.*

- virtual void **recordHint** (const [ItemMetadataAdjustmentHint](#) &hints)=0
- virtual void **recordHints** (const QList< [AlbumCopyMoveHint](#) > &hints)=0
- virtual void **recordHints** (const QList< [ItemChangeHint](#) > &hints)=0
- virtual void **recordHints** (const QList< [ItemCopyMoveHint](#) > &hints)=0

## 6.201 Digikam::CollectionScannerHintContainerImplementation Class Reference

Inheritance diagram for Digikam::CollectionScannerHintContainerImplementation:



### Public Member Functions

- void `clear()` override



- bool **hasAlbumHints** ()
- bool **hasAnyNormalHint** (qlonglong id)
- bool **hasMetadataAboutToAdjustHint** (qlonglong id)
- bool **hasMetadataAdjustedHint** (qlonglong id)
- bool **hasModificationHint** (qlonglong id)
- bool **hasRescanHint** (qlonglong id)
- void **recordHint** (const [ItemMetadataAdjustmentHint](#) &hint) override
- void **recordHints** (const QList< [AlbumCopyMoveHint](#) > &hints) override
- void **recordHints** (const QList< [ItemChangeHint](#) > &hints) override
- void **recordHints** (const QList< [ItemCopyMoveHint](#) > &hints) override

## Public Member Functions inherited from [Digikam::CollectionScannerHintContainer](#)

- **CollectionScannerHintContainer** ()=default

*Note: All methods of this class must be thread-safe.*

## Public Attributes

- QHash< [CollectionScannerHints::DstPath](#), [CollectionScannerHints::Album](#) > **albumHints**
- QHash< [NewlyAppearedFile](#), qlonglong > **itemHints**
- QReadWriteLock **lock**
- QHash< qlonglong, QDateTime > **metadataAboutToAdjustHints**
- QHash< qlonglong, QDateTime > **metadataAdjustedHints**
- QSet< qlonglong > **modifiedItemHints**
- QSet< qlonglong > **rescanItemHints**

## 6.201.1 Member Function Documentation

### 6.201.1.1 clear()

```
void Digikam::CollectionScannerHintContainerImplementation::clear ( ) [override], [virtual]
```

Implements [Digikam::CollectionScannerHintContainer](#).

### 6.201.1.2 recordHint()

```
void Digikam::CollectionScannerHintContainerImplementation::recordHint (
    const ItemMetadataAdjustmentHint & hint ) [override], [virtual]
```

Implements [Digikam::CollectionScannerHintContainer](#).

### 6.201.1.3 recordHints() [1/3]

```
void Digikam::CollectionScannerHintContainerImplementation::recordHints (
    const QList< AlbumCopyMoveHint > & hints ) [override], [virtual]
```

Implements [Digikam::CollectionScannerHintContainer](#).

**6.201.1.4 recordHints()** [2/3]

```
void Digikam::CollectionScannerHintContainerImplementation::recordHints (
    const QList< ItemChangeHint > & hints ) [override], [virtual]
```

Implements [Digikam::CollectionScannerHintContainer](#).

**6.201.1.5 recordHints()** [3/3]

```
void Digikam::CollectionScannerHintContainerImplementation::recordHints (
    const QList< ItemCopyMoveHint > & hints ) [override], [virtual]
```

Implements [Digikam::CollectionScannerHintContainer](#).

## 6.202 Digikam::CollectionScannerObserver Class Reference

Inheritance diagram for Digikam::CollectionScannerObserver:

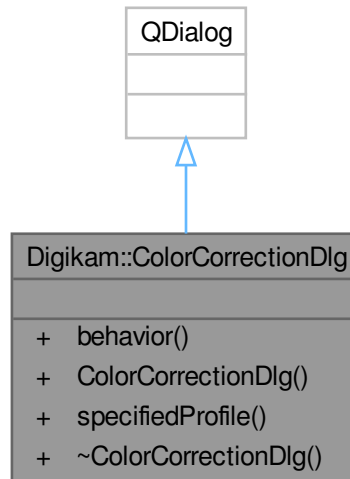


### Public Member Functions

- virtual bool `continueQuery ()=0`

## 6.203 Digikam::ColorCorrectionDlg Class Reference

Inheritance diagram for Digikam::ColorCorrectionDlg:



### Public Types

- enum **Mode** { **ProfileMismatch** , **MissingProfile** , **UncalibratedColor** }

### Public Member Functions

- `ICCSettingsContainer::Behavior` **behavior** () const
- **ColorCorrectionDlg** (Mode mode, const [DImg](#) &preview, const QString &file, QWidget \*const parent=nullptr)
- [IccProfile](#) **specifiedProfile** () const

## 6.204 Digikam::ColorFXContainer Class Reference

### Public Attributes

- int **colorFXType** = 0  
*ColorFXFilter::Solarize.*
- int **intensity** = 100
- int **iterations** = 2
- int **level** = 0
- QString **path**

## 6.205 Digikam::ColorFXFilter Class Reference

Inheritance diagram for Digikam::ColorFXFilter:



### Public Types

- enum **ColorFXFilterTypes** {
  - Solarize** = 0 , **Vivid** , **Neon** , **FindEdges** , **Lut3D** }

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Member Functions

- **ColorFXFilter** ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [ColorFXContainer](#) &settings=[ColorFXContainer](#)())
- **ColorFXFilter** ([QObject](#) \*const parent=nullptr)
- [FilterAction](#) filterAction () override
- [QString](#) filterIdentifier () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > [multithreadedSteps](#) (int **stop**, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > **supportedVersions** () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int **CurrentVersion** ()
- static [QString](#) **DisplayableName** ()
- static [QString](#) **FilterIdentifier** ()
- static [QList](#)< int > **SupportedVersions** ()

### Additional Inherited Members

#### Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

#### Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

#### Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

#### Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

#### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

#### Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
- *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.205.1 Member Function Documentation

### 6.205.1.1 filterAction()

`FilterAction` Digikam::ColorFXFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.205.1.2 filterIdentifier()

`QString` Digikam::ColorFXFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

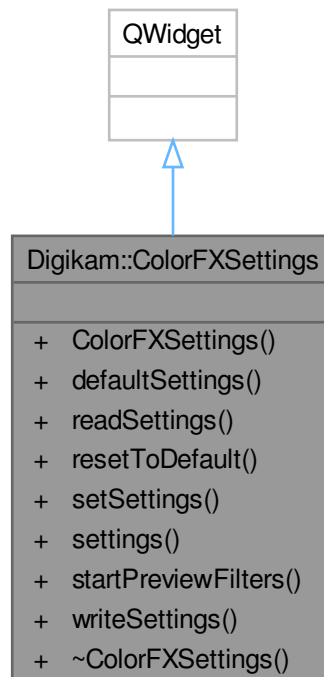
### 6.205.1.3 readParameters()

`void` Digikam::ColorFXFilter::readParameters (   
     const `FilterAction` & *action* ) [override], [virtual]

Implements [Digikam::DImgThreadedFilter](#).

## 6.206 Digikam::ColorFXSettings Class Reference

Inheritance diagram for Digikam::ColorFXSettings:





## Signals

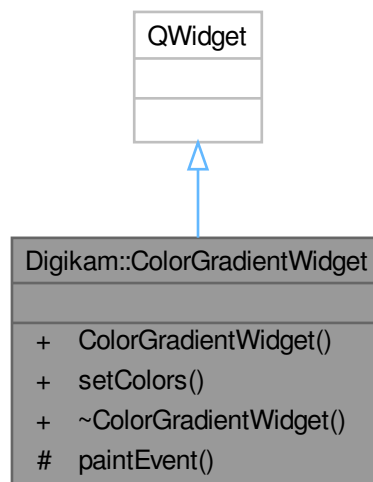
- void **signalSettingsChanged** ()

## Public Member Functions

- **ColorFXSettings** (QWidget \*const parent, bool useGenericImg=true)
- **ColorFXContainer defaultSettings** () const
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const **ColorFXContainer** &settings)
- **ColorFXContainer settings** () const
- void **startPreviewFilters** ()
- void **writeSettings** (KConfigGroup &group)

## 6.207 Digikam::ColorGradientWidget Class Reference

Inheritance diagram for Digikam::ColorGradientWidget:



## Public Member Functions

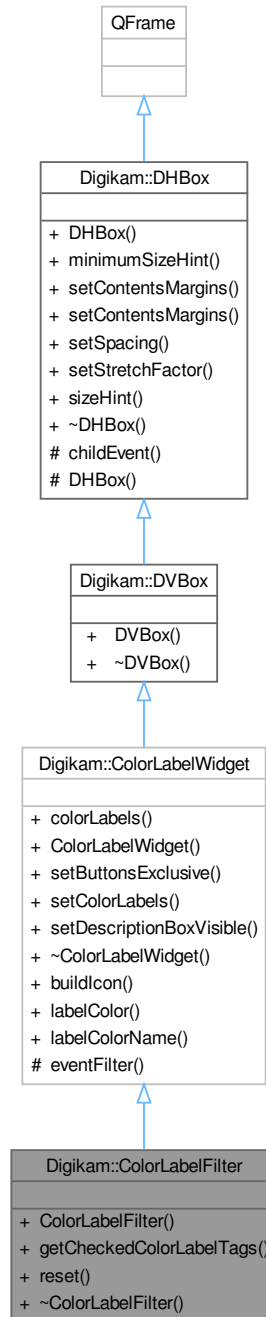
- **ColorGradientWidget** (Qt::Orientation orientation, int size, QWidget \*const parent=nullptr)
- void **setColors** (const QColor &col1, const QColor &col2)

## Protected Member Functions

- void **paintEvent** (QPaintEvent \*) override

## 6.208 Digikam::ColorLabelFilter Class Reference

Inheritance diagram for Digikam::ColorLabelFilter:



### Signals

- void **signalColorLabelSelectionChanged** (const QList< ColorLabel > &)

## Signals inherited from [Digikam::ColorLabelWidget](#)

- void **signalColorLabelChanged** (int)

## Public Member Functions

- **ColorLabelFilter** (QWidget \*const parent=nullptr)
- QList< [TAlbum](#) \* > **getCheckedColorLabelTags** ()
- void **reset** ()

## Public Member Functions inherited from [Digikam::ColorLabelWidget](#)

- QList< [ColorLabel](#) > **colorLabels** () const
- **ColorLabelWidget** (QWidget \*const parent=nullptr)
- void **setButtonsExclusive** (bool b)
- void **setColorLabels** (const QList< [ColorLabel](#) > &list)
- void **setDescriptionBoxVisible** (bool b)

## Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::ColorLabelWidget](#)

- static QIcon **buildIcon** ([ColorLabel](#) label, int size=12)
- static QColor **labelColor** ([ColorLabel](#) label)
- static QString **labelColorName** ([ColorLabel](#) label)

## Protected Member Functions inherited from [Digikam::ColorLabelWidget](#)

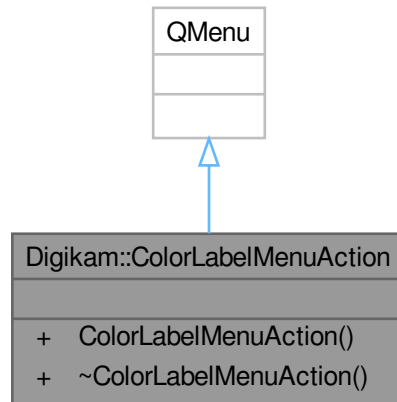
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.209 Digikam::ColorLabelMenuAction Class Reference

Inheritance diagram for Digikam::ColorLabelMenuAction:



### Signals

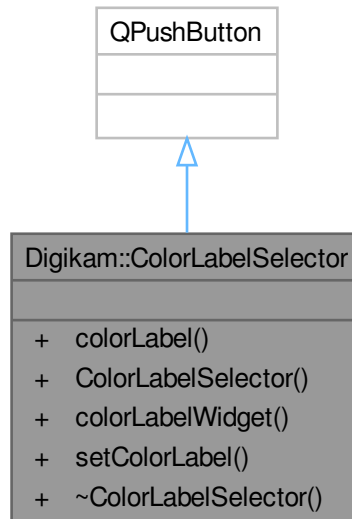
- void **signalColorLabelChanged** (int)

### Public Member Functions

- **ColorLabelMenuAction** (QMenu \*const parent=nullptr)

## 6.210 Digikam::ColorLabelSelector Class Reference

Inheritance diagram for Digikam::ColorLabelSelector:



### Signals

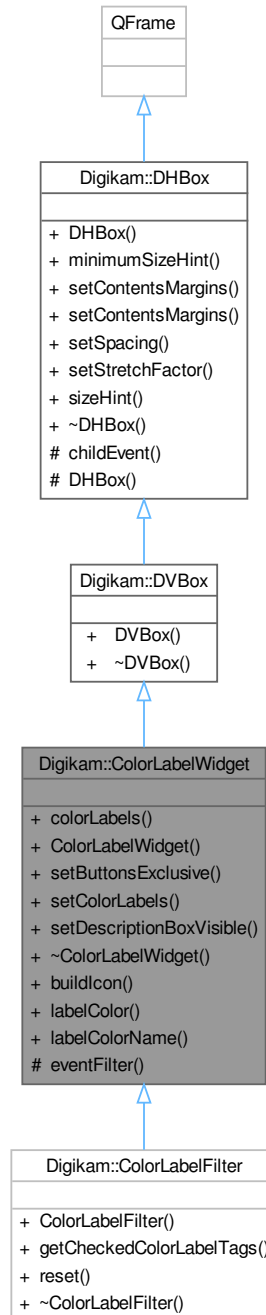
- void **signalColorLabelChanged** (int)

### Public Member Functions

- ColorLabel **colorLabel** ()
- **ColorLabelSelector** (QWidget \*const parent=nullptr)
- [ColorLabelWidget](#) \* **colorLabelWidget** () const
- void **setColorLabel** (ColorLabel label)

## 6.211 Digikam::ColorLabelWidget Class Reference

Inheritance diagram for Digikam::ColorLabelWidget:



### Signals

- void **signalColorLabelChanged** (int)

## Public Member Functions

- `QList< ColorLabel > colorLabels () const`
- `ColorLabelWidget (QWidget *const parent=nullptr)`
- `void setButtonsExclusive (bool b)`
- `void setColorLabels (const QList< ColorLabel > &list)`
- `void setDescriptionBoxVisible (bool b)`

## Public Member Functions inherited from Digikam::DVBox

- `DVBox (QWidget *const parent=nullptr)`

## Public Member Functions inherited from Digikam::DHBox

- `DHBox (QWidget *const parent=nullptr)`
- `QSize minimumSizeHint () const override`
- `void setContentsMargins (const QMargins &margins)`
- `void setContentsMargins (int left, int top, int right, int bottom)`
- `void setSpacing (int space)`
- `void setStretchFactor (QWidget *const widget, int stretch)`
- `QSize sizeHint () const override`

## Static Public Member Functions

- `static QIcon buildIcon (ColorLabel label, int size=12)`
- `static QColor labelColor (ColorLabel label)`
- `static QString labelColorName (ColorLabel label)`

## Protected Member Functions

- `bool eventFilter (QObject *obj, QEvent *ev) override`

## Protected Member Functions inherited from Digikam::DHBox

- `void childEvent (QChildEvent *e) override`
- `DHBox (bool vertical, QWidget *const parent)`

## 6.211.1 Member Function Documentation

### 6.211.1.1 colorLabels()

```
QList< ColorLabel > Digikam::ColorLabelWidget::colorLabels ( ) const
```

Return the list of Color Label buttons turned on or an empty list of none.

### 6.211.1.2 `setButtonsExclusive()`

```
void Digikam::ColorLabelWidget::setButtonsExclusive (
    bool b )
```

Set all Color Label buttons exclusive or not. Default is true as only one can be selected. Non-exclusive mode is dedicated for Advanced Search tool.

### 6.211.1.3 `setColorLabels()`

```
void Digikam::ColorLabelWidget::setColorLabels (
    const QList< ColorLabel > & list )
```

Turn on Color Label buttons using list. Pass an empty list to clear all selection.

### 6.211.1.4 `setDescriptionBoxVisible()`

```
void Digikam::ColorLabelWidget::setDescriptionBoxVisible (
    bool b )
```

Show or not on the bottom view the description of label with shortcuts.



## 6.212 Digikam::ComboBoxDelegate Class Reference

Inheritance diagram for Digikam::ComboBoxDelegate:



### Public Member Functions

- **ComboBoxDelegate** (`DItemsList *const`, `const QMap< int, QString > &`)
- void **startEditing** (`QTreeWidgetItem *`, `int`)
- void **paint** (`QPainter *`, `const QStyleOptionViewItem &`, `const QModelIndex &`) `const` override
- `QSize` **sizeHint** (`const QStyleOptionViewItem &`, `const QModelIndex &`) `const` override
- `QWidget *` **createEditor** (`QWidget *`, `const QStyleOptionViewItem &`, `const QModelIndex &`) `const` override
- void **setEditorData** (`QWidget *`, `const QModelIndex &`) `const` override
- void **setModelData** (`QWidget *`, `QAbstractItemModel *`, `const QModelIndex &`) `const` override

## 6.212.1 Member Function Documentation

### 6.212.1.1 `paint()`

```
void Digikam::ComboBoxDelegate::paint (
    QPainter * painter,
    const QStyleOptionViewItem & option,
    const QModelIndex & index ) const [override]
```

Overloaded functions to provide the delegate functionality.

### 6.212.1.2 `startEditing()`

```
void Digikam::ComboBoxDelegate::startEditing (
    QTreeWidgetItem * item,
    int column )
```

Whenever an element needs to be edited, this method should be called. It's actually a hack to prevent the item text shining through whenever editing occurs.

## 6.213 Digikam::CommentInfo Class Reference

### Public Member Functions

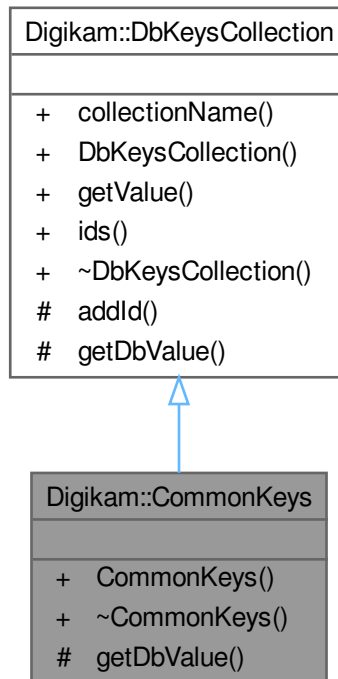
- `bool isNull () const`

### Public Attributes

- `QString author`
- `QString comment`
- `QDateTime date`
- `int id = -1`
- `qulonglong imageld = -1`
- `QString language`
- `DatabaseComment::Type type = DatabaseComment::UndefinedType`

## 6.214 Digikam::CommonKeys Class Reference

Inheritance diagram for Digikam::CommonKeys:



### Protected Member Functions

- `QString` [getDbValue](#) (const `QString` &key, [ParseSettings](#) &settings) override

### Protected Member Functions inherited from [Digikam::DbKeysCollection](#)

- void [addId](#) (const `QString` &id, const `QString` &description)

### Additional Inherited Members

### Public Member Functions inherited from [Digikam::DbKeysCollection](#)

- `QString` [collectionName](#) () const
- [DbKeysCollection](#) (const `QString` &n)
- `QString` [getValue](#) (const `QString` &key, [ParseSettings](#) &settings)
- `DbKeyIdsMap` [ids](#) () const

## 6.214.1 Member Function Documentation

### 6.214.1.1 getDbValue()

```
QString Digikam::CommonKeys::getDbValue (
    const QString & key,
    ParseSettings & settings ) [override], [protected], [virtual]
```

Abstract method for retrieving the value from the database for the given key.

This method has to be implemented by all child classes. It is called by the [getValue\(\)](#) method.

#### Parameters

<i>key</i>	the key representing the value in the database
<i>settings</i>	the ParseSettings object holding all relevant information about the image.

#### Returns

the value of the given database key

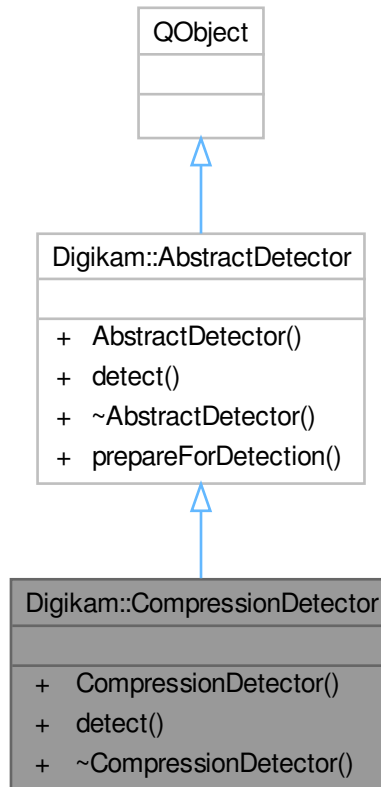
#### See also

[DbKeysCollection::getValue\(\)](#)

Implements [Digikam::DbKeysCollection](#).

## 6.215 Digikam::CompressionDetector Class Reference

Inheritance diagram for Digikam::CompressionDetector:



### Public Member Functions

- float [detect](#) (const cv::Mat &image) const override

### Public Member Functions inherited from [Digikam::AbstractDetector](#)

- **AbstractDetector** (QObject \*const parent=nullptr)

### Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::AbstractDetector](#)

- static cv::Mat [prepareForDetection](#) (const [DImg](#) &inputImage)

## 6.215.1 Member Function Documentation

### 6.215.1.1 detect()

```
float Digikam::CompressionDetector::detect (
    const cv::Mat & image ) const [override], [virtual]
```

Implements [Digikam::AbstractDetector](#).

## 6.216 Digikam::ContentAwareContainer Class Reference

### Public Types

- enum **EnergyFunction** {  
    **GradientNorm** = 0 , **SumOfAbsoluteValues** , **XAbsoluteValue** , **LumaGradientNorm** ,  
    **LumaSumOfAbsoluteValues** , **LumaXAbsoluteValue** }

### Public Attributes

- EnergyFunction **func** = GradientNorm
- uint **height** = 0
- QImage **mask**
- bool **preserve\_skin\_tones** = false
- Qt::Orientation **resize\_order** = Qt::Horizontal
- double **rigidity** = 0.0
- int **side\_switch\_freq** = 4
- int **step** = 1
- uint **width** = 0

## 6.217 Digikam::ContentAwareFilter Class Reference

Inheritance diagram for Digikam::ContentAwareFilter:



### Public Member Functions

- **ContentAwareFilter** (`Dimg *const orgImage`, `QObject *const parent=nullptr`, `const ContentAwareContainer &settings=ContentAwareContainer()`)

- **ContentAwareFilter** (QObject \*const parent=nullptr)
- [FilterAction filterAction](#) () override
- QString [filterIdentifier](#) () const override
- void **progressCallback** (int progress)
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- [DImgThreadedFilter](#) (DImg \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }



## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.217.1 Member Function Documentation

### 6.217.1.1 filterAction()

```
FilterAction Digikam::ContentAwareFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.217.1.2 filterIdentifier()

```
QString Digikam::ContentAwareFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.217.1.3 readParameters()

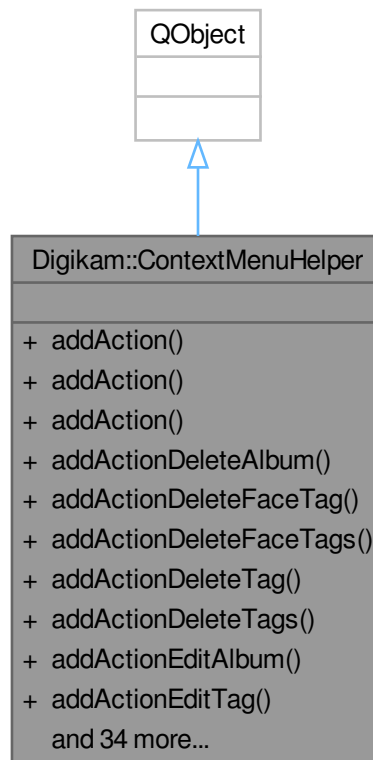
```
void Digikam::ContentAwareFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.218 Digikam::ContextMenuHelper Class Reference

A helper class to add actions and special menus to the context menu.

Inheritance diagram for Digikam::ContextMenuHelper:



## Classes

- class [Private](#)

## Public Types

- typedef const QList< qlonglong > **imagelds**

## Signals

- void **signalAddNewTagFromABCMenu** (const QString &)
- void **signalAddToExistingQueue** (int)
- void **signalAssignColorLabel** (int)
- void **signalAssignPickLabel** (int)
- void **signalAssignRating** (int)
- void **signalAssignTag** (int)
- void **signalCreateGroup** ()
- void **signalCreateGroupByFilename** ()
- void **signalCreateGroupByTime** ()
- void **signalCreateGroupByTimelapse** ()

- void **signalGotoAlbum** (const [ItemInfo](#) &)
- void **signalGotoDate** (const [ItemInfo](#) &)
- void **signalGotoTag** (int)
- void **signalPopupTagsView** ()
- void **signalRemoveFromGroup** ()
- void **signalRemoveTag** (int)
- void **signalSetThumbnail** (const [ItemInfo](#) &)
- void **signalUngroup** ()

## Public Member Functions

- void [addAction](#) (const QString &name, bool addDisabled=false)
- void [addAction](#) (QAction \*const action, bool addDisabled=false)
- void [addAction](#) (QAction \*const action, QObject \*const recv, const char \*const slot, bool addDisabled=false)
- void **addActionDeleteAlbum** ([AlbumModificationHelper](#) \*const helper, [PAlbum](#) \*const album)
- void [addActionDeleteFaceTag](#) ([TagModificationHelper](#) \*const helper, [TAlbum](#) \*const tag)
- void **addActionDeleteFaceTags** ([TagModificationHelper](#) \*const helper, const QList< [TAlbum](#) \* > &tags)
- void **addActionDeleteTag** ([TagModificationHelper](#) \*const helper, [TAlbum](#) \*const tag)
- void **addActionDeleteTags** ([TagModificationHelper](#) \*const helper, const QList< [TAlbum](#) \* > &tags)
- void **addActionEditAlbum** ([AlbumModificationHelper](#) \*const helper, [PAlbum](#) \*const album)
- void **addActionEditTag** ([TagModificationHelper](#) \*const helper, [TAlbum](#) \*const tag)
- void [addActionNewAlbum](#) ([AlbumModificationHelper](#) \*const helper, [PAlbum](#) \*const parentAlbum=nullptr)
- void [addActionNewTag](#) ([TagModificationHelper](#) \*const helper, [TAlbum](#) \*const parentTag=nullptr)
- void **addActionRenameAlbum** ([AlbumModificationHelper](#) \*const helper, [PAlbum](#) \*const album)
- void **addActionResetAlbumIcon** ([AlbumModificationHelper](#) \*const helper, [PAlbum](#) \*const album)
- void **addActionTagsToFaceTags** ([TagModificationHelper](#) \*const helper, const QList< [TAlbum](#) \* > &tags)
- void [addActionTagToFaceTag](#) ([TagModificationHelper](#) \*const helper, [TAlbum](#) \*const tag)
- void [addAlbumCheckUncheckActions](#) ([Album](#) \*const album)
- void [addAssignTagsMenu](#) (const imagelds &ids)
- void [addCreateTagFromAddressbookMenu](#) ()
- void [addExportMenu](#) ()
- void [addGotoMenu](#) (const imagelds &ids)
- void **addGroupActions** (const imagelds &ids)
- void [addGroupMenu](#) (const imagelds &ids, const QList< QAction \* > &extraMenuItems=QList< QAction \* >())
- void [addImportMenu](#) ()
- void [addIQSAction](#) (QObject \*const recv, const char \*const slot)
- void [addLabelsAction](#) ()
- void [addOpenAndNavigateActions](#) (const imagelds &ids, bool lightTable=false)
- void [addQueueManagerMenu](#) ()
- void [addRemoveAllTags](#) (const imagelds &ids)
- void [addRemoveTagsMenu](#) (const imagelds &ids)
- void [addSeparator](#) ()
- void [addServicesMenu](#) (const QList< QUrl > &selectedItems)
- void [addStandardActionCopy](#) (QObject \*const recv, const char \*const slot)
- void [addStandardActionCut](#) (QObject \*const recv, const char \*const slot)
- void [addStandardActionItemDelete](#) (QObject \*const recv, const char \*const slot, int quantity=1)
- void [addStandardActionLightTable](#) ()
- void [addStandardActionPaste](#) (QObject \*const recv, const char \*const slot)
- void [addStandardActionThumbnail](#) (const imagelds &ids, [Album](#) \*const album)
- void [addSubMenu](#) (QMenu \*subMenu)
- [ContextMenuHelper](#) (QMenu \*const parent)
- QAction \* [exec](#) (const QPoint &pos, QAction \*const at=nullptr)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)
- void [setItemFilterModel](#) ([ItemFilterModel](#) \*const model)

## 6.218.1 Detailed Description

The ContextMenuHelper class helps adding commonly used actions and menus. Use this class to add

- actions from the actionCollection
- standard actions (copy, paste, delete)
- temporary actions
- predefined special actions
- predefined submenus to the menu.

All [addAction\(\)](#) methods take a special parameter 'addDisabled'. This parameter controls if disabled actions are added to the menu. Normally adding disabled actions is turned off, to clean up the menu and make it more readable.

If the ContextMenuHelper class is used, you need to call its own [exec\(\)](#) method, instead the one from the parent menu. This way signals from special menus can be emitted and connected to the appropriate slots.

## 6.218.2 Constructor & Destructor Documentation

### 6.218.2.1 ContextMenuHelper()

```
Digikam::ContextMenuHelper::ContextMenuHelper (
    QMenu *const parent ) [explicit]
```

Constructs the helper class.

#### Parameters

<i>parent</i>	the menu the helper class is linked to
---------------	--

## 6.218.3 Member Function Documentation

### 6.218.3.1 addAction() [1/3]

```
void Digikam::ContextMenuHelper::addAction (
    const QString & name,
    bool addDisabled = false )
```

Add an action from the actionCollection.

This method adds actions from the actionCollection. The actionCollection can be set in the constructor of the [ContextMenuHelper](#) class.

#### Parameters

<i>name</i>	the name of the action in the actionCollection
<i>addDisabled</i>	if set, disabled actions are added to the menu

**6.218.3.2 addAction() [2/3]**

```
void Digikam::ContextMenuHelper::addAction (
    QAction *const action,
    bool addDisabled = false )
```

Add a temporary action.

Sometimes it is necessary to define actions that only exist in the current context menu content. Use this method to add such an action.

**Parameters**

<i>action</i>	the action to add
<i>addDisabled</i>	if set, disabled actions are added to the menu

**6.218.3.3 addAction() [3/3]**

```
void Digikam::ContextMenuHelper::addAction (
    QAction *const action,
    QObject *const recv,
    const char *const slot,
    bool addDisabled = false )
```

Add a temporary action and assign it to a custom slot.

Use this method if you want to add a temporary action and immediately connect it to the receiving slot.

**Parameters**

<i>action</i>	the action to add
<i>recv</i>	the receiver of the triggered action
<i>slot</i>	the slot to connect the triggered action to
<i>addDisabled</i>	if set, disabled actions are added to the menu

**6.218.3.4 addActionDeleteFaceTag()**

```
void Digikam::ContextMenuHelper::addActionDeleteFaceTag (
    TagModificationHelper *const helper,
    TAlbum *const tag )
```

Add action to delete tags from people sidebar.

**6.218.3.5 addActionNewAlbum()**

```
void Digikam::ContextMenuHelper::addActionNewAlbum (
    AlbumModificationHelper *const helper,
    PAlbum *const parentAlbum = nullptr )
```

Add actions to add, remove or edit a tag. The tag modification helper is used to execute the action. You must set the parent tag to use on modification helper.

### 6.218.3.6 addActionNewTag()

```
void Digikam::ContextMenuHelper::addActionNewTag (
    TagModificationHelper *const helper,
    TAlbum *const parentTag = nullptr )
```

Add actions to add, remove or edit a tag. The tag modification helper is used to execute the action. You must set the parent tag to use on modification helper.

### 6.218.3.7 addActionTagToFaceTag()

```
void Digikam::ContextMenuHelper::addActionTagToFaceTag (
    TagModificationHelper *const helper,
    TAlbum *const tag )
```

Add action to set tags as face tags.

### 6.218.3.8 addAlbumCheckUncheckActions()

```
void Digikam::ContextMenuHelper::addAlbumCheckUncheckActions (
    Album *const album )
```

Add a Select and Deselect menu to check and uncheck albums. Note: Call setAlbumModel before, or this will have no effect.

### 6.218.3.9 addAssignTagsMenu()

```
void Digikam::ContextMenuHelper::addAssignTagsMenu (
    const imageIds & ids )
```

Add "Assign Tags" menu.

This menu will provide a list of all tags available so that they can be assigned to the current selected items.

To make this menu work, you need to run `exec()` from this class, otherwise the signals are not emitted and you will not be able to react on triggered actions from this menu. Make sure to connect the signals to the appropriate slots in the context menu handling method.

#### Parameters

<code>ids</code>	the selected items
------------------	--------------------

#### See also

[exec\(\)](#)  
[signalAssignTag\(\)](#)

### 6.218.3.10 addCreateTagFromAddressbookMenu()

```
void Digikam::ContextMenuHelper::addCreateTagFromAddressbookMenu ( )
```

Add a menu to create new tags from adressbook entries.

### 6.218.3.11 addExportMenu()

```
void Digikam::ContextMenuHelper::addExportMenu ( )
```

Add Export Webservices actions menu.

### 6.218.3.12 addGotoMenu()

```
void Digikam::ContextMenuHelper::addGotoMenu (
    const imageIds & ids )
```

Add the Goto menu.

This menu will provide the following actions for the given item:

- Goto [Album](#)
- Goto Date
- Goto Tag To make this menu work, you need to run [exec\(\)](#) from this class, otherwise the signals are not emitted and you will not be able to react on triggered actions from this menu. Make sure to connect the signals to the appropriate slots in the context menu handling method.

#### Parameters

<i>ids</i>	the list of selected items
------------	----------------------------

#### See also

[exec\(\)](#)  
[signalGotoAlbum\(\)](#) [signalGotoDate\(\)](#) [signalGotoTag\(\)](#)

TODO:tags to be ported to multiple selection

### 6.218.3.13 addGroupMenu()

```
void Digikam::ContextMenuHelper::addGroupMenu (
    const imageIds & ids,
    const QList< QAction * > & extraMenuItems = QList<QAction*> ( ) )
```

Add a "Group" menu. This menu will provide actions open, close, add to, remove from, or split a group.

`addGroupActions` will add the actions as a flat list, not in a submenu. Note: Call `setItemFilterModel` before to have Open/Close group actions.



### 6.218.3.14 addImportMenu()

```
void Digikam::ContextMenuHelper::addImportMenu ( )
```

Add Import Webservices actions menu.

### 6.218.3.15 addIQSAction()

```
void Digikam::ContextMenuHelper::addIQSAction (
    QObject *const recv,
    const char *const slot )
```

Add the standard Image Quality Sorter action and connect it to the appropriate slot

#### Parameters

<i>recv</i>	the receiver of the triggered action
<i>slot</i>	the slot to connect the triggered action to

### 6.218.3.16 addLabelsAction()

```
void Digikam::ContextMenuHelper::addLabelsAction ( )
```

Add "Pick/Color/Rating Labels" action.

This action will provide methods to assign pick/color/rating labels to the currently selected items.

To make this menu work, you need to run [exec\(\)](#) from this class, otherwise the signals are not emitted and you will not be able to react on triggered actions from this menu. Make sure to connect the signals to the appropriate slots in the context menu handling method.

#### See also

- [exec\(\)](#)
- [signalAssignPickLabel\(\)](#)
- [signalAssignColorLabel\(\)](#)
- [signalAssignRating\(\)](#)

### 6.218.3.17 addOpenAndNavigateActions()

```
void Digikam::ContextMenuHelper::addOpenAndNavigateActions (
    const imageIds & ids,
    bool lightTable = false )
```

Add section for main views for opening and moving/going to albums.

This is a convenience function to ensure consistent menus and reduce code duplication.

## Parameters

<i>ids</i>	the list of selected items
<i>lightTable</i>	for the light table

**6.218.3.18 addQueueManagerMenu()**

```
void Digikam::ContextMenuHelper::addQueueManagerMenu ( )
```

Add Queue Manager actions menu.

**6.218.3.19 addRemoveAllTags()**

```
void Digikam::ContextMenuHelper::addRemoveAllTags (
    const imageIds & ids )
```

Add "Remove all Tags" action.

Removes all tags from the selected item ids except face tags.

## Parameters

<i>ids</i>	the selected items
------------	--------------------

**6.218.3.20 addRemoveTagsMenu()**

```
void Digikam::ContextMenuHelper::addRemoveTagsMenu (
    const imageIds & ids )
```

Add "Remove Tags" menu.

This menu will provide a list of all tags assigned to the current items. Actions triggered in here will remove the selected tag from the items.

To make this menu work, you need to run [exec\(\)](#) from this class, otherwise the signals are not emitted and you will not be able to react on triggered actions from this menu. Make sure to connect the signals to the appropriate slots in the context menu handling method.

## Parameters

<i>ids</i>	the selected items
------------	--------------------

## See also

[exec\(\)](#)  
[signalRemoveTag\(\)](#)

### 6.218.3.21 addSeparator()

```
void Digikam::ContextMenuHelper::addSeparator ( )
```

Add a separator to the context menu

### 6.218.3.22 addServicesMenu()

```
void Digikam::ContextMenuHelper::addServicesMenu (
    const QList< QUrl > & selectedItems )
```

Add the services menu to the menu.

The services menu is used to open the selected items in a different application. It will query the item for registered services and provide them in a submenu. The menu will be titled "Open With...".

#### Parameters

<i>selectedItems</i>	the list of selected items
----------------------	----------------------------

### 6.218.3.23 addStandardActionCopy()

```
void Digikam::ContextMenuHelper::addStandardActionCopy (
    QObject *const recv,
    const char *const slot )
```

Add the standard copy action and connect it to the appropriate slot

#### Parameters

<i>recv</i>	the receiver of the triggered action
<i>slot</i>	the slot to connect the triggered action to

### 6.218.3.24 addStandardActionCut()

```
void Digikam::ContextMenuHelper::addStandardActionCut (
    QObject *const recv,
    const char *const slot )
```

Add the standard cut action and connect it to the appropriate slot

#### Parameters

<i>recv</i>	the receiver of the triggered action
<i>slot</i>	the slot to connect the triggered action to

**6.218.3.25 addStandardActionItemDelete()**

```
void Digikam::ContextMenuHelper::addStandardActionItemDelete (
    QObject *const recv,
    const char *const slot,
    int quantity = 1 )
```

Add the standard delete action and connect it to the appropriate slot

**Parameters**

<i>recv</i>	the receiver of the triggered action
<i>slot</i>	the slot to connect the triggered action to
<i>quantity</i>	the number of the files that should be deleted. This parameter is used for the action name and is normally used when deleting more then one item.

**6.218.3.26 addStandardActionLightTable()**

```
void Digikam::ContextMenuHelper::addStandardActionLightTable ( )
```

Add the lighttable action to the menu.

Do not use [addAction\(\)](#) to add the lighttable action, because we need to handle special cases here. Depending on whether the lighttable window has already been created and filled with items, we set different actions.

**6.218.3.27 addStandardActionPaste()**

```
void Digikam::ContextMenuHelper::addStandardActionPaste (
    QObject *const recv,
    const char *const slot )
```

Add the standard paste action and connect it to the appropriate slot

**Parameters**

<i>recv</i>	the receiver of the triggered action
<i>slot</i>	the slot to connect the triggered action to

**6.218.3.28 addStandardActionThumbnail()**

```
void Digikam::ContextMenuHelper::addStandardActionThumbnail (
    const imageIds & ids,
    Album *const album )
```

Add the thumbnail action to the menu.

Do not use [addAction\(\)](#) to add the thumbnail action, because we need to handle special cases here. Depending on whether the current view is album or icon view, we set different actions.

## Parameters

<i>ids</i>	the selected items in the current view
<i>album</i>	the current album the AlbumIconView is displaying

**6.218.3.29 addSubMenu()**

```
void Digikam::ContextMenuHelper::addSubMenu (
    QMenu * subMenu )
```

Add a submenu to the parent context menu.

## Parameters

<i>subMenu</i>	the submenu to be added
----------------	-------------------------

**6.218.3.30 exec()**

```
QAction * Digikam::ContextMenuHelper::exec (
    const QPoint & pos,
    QAction *const at = nullptr )
```

Execute the registered parent menu and evaluate the triggered actions.

Always use this method instead the one from the parent menu. It will ensure that the signals are emitted and special cases are handled.

## Parameters

<i>pos</i>	position of the triggered action in the registered menu
<i>at</i>	the action that should be at the position pos

## Returns

the triggered action

**6.218.3.31 setAlbumModel()**

```
void Digikam::ContextMenuHelper::setAlbumModel (
    AbstractCheckableAlbumModel *const model )
```

Set an album model. The check/uncheck actions will operate directly on the model.

**6.218.3.32 setItemFilterModel()**

```
void Digikam::ContextMenuHelper::setItemFilterModel (
    ItemFilterModel *const model )
```

Set a filter model. Some of the group actions will operate directly on the model.

## 6.219 Digikam::ContextMenuHelper::Private Class Reference

### Public Member Functions

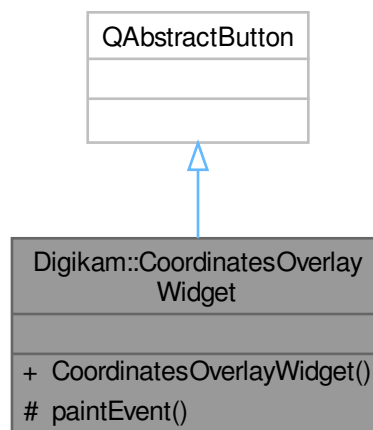
- QAction \* **copyFromMainCollection** (const QString &name) const
- QList< DPluginAction \* > **exportPluginActions** () const
- QModelIndex **indexForAlbumFromAction** (QObject \*const sender) const
- **Private** (ContextMenuHelper \*const qq)

### Public Attributes

- AbstractCheckableAlbumModel \* **albumModel** = nullptr
- QAction \* **gotoAlbumAction** = nullptr
- QAction \* **gotoDateAction** = nullptr
- ItemFilterModel \* **imageFilterModel** = nullptr
- QMap< QString, DServiceInfo > **newServicesMap**
- QMenu \* **parent** = nullptr
- ContextMenuHelper \* **q** = nullptr
- QMap< int, QAction \* > **queueActions**
- QList< qlonglong > **selectedIds**
- QList< QUrl > **selectedItems**
- QMap< QString, KService::Ptr > **servicesMap**
- QAction \* **setThumbnailAction** = nullptr
- KActionCollection \* **stdActionCollection** = nullptr

## 6.220 Digikam::CoordinatesOverlayWidget Class Reference

Inheritance diagram for Digikam::CoordinatesOverlayWidget:



**Public Member Functions**

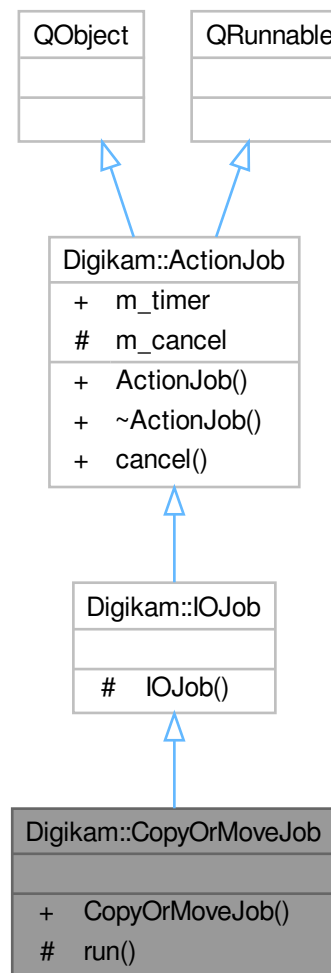
- **CoordinatesOverlayWidget** (QWidget \*const parent=nullptr)

**Protected Member Functions**

- void **paintEvent** (QPaintEvent \*) override

**6.221 Digikam::CopyOrMoveJob Class Reference**

Inheritance diagram for Digikam::CopyOrMoveJob:

**Public Member Functions**

- **CopyOrMoveJob** ([IOJobData](#) \*const data)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void [run](#) () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Signals inherited from [Digikam::IOJob](#)

- void [signalError](#) (const QString &errMsg)
- void [signalOneProcessed](#) (const QUrl &url)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.222 Digikam::CopyrightInfo Class Reference

### Public Member Functions

- bool [isNull](#) () const

### Public Attributes

- QString [extraValue](#)
- qlonglong [id](#) = -1
- QString [property](#)
- QString [value](#)



## 6.223 Digikam::CoreDB Class Reference

### Public Types

- enum **CopyrightPropertyUnique** { **PropertyUnique** , **PropertyExtraValueUnique** , **PropertyNoConstraint** }
- enum **ItemSortOrder** { **NoItemSorting** , **ByItemName** , **ByItemPath** , **ByItemDate** , **ByItemRating** }

### Public Member Functions

- int **addAlbum** (int albumRootId, const QString &relativePath, const QString &caption, const QDate &date, const QString &collection) const
- int **addAlbumRoot** (**CollectionLocation::Type** type, const QString &identifier, const QString &specificPath, const QString &label) const
- void **addImageMetadata** (qulonglong imageID, const QVariantList &infos, DatabaseFields::ImageMetadata fields=DatabaseFields::ImageMetadataAll)
- void **addImageRelation** (const **ImageRelation** &relation)
- void **addImageRelation** (qulonglong subjectId, qulonglong objectId, DatabaseRelation::Type type)
- void **addImageRelations** (const QList< qulonglong > &subjectIds, const QList< qulonglong > &objectIds, DatabaseRelation::Type type)
- void **addImageTagProperty** (const **ImageTagProperty** &property)
- void **addImageTagProperty** (qulonglong imageId, int tagId, const QString &property, const QString &value)
- qulonglong **addItem** (int albumID, const QString &name, DatabaseItem::Status status, DatabaseItem::Category category, const QDateTime &modificationDate, qulonglong fileSize, const QString &uniqueHash) const
- void **addItemInformation** (qulonglong imageID, const QVariantList &infos, DatabaseFields::ItemInformation fields=DatabaseFields::ItemInformationAll)
- void **addItemPosition** (qulonglong imageID, const QVariantList &infos, DatabaseFields::ItemPositions fields=DatabaseFields::ItemPositionsAll)
- void **addItemTag** (int albumID, const QString &name, int tagID)
- void **addItemTag** (qulonglong imageID, int tagID, bool newTag=false)
- int **addSearch** (DatabaseSearch::Type type, const QString &name, const QString &query) const
- int **addTag** (int parentTagID, const QString &name, const QString &iconKDE, qulonglong iconID) const
- void **addTagProperty** (const **TagProperty** &property)
- void **addTagProperty** (int tagId, const QString &property, const QString &value)
- void **addTagsToItems** (const QList< qulonglong > &imageIDs, const QList< int > &tagIDs)
- int **addToDownloadHistory** (const QString &identifier, const QString &name, qulonglong fileSize, const QDateTime &date) const
- void **addVideoMetadata** (qulonglong imageID, const QVariantList &infos, DatabaseFields::VideoMetadata fields=DatabaseFields::VideoMetadataAll)
- void **changeImageComment** (int commentId, qulonglong imageID, const QVariantList &infos, DatabaseFields::ItemComments fields=DatabaseFields::ItemCommentsAll)
- void **changeImageMetadata** (qulonglong imageID, const QVariantList &infos, DatabaseFields::ImageMetadata fields=DatabaseFields::ImageMetadataAll)
- void **changeItemInformation** (qulonglong imageID, const QVariantList &infos, DatabaseFields::ItemInformation fields=DatabaseFields::ItemInformationAll)
- void **changeItemPosition** (qulonglong imageID, const QVariantList &infos, DatabaseFields::ItemPositions fields=DatabaseFields::ItemPositionsAll)
- void **changeVideoMetadata** (qulonglong imageID, const QVariantList &infos, DatabaseFields::VideoMetadata fields=DatabaseFields::VideoMetadataAll)
- bool **copyAlbumProperties** (int srcAlbumID, int dstAlbumID) const
- void **copyImageAttributes** (qulonglong srcId, qulonglong destId)

- void [copyImageProperties](#) (qulonglong srcId, qulonglong dstId)
- void [copyImageTags](#) (qulonglong srcId, qulonglong dstId)
- qulonglong [copyItem](#) (int srcAlbumID, const QString &srcName, int dstAlbumID, const QString &dstName)
- [CoreDB](#) ([CoreDbBackend](#) \*const backend)
- QUuid [databaseUuid](#) ()
- void [deleteAlbum](#) (int albumID)
- void [deleteAlbumRoot](#) (int rootId)
- void [deleteItem](#) (int albumID, const QString &file)
- void [deleteItem](#) (qulonglong imageId)
- void [deleteObsoleteItem](#) (qulonglong imageId)
- void [deleteRemovedItems](#) ()
- void [deleteSearch](#) (int searchID)
- void [deleteSearches](#) (DatabaseSearch::Type type)
- void [deleteStaleAlbums](#) ()
- void [deleteTag](#) (int tagID)
- QList< qulonglong > [findByNameAndCreationDate](#) (const QString &fileName, const QDateTime &creationDate) const
- qulonglong [findImageId](#) (int albumID, const QString &name, DatabaseItem::Status status, DatabaseItem::Category category, qulonglong fileSize, const QString &uniqueHash) const
- int [findInDownloadHistory](#) (const QString &identifier, const QString &name, qulonglong fileSize, const QDateTime &date) const
- QList< int > [getAlbumAndSubalbumsForPath](#) (int albumRootId, const QString &relativePath) const
- QDateTime [getAlbumAverageDate](#) (int albumID) const
- int [getAlbumForPath](#) (int albumRootId, const QString &relativePath, bool create=true) const
- QDateTime [getAlbumHighestDate](#) (int albumID) const
- QDateTime [getAlbumLowestDate](#) (int albumID) const
- QDateTime [getAlbumModificationDate](#) (int albumID) const
- QMap< QString, QDateTime > [getAlbumModificationMap](#) (int albumRootId) const
- QString [getAlbumRelativePath](#) (int albumID) const
- int [getAlbumRootId](#) (int albumID) const
- QList< AlbumRootInfo > [getAlbumRoots](#) () const
- QList< AlbumShortInfo > [getAlbumShortInfos](#) () const
- QList< int > [getAlbumsOnAlbumRoot](#) (int albumRootId) const
- QVariantList [getAllCreationDates](#) () const
- QStringList [getAllImagePropertiesByName](#) (const QString &property) const
- QList< qulonglong > [getAllItems](#) () const
- QHash< qulonglong, QPair< int, int > > [getAllItemsWithAlbum](#) () const
- QString [getDatabaseEncoding](#) () const
- QStringList [getDirtyOrMissingFaceImageUrls](#) () const
- void [getFilterSettings](#) (QStringList \*imageFilter, QStringList \*videoFilter, QStringList \*audioFilter)
- qulonglong [getFirstItemWithFaceTag](#) (int tagId) const
- QMap< QString, int > [getFormatStatistics](#) () const
- QMap< QString, int > [getFormatStatistics](#) (DatabaseItem::Category category) const
- QList< ItemScanInfo > [getIdenticalFiles](#) (const QString &uniqueHash, qulonglong fileSize, qulonglong sourceId=-1) const
- QList< ItemScanInfo > [getIdenticalFiles](#) (qulonglong id) const
- void [getIgnoreDirectoryFilterSettings](#) (QStringList \*ignoreDirectoryFilter)
- qulonglong [getImageId](#) (int albumID, const QString &name) const
- QList< qulonglong > [getImageIds](#) (DatabaseItem::Status status) const
- QList< qulonglong > [getImageIds](#) (DatabaseItem::Status status, DatabaseItem::Category category) const
- QList< qulonglong > [getImageIds](#) (int albumID, const QString &name, DatabaseItem::Status status) const
- QList< qulonglong > [getImageIds](#) (int albumID, DatabaseItem::Status status, bool scanned=true) const
- QList< QVariant > [getImageIdsFromArea](#) (qreal lat1, qreal lat2, qreal lng1, qreal lng2, int sortMode, const QString &sortBy) const

- QVariantList [getImageMetadata](#) (qlonglong imageID, DatabaseFields::ImageMetadata metadata← Fields=DatabaseFields::ImageMetadataAll) const
- QString [getImageProperty](#) (qlonglong imageID, const QString &property) const
- QVariantList [getImagesFields](#) (qlonglong imageID, DatabaseFields::Images imagesFields) const
- QVector< QList< qlonglong > > [getImagesRelatedFrom](#) (const QList< qlonglong > &subjectIds, DatabaseRelation::Type type=DatabaseRelation::UndefinedType) const
- QList< qlonglong > [getImagesRelatedFrom](#) (qlonglong subjectId, DatabaseRelation::Type type=Database← Relation::UndefinedType) const
- QVector< QList< qlonglong > > [getImagesRelatingTo](#) (const QList< qlonglong > &objectIds, Database← Relation::Type type=DatabaseRelation::UndefinedType) const
- QList< qlonglong > [getImagesRelatingTo](#) (qlonglong objectId, DatabaseRelation::Type type=Database← Relation::UndefinedType) const
- QList< qlonglong > [getImagesWithImageTagProperty](#) (int tagId, const QString &property) const
- QList< qlonglong > [getImagesWithProperty](#) (const QString &property) const
- QList< [ImageTagProperty](#) > [getImageTagProperties](#) (qlonglong imageId, int tagId=-1) const
- QString [getImageUuid](#) (qlonglong imageId) const
- int [getItemAlbum](#) (qlonglong imageID) const
- QList< [CommentInfo](#) > [getItemComments](#) (qlonglong imageID) const
- QList< int > [getItemCommonTagIDs](#) (const QList< qlonglong > &imageIDList) const
- QList< [CopyrightInfo](#) > [getItemCopyright](#) (qlonglong imageID, const QString &property=QString()) const
- qlonglong [getItemFromAlbum](#) (int albumID, const QString &fileName) const
- [ImageHistoryEntry](#) [getItemHistory](#) (qlonglong imageId) const
- QMap< qlonglong, QString > [getItemIDsAndURLsInAlbum](#) (int albumID) const
- QList< qlonglong > [getItemIDsInAlbum](#) (int albumID) const
- QList< qlonglong > [getItemIDsInTag](#) (int tagID, bool recursive=false) const
- QVariantList [getItemInformation](#) (qlonglong imageID, DatabaseFields::ItemInformation infoFields=Database← Fields::ItemInformationAll) const
- QString [getItemName](#) (qlonglong imageID) const
- QStringList [getItemNamesInAlbum](#) (int albumID, bool recursive=false) const
- QVariantList [getItemPosition](#) (qlonglong imageID, DatabaseFields::ItemPositions positionFields=Database← Fields::ItemPositionsAll) const
- QVariantList [getItemPositions](#) (const QList< qlonglong > &imageIDs, DatabaseFields::ItemPositions fields) const
- [ItemScanInfo](#) [getItemScanInfo](#) (qlonglong imageID) const
- QList< [ItemScanInfo](#) > [getItemScanInfos](#) (int albumID) const
- QList< qlonglong > [getItemsForUuid](#) (const QString &uuid) const
- [ItemShortInfo](#) [getItemShortInfo](#) (int albumRootId, const QString &relativePath, const QString &name) const
- [ItemShortInfo](#) [getItemShortInfo](#) (qlonglong imageID) const
- QVector< QList< int > > [getItemsTagIDs](#) (const QList< qlonglong > &imageIds) const
- QStringList [getItemsURLsWithTag](#) (int tagId) const
- QList< int > [getItemTagIDs](#) (qlonglong imageID) const
- QStringList [getItemTagNames](#) (qlonglong imageID) const
- QStringList [getItemURLsInAlbum](#) (int albumID, ItemSortOrder order=NoItemSorting) const
- QStringList [getItemURLsInTag](#) (int tagID, bool recursive=false) const
- QStringList [getListFromImageMetadata](#) (DatabaseFields::ImageMetadata field) const
- QPair< int, int > [getNumberOfAllItemsAndAlbums](#) (int albumID) const
- QHash< int, int > [getNumberOfImagesInAlbums](#) () const
- QHash< int, int > [getNumberOfImagesInTagProperties](#) (const QString &property) const
- int [getNumberOfImagesInTagProperties](#) (int tagId, const QString &property) const
- QHash< int, int > [getNumberOfImagesInTags](#) () const
- int [getNumberOfItemsInAlbum](#) (int albumID) const
- QList< qlonglong > [getObsoleteItemIds](#) () const
- QList< qlonglong > [getOneRelatedImageEach](#) (const QList< qlonglong > &ids, DatabaseRelation::Type type=DatabaseRelation::UndefinedType) const
- QList< int > [getRecentlyAssignedTags](#) () const

- `QList< qlonglong > getRelatedImagesToByType (DatabaseRelation::Type type) const`
- `QList< QPair< qlonglong, qlonglong > > getRelationCloud (qlonglong imageId, DatabaseRelation::Type type=DatabaseRelation::UndefinedType) const`
- `SearchInfo getSearchInfo (int searchId) const`
- `QString getSearchQuery (int searchId) const`
- `QString getSetting (const QString &keyword) const`
- `QList< int > getTagIdsWithProperties (qlonglong imageId) const`
- `TagInfo getTagInfo (int tagId) const`
- `QList< TagProperty > getTagProperties () const`
- `QList< TagProperty > getTagProperties (const QString &property) const`
- `QList< TagProperty > getTagProperties (int tagID) const`
- `QList< TagShortInfo > getTagShortInfos () const`
- `QList< int > getTagsWithProperty (const QString &property) const`
- `int getUniqueHashVersion () const`
- `void getUserFilterSettings (QString *imageFilterString, QString *videoFilterString, QString *audioFilterString)`
- `void getUserIgnoreDirectoryFilterSettings (QString *ignoreDirectoryFilterString)`
- `QVariantList getVideoMetadata (qlonglong imageID, DatabaseFields::VideoMetadata metadata=DatabaseFields::VideoMetadataAll) const`
- `bool hasImageHistory (qlonglong imageId) const`
- `bool hasImagesRelatedFrom (qlonglong subjectId, DatabaseRelation::Type type=DatabaseRelation::UndefinedType) const`
- `bool hasImagesRelatingTo (qlonglong objectId, DatabaseRelation::Type type=DatabaseRelation::UndefinedType) const`
- `bool hasTags (const QList< qlonglong > &imageIDList) const`
- `bool integrityCheck () const`
- `void makeStaleAlbum (int albumID)`
- `void migrateAlbumRoot (int rootId, const QString &identifier)`
- `void moveItem (int srcAlbumID, const QString &srcName, int dstAlbumID, const QString &dstName)`
- `void removeAllImageComments (qlonglong imageID)`
- `void removeAllImageProperties (qlonglong imageID)`
- `QList< qlonglong > removeAllImageRelationsFrom (qlonglong subjectId, DatabaseRelation::Type type) const`
- `QList< qlonglong > removeAllImageRelationsTo (qlonglong objectId, DatabaseRelation::Type type) const`
- `void removeAllItemCopyrightProperties (qlonglong imageID)`
- `void removeImageComment (int commentId, qlonglong imageID)`
- `void removeImageProperty (qlonglong imageID, const QString &property)`
- `void removeImagePropertyByName (const QString &property)`
- `void removeImageRelation (const ImageRelation &relation)`
- `void removeImageRelation (qlonglong subjectId, qlonglong objectId, DatabaseRelation::Type type)`
- `void removeImageTagProperties (qlonglong imageId, int tagId=-1, const QString &property=QString(), const QString &value=QString())`
- `void removeItemAllTags (qlonglong imageID, const QList< int > &currentTagIds)`
- `void removeItemCopyrightProperties (qlonglong imageID, const QString &property=QString(), const QString &extraValue=QString(), const QString &value=QString())`
- `void removeItemPosition (qlonglong imageId)`
- `void removeItemPositionAltitude (qlonglong imageId)`
- `void removeItems (const QList< qlonglong > &itemIDs, const QList< int > &albumIDs=QList< int >())`
- `void removeItemsFromAlbum (int albumID, const QList< qlonglong > &ids_forInformation=QList< qlonglong >())`
- `void removeItemsPermanently (const QList< qlonglong > &itemIDs, const QList< int > &albumIDs=QList< int >())`
- `void removeItemTag (qlonglong imageID, int tagID)`
- `void removeTagProperties (int tagId, const QString &property=QString(), const QString &value=QString())`
- `void removeTagsFromItems (const QList< qlonglong > &imageIDs, const QList< int > &tagIDs)`

- void [renameAlbum](#) (int albumID, int newAlbumRootId, const QString &newRelativePath)
- void [renameltem](#) (qulonglong imageID, const QString &newName)
- AlbumInfo::List [scanAlbums](#) () const
- SearchInfo::List [scanSearches](#) () const
- TagInfo::List [scanTags](#) () const
- void [setAlbumCaption](#) (int albumID, const QString &caption)
- void [setAlbumCategory](#) (int albumID, const QString &category)
- void [setAlbumDate](#) (int albumID, const QDate &date)
- void [setAlbumIcon](#) (int albumID, qulonglong iconID)
- void [setAlbumModificationDate](#) (int albumID, const QDateTime &modificationDate)
- void [setAlbumRootCaseSensitivity](#) (int rootId, [CollectionLocation::CaseSensitivity](#) caseSensitivity)
- void [setAlbumRootLabel](#) (int rootId, const QString &newLabel)
- void [setAlbumRootPath](#) (int rootId, const QString &newPath)
- void [setAlbumRootType](#) (int rootId, [CollectionLocation::Type](#) newType)
- void [setFilterSettings](#) (const QStringList &imageFilter, const QStringList &videoFilter, const QStringList &audioFilter)
- void [setIgnoreDirectoryFilterSettings](#) (const QStringList &ignoreDirectoryFilter)
- int [setImageComment](#) (qulonglong imageID, const QString &comment, DatabaseComment::Type type, const QString &language=QString(), const QString &author=QString(), const QDateTime &date=QDateTime()) const
- void [setImageProperty](#) (qulonglong imageID, const QString &property, const QString &value)
- void [setImageUuid](#) (qulonglong imageId, const QString &uuid)
- void [setItemAlbum](#) (qulonglong imageID, qulonglong albumId)
- void [setItemCopyrightProperty](#) (qulonglong imageID, const QString &property, const QString &value, const QString &extraValue=QString(), CopyrightPropertyUnique uniqueness=PropertyUnique)
- void [setItemHistory](#) (qulonglong imageId, const QString &history)
- void [setItemManualOrder](#) (qulonglong imageID, qulonglong value)
- void [setItemModificationDate](#) (qulonglong imageID, const QDateTime &modificationDate)
- void [setItemStatus](#) (qulonglong imageID, DatabaseItem::Status status)
- void [setSetting](#) (const QString &keyword, const QString &value)
- void [setTagIcon](#) (int tagID, const QString &iconKDE, qulonglong iconID)
- void [setTagName](#) (int tagID, const QString &name)
- void [setTagParentID](#) (int tagID, int newParentTagID)
- void [setUniqueHashVersion](#) (int version)
- void [setUserFilterSettings](#) (const QStringList &imageFilter, const QStringList &videoFilter, const QStringList &audioFilter)
- void [setUserIgnoreDirectoryFilterSettings](#) (const QStringList &ignoreDirectoryFilters)
- void [updateItem](#) (qulonglong imageID, DatabaseItem::Category category, const QDateTime &modificationDate, qulonglong fileSize, const QString &uniqueHash)
- void [updateSearch](#) (int searchID, DatabaseSearch::Type type, const QString &name, const QString &query)
- void [vacuum](#) ()
- [~CoreDB](#) ()

### Static Public Member Functions

- static void [addBoundValuePlaceholders](#) (QString &query, int count)
- static QStringList [imageCommentsFieldList](#) (DatabaseFields::ItemComments fields)
- static QStringList [imageInformationFieldList](#) (DatabaseFields::ItemInformation fields)
- static QStringList [imageMetadataFieldList](#) (DatabaseFields::ImageMetadata fields)
- static QStringList [imagePositionsFieldList](#) (DatabaseFields::ItemPositions fields)
- static QStringList [imagesFieldList](#) (DatabaseFields::Images fields)
- static QStringList [videoMetadataFieldList](#) (DatabaseFields::VideoMetadata fields)

## Protected Member Functions

- QVector< QList< qlonglong > > **getRelatedImages** (QList< qlonglong > ids, bool fromOrTo, Database↔Relation::Type type, bool boolean) const
- QList< qlonglong > **getRelatedImages** (qlonglong id, bool fromOrTo, DatabaseRelation::Type type, bool boolean) const

## Friends

- class **Digikam::CoreDbAccess**

## 6.223.1 Constructor & Destructor Documentation

### 6.223.1.1 CoreDB()

```
Digikam::CoreDB::CoreDB (
    CoreDbBackend *const backend ) [explicit]
```

Constructor

### 6.223.1.2 ~CoreDB()

```
Digikam::CoreDB::~CoreDB ( )
```

Destructor

## 6.223.2 Member Function Documentation

### 6.223.2.1 addAlbum()

```
int Digikam::CoreDB::addAlbum (
    int albumRootId,
    const QString & relativePath,
    const QString & caption,
    const QDate & date,
    const QString & collection ) const
```

Add a new album to the database with the given attributes

#### Parameters

<i>album↔RootId</i>	id of the album root of the new album
<i>relativePath</i>	url of the album
<i>caption</i>	the album caption
<i>date</i>	the date for the album
<i>collection</i>	the album collection

**Returns**

the id of the album added or -1 if it failed

**6.223.2.2 addAlbumRoot()**

```
int Digikam::CoreDB::addAlbumRoot (
    CollectionLocation::Type type,
    const QString & identifier,
    const QString & specificPath,
    const QString & label ) const
```

Add a new album to the database with the given attributes

**Parameters**

<i>type</i>	The type of the album root
<i>identifier</i>	The album root identifier
<i>specificPath</i>	The path specific to volume
<i>label</i>	An (optional) user-visible label

**Returns**

the album root id of the newly created root

**6.223.2.3 addImageMetadata()**

```
void Digikam::CoreDB::addImageMetadata (
    qulonglong imageID,
    const QVariantList & infos,
    DatabaseFields::ImageMetadata fields = DatabaseFields::ImageMetadataAll )
```

Add (or replace) the ImageMetadata of the specified item. If there is already an entry, it will be discarded. The QVariantList shall have at most 16 entries, of types as defined in the DBSCHEMA and in metadatainfo.h, in this order:

0) String make 1) String model 2) String lens 3) Double aperture 4) Double focalLength 5) Double focalLength35 6) Double exposureTime 7) Int exposureProgram 8) Int exposureMode 9) Int sensitivity 10) Int flash 11) Int WhiteBalance 12) Int WhiteBalanceColorTemperature 13) Int meteringMode 14) Double subjectDistance 15) Double subjectDistanceCategory

**Note**

: you can leave out entries from this list. Indicate the values that you have passed in the ImageMetadata flag in the third parameters.

**6.223.2.4 addImageRelation()**

```
void Digikam::CoreDB::addImageRelation (
    qulonglong subjectId,
    qulonglong objectId,
    DatabaseRelation::Type type )
```

Adds an image relation entry.

### 6.223.2.5 addImageRelations()

```
void Digikam::CoreDB::addImageRelations (
    const QList< qlonglong > & subjectIds,
    const QList< qlonglong > & objectIds,
    DatabaseRelation::Type type )
```

This method requires two lists of same size and will add list1[0]->list2[0],...,list1[n]->list2[n]

### 6.223.2.6 addImageTagProperty()

```
void Digikam::CoreDB::addImageTagProperty (
    qlonglong imageId,
    int tagId,
    const QString & property,
    const QString & value )
```

Adds a tag property. Note that this never replaces existing entries. It is also all right to add multiple entries for a tag with the same property. To replace an existing entry, remove the entry before.

### 6.223.2.7 addItem()

```
qlonglong Digikam::CoreDB::addItem (
    int albumID,
    const QString & name,
    DatabaseItem::Status status,
    DatabaseItem::Category category,
    const QDateTime & modificationDate,
    qlonglong fileSize,
    const QString & uniqueHash ) const
```

Put a new item in the database or replace an existing one.

#### Returns

the id of item added or -1 if it fails

### 6.223.2.8 addItemInformation()

```
void Digikam::CoreDB::addItemInformation (
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::ItemInformation fields = DatabaseFields::ItemInformationAll )
```

Add (or replace) the ItemInformation of the specified item. If there is already an entry, it will be discarded. The QVariantList shall have 9 entries, of types in this order:

0) Int rating 1) DateTime\* creationDate 2) DateTime\* digitizationDate 3) Int orientation 4) Int width 5) Int height 6) String format 7) Int colorDepth 8) Int colorModel

#### Note

: you can provide the date also as a string in the format Qt::IsoDate. You can leave out entries from this list, which will then be filled with null values. Indicate the values that you have passed in the ItemInformation flag in the third parameters.



**6.223.2.9 addItemPosition()**

```
void Digikam::CoreDB::addItemPosition (
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::ItemPositions fields = DatabaseFields::ItemPositionsAll )
```

Add (or replace) the [ItemPosition](#) of the specified item. If there is already an entry, it will be discarded. The QVariantList shall have at most 10 entries, of types in this order:

0) String Latitude 1) Double LatitudeNumber 2) String Longitude 3) Double LongitudeNumber 4) Double Altitude 5) Double Orientation 6) Double Tilt 7) Double Roll 8) Double Accuracy 9) String Description

**Note**

: you can leave out entries from this list. Indicate the values that you have passed in the [ItemInfo](#) flag in the third parameters.

**6.223.2.10 addItemTag() [1/2]**

```
void Digikam::CoreDB::addItemTag (
    int albumID,
    const QString & name,
    int tagID )
```

Add a tag for the item

**Parameters**

<i>albumID</i>	the albumID of the item
<i>name</i>	the name of the item
<i>tagID</i>	the tagID for the tag

**6.223.2.11 addItemTag() [2/2]**

```
void Digikam::CoreDB::addItemTag (
    qlonglong imageID,
    int tagID,
    bool newTag = false )
```

Add a tag for the item

**Parameters**

<i>imageID</i>	the ID of the item
<i>tagID</i>	the tagID for the tag
<i>newTag</i>	add to last assigned tag list

**6.223.2.12 addSearch()**

```
int Digikam::CoreDB::addSearch (
    DatabaseSearch::Type type,
    const QString & name,
    const QString & query ) const
```

Add a new search to the database with the given attributes

**Parameters**

<i>type</i>	search type
<i>name</i>	name of the search
<i>query</i>	search query to use

**Returns**

the id of the album added or -1 if it failed

**6.223.2.13 addTag()**

```
int Digikam::CoreDB::addTag (
    int parentTagID,
    const QString & name,
    const QString & iconKDE,
    qlonglong iconID ) const
```

Adds a new tag to the database with given name, icon and parent id.

**Parameters**

<i>parentTagID</i>	the id of the tag which will become the new tags parent
<i>name</i>	the name of the tag
<i>iconKDE</i>	the name of the icon file (this is filename which kde iconloader can load up)
<i>iconID</i>	the id of the icon file Note: if the iconKDE parameter is empty, then the iconID parameter is used

**Returns**

the id of the tag added or -1 if it failed

**6.223.2.14 addTagProperty()**

```
void Digikam::CoreDB::addTagProperty (
    int tagId,
    const QString & property,
    const QString & value )
```

Adds a tag property. Note that this never replaces existing entries. It is also all right to add multiple entries for a tag with the same property. To replace an existing entry, remove the entry before.

### 6.223.2.15 addTagsToItems()

```
void Digikam::CoreDB::addTagsToItems (
    const QList< qlonglong > & imageIDs,
    const QList< int > & tagIDs )
```

Add each tag of a list of tags to each member of a list of items.

### 6.223.2.16 addToDownloadHistory()

```
int Digikam::CoreDB::addToDownloadHistory (
    const QString & identifier,
    const QString & name,
    qlonglong fileSize,
    const QDateTime & date ) const
```

Add the specified fingerprint to the download history table. Returns the id of the entry.

### 6.223.2.17 addVideoMetadata()

```
void Digikam::CoreDB::addVideoMetadata (
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::VideoMetadata fields = DatabaseFields::VideoMetadataAll )
```

Add (or replace) the VideoMetadata of the specified item. If there is already an entry, it will be discarded. The QVariantList shall have 8 entries, of types in this order:

- 0) String AspectRatio
- 1) String AudioBitRate
- 2) String AudioChannelType
- 3) String AudioCodec
- 4) String Duration
- 5) String FrameRate
- 6) String VideoCodec

#### Note

: you can leave out entries from this list, which will then be filled with null values. Indicate the values that you have passed in the VideoMetadata flag in the third parameters.

### 6.223.2.18 changeImageComment()

```
void Digikam::CoreDB::changeImageComment (
    int commentId,
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::ItemComments fields = DatabaseFields::ItemCommentsAll )
```

Changes the properties of a comment. The QVariantList shall have at most 5 entries, of types in this order:

- 0) Int Type
- 1) String Language
- 2) String Author
- 3) DateTime Date
- 4) String Comment

### 6.223.2.19 changelImageMetadata()

```
void Digikam::CoreDB::changeImageMetadata (
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::ImageMetadata fields = DatabaseFields::ImageMetadataAll )
```

Change the indicated fields of the image information for the specified item. This method does nothing if the item does not yet have an entry in the ItemInformation table. The parameters are as for the method above.

### 6.223.2.20 changelItemInformation()

```
void Digikam::CoreDB::changeItemInformation (
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::ItemInformation fields = DatabaseFields::ItemInformationAll )
```

Change the indicated fields of the image information for the specified item. Fields not indicated by the fields parameter will not be touched. This method does nothing if the item does not yet have an entry in the ItemInformation table. The parameters are as for the method above.

### 6.223.2.21 changelItemPosition()

```
void Digikam::CoreDB::changeItemPosition (
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::ItemPositions fields = DatabaseFields::ItemPositionsAll )
```

Change the indicated fields of the image information for the specified item. This method does nothing if the item does not yet have an entry in the ItemInformation table. The parameters are as for the method above.

### 6.223.2.22 changeVideoMetadata()

```
void Digikam::CoreDB::changeVideoMetadata (
    qlonglong imageID,
    const QVariantList & infos,
    DatabaseFields::VideoMetadata fields = DatabaseFields::VideoMetadataAll )
```

Change the indicated fields of the video information for the specified item. This method does nothing if the item does not yet have an entry in the ItemInformation table. The parameters are as for the method above.

### 6.223.2.23 copyAlbumProperties()

```
bool Digikam::CoreDB::copyAlbumProperties (
    int srcAlbumID,
    int dstAlbumID ) const
```

Copy the properties of the given srcAlbum to the dstAlbum. Both albums must exist.

#### Returns

true if the operations succeeds

**6.223.2.24 copyImageAttributes()**

```
void Digikam::CoreDB::copyImageAttributes (
    qlonglong srcId,
    qlonglong destId )
```

Copies all image-specific information, in all tables, from image srcId to destId.

**6.223.2.25 copyImageProperties()**

```
void Digikam::CoreDB::copyImageProperties (
    qlonglong srcId,
    qlonglong dstId )
```

Copies all entries in the ImageProperties table.

**6.223.2.26 copyImageTags()**

```
void Digikam::CoreDB::copyImageTags (
    qlonglong srcId,
    qlonglong dstId )
```

Copies all entries in the ImageTags table.

**6.223.2.27 copyItem()**

```
qlonglong Digikam::CoreDB::copyItem (
    int srcAlbumID,
    const QString & srcName,
    int dstAlbumID,
    const QString & dstName )
```

Copy the attributes of an item to a different item. Useful when say a file is copied. The operation fails (returns -1) if src and dest are identical.

**Parameters**

<i>srcAlbumID</i>	the id of the source album
<i>dstAlbumID</i>	the id of the destination album
<i>srcName</i>	the name of the source file
<i>dstName</i>	the name of the destination file

**Returns**

the id of item added or -1 if it fails

**6.223.2.28 databaseUuid()**

```
QUuid Digikam::CoreDB::databaseUuid ( )
```

Returns a UUID for the database file. This UUID is kept stable over schema updates.

**6.223.2.29 deleteAlbum()**

```
void Digikam::CoreDB::deleteAlbum (
    int albumID )
```

Deletes an album from the database. This will not delete the subalbums of the album.

**Parameters**

<i>albumID</i>	the id of the album
----------------	---------------------

**6.223.2.30 deleteAlbumRoot()**

```
void Digikam::CoreDB::deleteAlbumRoot (
    int rootId )
```

Deletes an album root from the database.

**Parameters**

<i>rootId</i>	the id of the album root
---------------	--------------------------

**6.223.2.31 deleteItem() [1/2]**

```
void Digikam::CoreDB::deleteItem (
    int albumID,
    const QString & file )
```

Deletes an item from the database.

**Parameters**

<i>albumID</i>	The id of the album.
<i>file</i>	The filename of the file to delete.

**6.223.2.32 deleteItem() [2/2]**

```
void Digikam::CoreDB::deleteItem (
    qulonglong imageId )
```

Deletes an item from the database if it does not belong to an album. This method can only be used if the album of the image is null!

**Parameters**

<i>imageId</i>	The id of the image.
----------------	----------------------

### 6.223.2.33 deleteObsoleteItem()

```
void Digikam::CoreDB::deleteObsoleteItem (
    qlonglong imageId )
```

Deletes an item from the database without checking the album.

#### Parameters

<i>imageId</i>	The id of the image.
----------------	----------------------

### 6.223.2.34 deleteRemovedItems()

```
void Digikam::CoreDB::deleteRemovedItems ( )
```

Delete all items from the database that are marked as removed.

#### Warning

: Use with care!

### 6.223.2.35 deleteSearch()

```
void Digikam::CoreDB::deleteSearch (
    int searchID )
```

Delete a search from the database.

#### Parameters

<i>searchID</i>	the id of the search
-----------------	----------------------

### 6.223.2.36 deleteSearches()

```
void Digikam::CoreDB::deleteSearches (
    DatabaseSearch::Type type )
```

Delete all search with the given type

### 6.223.2.37 deleteStaleAlbums()

```
void Digikam::CoreDB::deleteStaleAlbums ( )
```

Deletes albums from the database that were previously removed with [makeStaleAlbum\(\)](#)

**6.223.2.38 deleteTag()**

```
void Digikam::CoreDB::deleteTag (
    int tagID )
```

Deletes a tag from the database. This will not delete the subtags of the tag.

**Parameters**

<i>tagID</i>	the id of the tag
--------------	-------------------

**6.223.2.39 findByNameAndCreationDate()**

```
QList< qlonglong > Digikam::CoreDB::findByNameAndCreationDate (
    const QString & fileName,
    const QDateTime & creationDate ) const
```

Returns all items with the given file name and creation date.

**6.223.2.40 findImageId()**

```
qlonglong Digikam::CoreDB::findImageId (
    int albumID,
    const QString & name,
    DatabaseItem::Status status,
    DatabaseItem::Category category,
    qlonglong fileSize,
    const QString & uniqueHash ) const
```

Find the imageId fitting to the information given for the item

**Parameters**

<i>albumID</i>	the albumID of the item (-1 means null)
<i>name</i>	the name of the item
<i>status</i>	the status of the item
<i>category</i>	the category of the item
<i>fileSize</i>	the file size
<i>uniqueHash</i>	the unique hash

**Returns**

the ImageId for the item, or -1 if no matching or more than one infos were found.

**6.223.2.41 findInDownloadHistory()**

```
int Digikam::CoreDB::findInDownloadHistory (
    const QString & identifier,
```



```
const QString & name,
qlonglong fileSize,
const QDateTime & date ) const
```

Search for the specified fingerprint in the download history table. Returns the id of the entry, or -1 if not found.

#### 6.223.2.42 getAlbumAndSubalbumsForPath()

```
QList< int > Digikam::CoreDB::getAlbumAndSubalbumsForPath (
    int albumRootId,
    const QString & relativePath ) const
```

Find out the album ids for a given relative path, including the subalbums.

##### Parameters

<i>albumRootId</i>	id of the album root of the album
<i>relativePath</i>	The path for which you want the albumIDs relative to the album root

##### Returns

a list of album ids. The list is empty if no albums are found.

#### 6.223.2.43 getAlbumAverageDate()

```
QDate Digikam::CoreDB::getAlbumAverageDate (
    int albumID ) const
```

Returns the average date of all images for that album.

##### Parameters

<i>albumID</i>	the id of the album to calculate
----------------	----------------------------------

##### Returns

the date.

#### 6.223.2.44 getAlbumForPath()

```
int Digikam::CoreDB::getAlbumForPath (
    int albumRootId,
    const QString & relativePath,
    bool create = true ) const
```

Find out the album for a given folder.

## Parameters

<i>album</i> ↔ <i>RootId</i>	id of the album root of the album
<i>relativePath</i>	The relative path for which you want the albumID relative to the album root
<i>create</i>	If true, an album is newly created if it does not yet exist. If false, -1 is returned if no album exists.

## Returns

The albumID for that folder, or -1 if it does not exist and create is false.

**6.223.2.45 getAlbumHighestDate()**

```
QDate Digikam::CoreDB::getAlbumHighestDate (
    int albumID ) const
```

Returns the highest/newest date of all images for that album.

## Parameters

<i>albumID</i>	the id of the album to calculate
----------------	----------------------------------

## Returns

the date.

**6.223.2.46 getAlbumLowestDate()**

```
QDate Digikam::CoreDB::getAlbumLowestDate (
    int albumID ) const
```

Returns the lowest/oldest date of all images for that album.

## Parameters

<i>albumID</i>	the id of the album to calculate
----------------	----------------------------------

## Returns

the date.

**6.223.2.47 getAlbumModificationDate()**

```
QDateTime Digikam::CoreDB::getAlbumModificationDate (
    int albumID ) const
```

Returns the QDateTime of the album modification date.

## Parameters

<i>albumID</i>	the id of the album
----------------	---------------------

**6.223.2.48 getAlbumModificationMap()**

```
QMap< QString, QDateTime > Digikam::CoreDB::getAlbumModificationMap (
    int albumRootId ) const
```

Returns a QMap with relative path and the album modification date.

## Parameters

<i>album↔ RootId</i>	id of the album root of the album
--------------------------	-----------------------------------

**6.223.2.49 getAlbumRelativePath()**

```
QString Digikam::CoreDB::getAlbumRelativePath (
    int albumID ) const
```

Given an albumid, this returns the relative path for that album (the path below the album root, starting with a slash)

## Parameters

<i>albumID</i>	the id of the album
----------------	---------------------

## Returns

the url of the album

**6.223.2.50 getAlbumRootId()**

```
int Digikam::CoreDB::getAlbumRootId (
    int albumID ) const
```

Given an albumid, this returns the album root id for that album

## Parameters

<i>albumID</i>	the id of the albumdb
----------------	-----------------------

## Returns

the id of the album root of this album

### 6.223.2.51 `getAlbumRoots()`

```
QList< AlbumRootInfo > Digikam::CoreDB::getAlbumRoots ( ) const
```

Returns all albums and their attributes in the database

#### Returns

a list of albums and their attributes

### 6.223.2.52 `getAlbumShortInfos()`

```
QList< AlbumShortInfo > Digikam::CoreDB::getAlbumShortInfos ( ) const
```

Returns all albums in the database with their albumRoot and ID, ordered by id.

### 6.223.2.53 `getAlbumsOnAlbumRoot()`

```
QList< int > Digikam::CoreDB::getAlbumsOnAlbumRoot (
    int albumRootId ) const
```

Find out all album ids of a given album root

#### Returns

a list of album ids.

### 6.223.2.54 `getAllCreationDates()`

```
QVariantList Digikam::CoreDB::getAllCreationDates ( ) const
```

Returns a QVariantList of creationDate of all items

### 6.223.2.55 `getAllItems()`

```
QList< qlonglong > Digikam::CoreDB::getAllItems ( ) const
```

Returns all ids of items in images table.

### 6.223.2.56 `getAllItemsWithAlbum()`

```
QHash< qlonglong, QPair< int, int > > Digikam::CoreDB::getAllItemsWithAlbum ( ) const
```

Returns all ids of items with album ids in images table. QPair.first == albumRootID QPair.second == albumID

### 6.223.2.57 `getDatabaseEncoding()`

```
QString Digikam::CoreDB::getDatabaseEncoding ( ) const
```

Returns database encoding. For SQLite should UTF-8. For MySQL like UTF8MB4.

### 6.223.2.58 `getDirtyOrMissingFacelImageUrls()`

```
QStringList Digikam::CoreDB::getDirtyOrMissingFaceImageUrls ( ) const
```

Returns a list of all images where the Faces have either not been detected yet, or is outdated because the file is identified as changed since the generation of the fingerprint. Return image ids or item URLs.

### 6.223.2.59 `getFilterSettings()`

```
void Digikam::CoreDB::getFilterSettings (
    QStringList * imageFilter,
    QStringList * videoFilter,
    QStringList * audioFilter )
```

Get the settings for the file name filters of this database. Returns a list with lowercase suffixes only, no wildcards added ("png", not "\*.png") Returned is a joint result of main and user settings. If you are not interested in a specific value, pass 0.

### 6.223.2.60 `getFirstItemWithFaceTag()`

```
qulonglong Digikam::CoreDB::getFirstItemWithFaceTag (
    int tagId ) const
```

Returns the first item that has a confirmed face with the tag.

### 6.223.2.61 `getFormatStatistics()`

```
QMap< QString, int > Digikam::CoreDB::getFormatStatistics ( ) const
```

Returns a QMap<QString,int> of ItemInformation.format corresponding to count of items with that format.

### 6.223.2.62 `getIdenticalFiles()`

```
QList< ItemScanInfo > Digikam::CoreDB::getIdenticalFiles (
    qulonglong id ) const
```

Find items that are, with reasonable certainty, identical to the file pointed to by id. Criteria: Unique Hash, file size and album non-null. The first variant will not return an [ItemScanInfo](#) for id. The second allows to pass one id as source id for exclusion from the list. If this is -1, no id is excluded.

### 6.223.2.63 `getImageId()`

```
qulonglong Digikam::CoreDB::getImageId (
    int albumID,
    const QString & name ) const
```

Get the imageId of the item

## Parameters

<i>albumID</i>	the albumID of the item
<i>name</i>	the name of the item

## Returns

the ImageId for the item, or -1 if it does not exist

**6.223.2.64 getMagelds() [1/4]**

```
QList< qlonglong > Digikam::CoreDB::getImageIds (
    DatabaseItem::Status status ) const
```

Returns all image ids with the given status.

## Parameters

<i>status</i>	The status.
---------------	-------------

## Returns

The ids of the images that have the given status.

**6.223.2.65 getMagelds() [2/4]**

```
QList< qlonglong > Digikam::CoreDB::getImageIds (
    DatabaseItem::Status status,
    DatabaseItem::Category category ) const
```

Returns all image ids with the given status and category.

## Parameters

<i>status</i>	The status.
<i>category</i>	The category.

## Returns

The ids of the images that have the given status.

**6.223.2.66 getMagelds() [3/4]**

```
QList< qlonglong > Digikam::CoreDB::getImageIds (
    int albumID,
```

```
const QString & name,  
DatabaseItem::Status status ) const
```

Get the imageId fitting to the information given for the item

## Parameters

<i>albumID</i>	the albumID of the item (-1 means NULL)
<i>name</i>	the name of the item
<i>status</i>	the status of the item

## Returns

the ImageIds for the item, or an empty list if there are no matching entries.

**6.223.2.67 getImagelds() [4/4]**

```
QList< qlonglong > Digikam::CoreDB::getImageIds (
    int albumID,
    DatabaseItem::Status status,
    bool scanned = true ) const
```

Get the imageld fitting to the information given for the item

## Parameters

<i>albumID</i>	the albumID of the item (-1 means NULL)
<i>status</i>	the status of the item
<i>scanned</i>	return scanned/unscanned items

## Returns

the ImageIds for the item, or an empty list if there are no matching entries.

**6.223.2.68 getImageMetadata()**

```
QVariantList Digikam::CoreDB::getImageMetadata (
    qlonglong imageID,
    DatabaseFields::ImageMetadata metadataFields = DatabaseFields::ImageMetadataAll )
const
```

Read image metadata. Parameters as above.

**6.223.2.69 getImageProperty()**

```
QString Digikam::CoreDB::getImageProperty (
    qlonglong imageID,
    const QString & property ) const
```

Returns the property with the specified name for the specified image



**6.223.2.70 getImagesFields()**

```
QVariantList Digikam::CoreDB::getImagesFields (
    qlonglong imageID,
    DatabaseFields::Images imagesFields ) const
```

Returns the requested fields from the Images table. Choose the fields with the mask. The fields will be returned in the following order and type: 0) Int Album 1) String Name 2) Int Status 3) Int Category 4) DateTime ModificationDate 5) int FileSize 6) String uniqueHash

**6.223.2.71 getImagesRelatedFrom()**

```
QList< qlonglong > Digikam::CoreDB::getImagesRelatedFrom (
    qlonglong subjectId,
    DatabaseRelation::Type type = DatabaseRelation::UndefinedType ) const
```

Retrieves all images that the given image is related to (retrieves objects, given image is subject) If type is given, filters by type, otherwise returns all types. "Get images related to from this"

**6.223.2.72 getImagesRelatingTo()**

```
QList< qlonglong > Digikam::CoreDB::getImagesRelatingTo (
    qlonglong objectId,
    DatabaseRelation::Type type = DatabaseRelation::UndefinedType ) const
```

Retrieves all images that relate to the given image (retrieves subject, given image is object) If type is given, filters by type, otherwise returns all types. "Get images this image is relating to"

**6.223.2.73 getImagesWithImageTagProperty()**

```
QList< qlonglong > Digikam::CoreDB::getImagesWithImageTagProperty (
    int tagId,
    const QString & property ) const
```

Returns all image ids that are associated to the tag with the given property.

**6.223.2.74 getImagesWithProperty()**

```
QList< qlonglong > Digikam::CoreDB::getImagesWithProperty (
    const QString & property ) const
```

Returns all image ids that are associated to the given property.

**6.223.2.75 getImageTagProperties()**

```
QList< ImageTagProperty > Digikam::CoreDB::getImageTagProperties (
    qlonglong imageId,
    int tagId = -1 ) const
```

Get the properties for the given image/tag pair. If the tagID is -1, returns the ImageTagProperties for all tagIDs of the given image.

### 6.223.2.76 getImageUuid()

```
QString Digikam::CoreDB::getImageUuid (
    qlonglong imageId ) const
```

Retrieves the image UUID

### 6.223.2.77 getItemAlbum()

```
int Digikam::CoreDB::getItemAlbum (
    qlonglong imageID ) const
```

Find the album of an item

#### Parameters

<i>imageID</i>	The ID of the item
----------------	--------------------

#### Returns

The ID of the [PA](#)lbum of the item, or -1 if not found

### 6.223.2.78 getItemComments()

```
QList< CommentInfo > Digikam::CoreDB::getItemComments (
    qlonglong imageID ) const
```

Retrieves all available comments for the specified item.

### 6.223.2.79 getItemCommonTagIDs()

```
QList< int > Digikam::CoreDB::getItemCommonTagIDs (
    const QList< qlonglong > & imageIDList ) const
```

Given a set of items (identified by their IDs), get a list of ID of all common tags

#### Parameters

<i>imageIDList</i>	a list of IDs of the items
--------------------	----------------------------

#### Returns

the list of common IDs of the given items

### 6.223.2.80 getItemCopyright()

```
QList< CopyrightInfo > Digikam::CoreDB::getItemCopyright (
```

```
qulonglong imageID,
const QString & property = QString() ) const
```

Returns the copyright properties of the specified image. If property is not null, only the given property is returned.

### 6.223.2.81 getItemFromAlbum()

```
qulonglong Digikam::CoreDB::getItemFromAlbum (
    int albumID,
    const QString & fileName ) const
```

Returns the id of the item with the given filename in the album with the given id.

#### Parameters

<i>albumID</i>	The albumId in which we search the item.
<i>fileName</i>	The name of the item file.

#### Returns

The item id or -1 if not existent.

### 6.223.2.82 getItemHistory()

```
ImageHistoryEntry Digikam::CoreDB::getItemHistory (
    qulonglong imageId ) const
```

Retrieves the history entry for the given image.

### 6.223.2.83 getItemIDsAndURLsInAlbum()

```
QMap< qulonglong, QString > Digikam::CoreDB::getItemIDsAndURLsInAlbum (
    int albumID ) const
```

Given a albumID, get a map of ids and urls of all items in the album

#### Note

: Uses the [CollectionManager](#)

#### Parameters

<i>albumID</i>	the id of the album
----------------	---------------------

#### Returns

a map of ids and urls for the items in the album. The urls are the absolute path of the items

**6.223.2.84 getItemIDsInAlbum()**

```
QList< qlonglong > Digikam::CoreDB::getItemIDsInAlbum (
    int albumID ) const
```

Given a albumID, get a list of lds of all items in the album

**Parameters**

<i>albumID</i>	the id of the album
----------------	---------------------

**Returns**

a list of lds for the items in the album.

**6.223.2.85 getItemIDsInTag()**

```
QList< qlonglong > Digikam::CoreDB::getItemIDsInTag (
    int tagID,
    bool recursive = false ) const
```

Given a tagID, get a list of lds of all items in the tag

**Parameters**

<i>tagID</i>	the id of the tag
<i>recursive</i>	perform a recursive folder hierarchy parsing

**Returns**

a list of lds for the items in the tag.

**6.223.2.86 getItemInformation()**

```
QVariantList Digikam::CoreDB::getItemInformation (
    qlonglong imageID,
    DatabaseFields::ItemInformation infoFields = DatabaseFields::ItemInformationAll )
const
```

Read image information. Parameters as above.

**6.223.2.87 getItemName()**

```
QString Digikam::CoreDB::getItemName (
    qlonglong imageID ) const
```

Retrieve the name of the item

## Parameters

<i>imageID</i>	The ID of the item
----------------	--------------------

## Returns

The name of the item, or a null string if not found

**6.223.2.88 getItemNamesInAlbum()**

```
QStringList Digikam::CoreDB::getItemNamesInAlbum (
    int albumID,
    bool recursive = false ) const
```

Returns all items for a given albumid. This is used to verify if all items on disk are consistent with the database in the [CollectionScanner](#) class.

## Parameters

<i>albumID</i>	The albumID for which you want all items.
<i>recursive</i>	perform a recursive folder hierarchy parsing

## Returns

It returns a QStringList with the filenames.

**6.223.2.89 getItemPosition()**

```
QVariantList Digikam::CoreDB::getItemPosition (
    qlonglong imageID,
    DatabaseFields::ItemPositions positionFields = DatabaseFields::ItemPositionsAll )
const
```

Read image metadata. Parameters as above.

**6.223.2.90 getItemScanInfo()**

```
ItemScanInfo Digikam::CoreDB::getItemScanInfo (
    qlonglong imageID ) const
```

Get scan info from the image ID

**6.223.2.91 getItemScanInfos()**

```
QList< ItemScanInfo > Digikam::CoreDB::getItemScanInfos (
    int albumID ) const
```

Returns an [ItemScanInfo](#) object for each item in the album with the specified album id.

**6.223.2.92 getItemsForUuid()**

```
QList< qlonglong > Digikam::CoreDB::getItemsForUuid (
    const QString & uuid ) const
```

Retrieves the images with the given UUID

**6.223.2.93 getItemShortInfo() [1/2]**

```
ItemShortInfo Digikam::CoreDB::getItemShortInfo (
    int albumRootId,
    const QString & relativePath,
    const QString & name ) const
```

Get item and album if from albumRootId, album path and file name.

**6.223.2.94 getItemShortInfo() [2/2]**

```
ItemShortInfo Digikam::CoreDB::getItemShortInfo (
    qlonglong imageID ) const
```

Get item and album info from the image ID

**6.223.2.95 getItemsTagIDs()**

```
QVector< QList< int > > Digikam::CoreDB::getItemsTagIDs (
    const QList< qlonglong > & imageIds ) const
```

For a list of items, return the tag ids associated with the item. Amounts to calling getItemTagIDs for each id in imageIds, but is optimized.

**6.223.2.96 getItemsURLsWithTag()**

```
QStringList Digikam::CoreDB::getItemsURLsWithTag (
    int tagId ) const
```

Returns a list of all images where tagId is assigned Return item URLs.

**6.223.2.97 getItemTagIDs()**

```
QList< int > Digikam::CoreDB::getItemTagIDs (
    qlonglong imageID ) const
```

Get a list of IDs of all the tags for the item

**Parameters**

<i>imageID</i>	the ID of the item
----------------	--------------------

**Returns**

the list of IDs of all tags for the item

**6.223.2.98 getItemTagNames()**

```
QStringList Digikam::CoreDB::getItemTagNames (
    qlonglong imageID ) const
```

Get a list of names of all the tags for the item

**Parameters**

<i>imageID</i>	the ID of the item
----------------	--------------------

**Returns**

the list of names of all tags for the item

**6.223.2.99 getItemURLsInAlbum()**

```
QStringList Digikam::CoreDB::getItemURLsInAlbum (
    int albumID,
    ItemSortOrder order = NoItemSorting ) const
```

Given a albumID, get a list of the url of all items in the album

**Note**

: Uses the [CollectionManager](#)

**Parameters**

<i>albumID</i>	the id of the album
<i>order</i>	order for the returned items to use

**Returns**

a list of urls for the items in the album. The urls are the absolute path of the items

**6.223.2.100 getItemURLsInTag()**

```
QStringList Digikam::CoreDB::getItemURLsInTag (
    int tagID,
    bool recursive = false ) const
```

Given a tagid, get a list of the url of all items in the tag

**Note**

: Uses the [CollectionManager](#)

**Parameters**

<i>tagID</i>	the id of the tag
<i>recursive</i>	perform a recursive folder hierarchy parsing

**Returns**

a list of urls for the items in the tag. The urls are the absolute path of the items

**6.223.2.101 getListFromImageMetadata()**

```
QStringList Digikam::CoreDB::getListFromImageMetadata (
    DatabaseFields::ImageMetadata field ) const
```

Return a list from a field from imageMetadata

**6.223.2.102 getNumberOfAllItemsAndAlbums()**

```
QPair< int, int > Digikam::CoreDB::getNumberOfAllItemsAndAlbums (
    int albumID ) const
```

Returns the QPair<int, int> of all items (first) and albums (second) as a counter in the album.

**Parameters**

<i>albumID</i>	the id of the album
----------------	---------------------

**6.223.2.103 getNumberOfImagesInAlbums()**

```
QHash< int, int > Digikam::CoreDB::getNumberOfImagesInAlbums ( ) const
```

Returns a QHash<int, int> of album id -> count of items in the album

**6.223.2.104 getNumberOfImagesInTagProperties() [1/2]**

```
QHash< int, int > Digikam::CoreDB::getNumberOfImagesInTagProperties (
    const QString & property ) const
```

Returns a QHash<int, int> of tag id -> count of items with the given tag property



**6.223.2.105 getNumberOfImagesInTagProperties()** [2/2]

```
int Digikam::CoreDB::getNumberOfImagesInTagProperties (
    int tagId,
    const QString & property ) const
```

Returns the count of images that have a tag property for the given tag.

**6.223.2.106 getNumberOfImagesInTags()**

```
QHash< int, int > Digikam::CoreDB::getNumberOfImagesInTags ( ) const
```

Returns a QHash<int, int> of tag id -> count of items with the tag

**6.223.2.107 getNumberOfItemsInAlbum()**

```
int Digikam::CoreDB::getNumberOfItemsInAlbum (
    int albumID ) const
```

Returns the number of items in the album.

**Parameters**

<i>albumID</i>	the id of the album
----------------	---------------------

**6.223.2.108 getObsoleteItemIds()**

```
QList< qlonglong > Digikam::CoreDB::getObsoleteItemIds ( ) const
```

Get obsolete item ids.

**6.223.2.109 getOneRelatedImageEach()**

```
QList< qlonglong > Digikam::CoreDB::getOneRelatedImageEach (
    const QList< qlonglong > & ids,
    DatabaseRelation::Type type = DatabaseRelation::UndefinedType ) const
```

For each of the given ids, find one single related image (direction does not matter). Ids are unique in the returned list, and do not correspond by index to the given list.

**6.223.2.110 getRecentlyAssignedTags()**

```
QList< int > Digikam::CoreDB::getRecentlyAssignedTags ( ) const
```

Get a list of recently assigned tags (only last 6 tags are listed)

**Returns**

the list of recently assigned tags

### 6.223.2.111 getRelatedImagesToByType()

```
QList< qlonglong > Digikam::CoreDB::getRelatedImagesToByType (
    DatabaseRelation::Type type ) const
```

Retrieves all images that related to (retrieves objects) by given type.

### 6.223.2.112 getRelationCloud()

```
QList< QPair< qlonglong, qlonglong > > Digikam::CoreDB::getRelationCloud (
    qlonglong imageId,
    DatabaseRelation::Type type = DatabaseRelation::UndefinedType ) const
```

For the given image id, retrieves all relations of all related images: Each pair (a,b) means "a is related to b". Each a and b in the list will have a direct or indirect relation to the initial imageId. If type is given, filters by type, otherwise returns all types.

### 6.223.2.113 getSearchInfo()

```
SearchInfo Digikam::CoreDB::getSearchInfo (
    int searchId ) const
```

Get information about the specified search

### 6.223.2.114 getSearchQuery()

```
QString Digikam::CoreDB::getSearchQuery (
    int searchId ) const
```

Get the query for the search specified by its id

### 6.223.2.115 getSetting()

```
QString Digikam::CoreDB::getSetting (
    const QString & keyword ) const
```

This function returns the value which is stored in the database (table Settings).

#### Parameters

<i>keyword</i>	The keyword for which the value has to be returned.
----------------	---

#### Returns

The values which belongs to the keyword, or a null string if no value is set.

**6.223.2.116 getTagIdsWithProperties()**

```
QList< int > Digikam::CoreDB::getTagIdsWithProperties (
    qlonglong imageId ) const
```

Get all tagIds for which ImageTagProperties exist for the given image.

**6.223.2.117 getTagProperties() [1/3]**

```
QList< TagProperty > Digikam::CoreDB::getTagProperties ( ) const
```

Returns the list of all tag properties (ordered by tag id, then property).

**6.223.2.118 getTagProperties() [2/3]**

```
QList< TagProperty > Digikam::CoreDB::getTagProperties (
    const QString & property ) const
```

Returns the list of tag properties with the given attribute.

**6.223.2.119 getTagProperties() [3/3]**

```
QList< TagProperty > Digikam::CoreDB::getTagProperties (
    int tagID ) const
```

Returns the list of tag properties of the given tag.

**6.223.2.120 getTagShortInfos()**

```
QList< TagShortInfo > Digikam::CoreDB::getTagShortInfos ( ) const
```

Returns all tags in the database with their parent id and name, ordered by id.

**6.223.2.121 getTagsWithProperty()**

```
QList< int > Digikam::CoreDB::getTagsWithProperty (
    const QString & property ) const
```

Returns a list of tag ids with the specified property.

**6.223.2.122 getUniqueHashVersion()**

```
int Digikam::CoreDB::getUniqueHashVersion ( ) const
```

Returns the version used for the unique hash in this database. The value is cached.

**6.223.2.123 getUserFilterSettings()**

```
void Digikam::CoreDB::getUserFilterSettings (
    QString * imageFilterString,
    QString * videoFilterString,
    QString * audioFilterString )
```

Returns the user-configurable filter settings. If you are not interested in a specific value, pass 0.

**6.223.2.124 getVideoMetadata()**

```
QVariantList Digikam::CoreDB::getVideoMetadata (
    qlonglong imageID,
    DatabaseFields::VideoMetadata metadataFields = DatabaseFields::VideoMetadataAll )
const
```

Read video metadata. Parameters as above.

**6.223.2.125 hasImageHistory()**

```
bool Digikam::CoreDB::hasImageHistory (
    qlonglong imageId ) const
```

Returns true if the image has a history stored in DB. If not, it returns false.

**6.223.2.126 hasTags()**

```
bool Digikam::CoreDB::hasTags (
    const QList< qlonglong > & imageIDList ) const
```

Given a set of items (identified by their IDs), this will see if any of the items has a tag.

**Parameters**

<i>imageIDList</i>	a list of IDs of the items
--------------------	----------------------------

**Returns**

true if at least one of the items has a tag

**6.223.2.127 integrityCheck()**

```
bool Digikam::CoreDB::integrityCheck ( ) const
```

Returns true if the integrity of the database is preserved.

**6.223.2.128 makeStaleAlbum()**

```
void Digikam::CoreDB::makeStaleAlbum (
    int albumID )
```

Makes the album a stale entry by setting the albumRoot to 0. Emits the same changeset as [deleteAlbum\(\)](#)

**6.223.2.129 migrateAlbumRoot()**

```
void Digikam::CoreDB::migrateAlbumRoot (
    int rootId,
    const QString & identifier )
```

Migrates a given album root to a new disk location. This only changes the values in the AlbumRoots table. It is expected that this merely reflects underlying partition changes, still pointing to the same data.

**6.223.2.130 moveItem()**

```
void Digikam::CoreDB::moveItem (
    int srcAlbumID,
    const QString & srcName,
    int dstAlbumID,
    const QString & dstName )
```

Move the attributes of an item to a different item. Useful when say a file is renamed

**Parameters**

<i>srcAlbumID</i>	the id of the source album
<i>dstAlbumID</i>	the id of the destination album
<i>srcName</i>	the name of the source file
<i>dstName</i>	the name of the destination file

**6.223.2.131 removeAllImageComments()**

```
void Digikam::CoreDB::removeAllImageComments (
    qulonglong imageID )
```

Remove all [ItemComments](#)

**6.223.2.132 removeAllItemCopyrightProperties()**

```
void Digikam::CoreDB::removeAllItemCopyrightProperties (
    qulonglong imageID )
```

Removes all copyright properties for the given image id.

**6.223.2.133 removeImageComment()**

```
void Digikam::CoreDB::removeImageComment (
    int commentId,
    qlonglong imageID )
```

Remove the specified entry in [ItemComments](#)

**6.223.2.134 removeImageRelation()**

```
void Digikam::CoreDB::removeImageRelation (
    qlonglong subjectId,
    qlonglong objectId,
    DatabaseRelation::Type type )
```

Removes image relations. The batch methods return all removed partners.

**6.223.2.135 removeImageTagProperties()**

```
void Digikam::CoreDB::removeImageTagProperties (
    qlonglong imageId,
    int tagId = -1,
    const QString & property = QString(),
    const QString & value = QString() )
```

Removes properties for the given tag. If the value is given, removes only the entries with the given property/value pair. If only property is given, removes all properties with the given name. If property is null, removes all properties for the given tag. If tagId is -1, removes all image tag properties for the given image.

**Note**

: After the first parameter you give as a wildcard, the following will be ignored and taken as wildcard as well.

**6.223.2.136 removeItemAllTags()**

```
void Digikam::CoreDB::removeItemAllTags (
    qlonglong imageID,
    const QList< int > & currentTagIds )
```

Remove all tags for the item

**Parameters**

<i>imageID</i>	the ID of the item
<i>currentTagIds</i>	the current tags ids assigned to the item

**6.223.2.137 removeItemCopyrightProperties()**

```
void Digikam::CoreDB::removeItemCopyrightProperties (
    qlonglong imageID,
    const QString & property = QString(),
    const QString & extraValue = QString(),
    const QString & value = QString() )
```

Removes copyright properties for the given image id. All values after the first null value, in order of parameters, are treated as wild cards (you can give value as wildcard; value and extraValue; or property, extraValue and value).

**Warning**

: extraValue is ordered before value in this method! Take a care to the parameter order.

**6.223.2.138 removeItemPosition()**

```
void Digikam::CoreDB::removeItemPosition (
    qlonglong imageid )
```

Remove the entry in ItemPositions for the given image

**6.223.2.139 removeItemPositionAltitude()**

```
void Digikam::CoreDB::removeItemPositionAltitude (
    qlonglong imageid )
```

Remove the altitude in ItemPositions for the given image

**6.223.2.140 removeItems()**

```
void Digikam::CoreDB::removeItems (
    const QList< qlonglong > & itemIDs,
    const QList< int > & albumIDs = QList<int>() )
```

Marks all items in the list as removed, resets their dirids. The items can later be removed by [deleteRemovedItems\(\)](#).

**Parameters**

<i>itemIDs</i>	a list of item IDs to be marked
<i>albumIDs</i>	this parameter is purely informational. it shall contain the albums that the items are removed from.

**6.223.2.141 removeItemsFromAlbum()**

```
void Digikam::CoreDB::removeItemsFromAlbum (
    int albumID,
    const QList< qlonglong > & ids_forInformation = QList<qlonglong>() )
```

Marks all items in the specified album as removed, resets their dirids. The album can be deleted afterwards without removing the entries for the items, which can later be removed by [deleteRemovedItems\(\)](#).

#### Parameters

<i>albumID</i>	The id of the album
<i>ids_forInformation</i>	Fully optional: The image ids in the album, if you know them anyway. This parameter is only used for distributing the change notification.

#### 6.223.2.142 removeItemsPermanently()

```
void Digikam::CoreDB::removeItemsPermanently (
    const QList< qlonglong > & itemIDs,
    const QList< int > & albumIDs = QList<int>() )
```

Marks all items in the list as obsolete, resets their dirids. The items can later be removed by [deleteRemovedItems\(\)](#).

#### Parameters

<i>itemIDs</i>	a list of item IDs to be marked
<i>albumIDs</i>	this parameter is purely informational. it shall contain the albums that the items are removed from.

#### 6.223.2.143 removeItemTag()

```
void Digikam::CoreDB::removeItemTag (
    qlonglong imageID,
    int tagID )
```

Remove a specific tag for the item

#### Parameters

<i>imageID</i>	the ID of the item
<i>tagID</i>	the tagID for the tag

#### 6.223.2.144 removeTagProperties()

```
void Digikam::CoreDB::removeTagProperties (
    int tagId,
    const QString & property = QString(),
    const QString & value = QString() )
```

Removes properties for the given tag. If the value is given, removes only the entries with the given property/value pair. If only property is given, removes all properties with the given name. If property is null, removes all properties for the given tag.



**6.223.2.145 removeTagsFromItems()**

```
void Digikam::CoreDB::removeTagsFromItems (
    const QList< qlonglong > & imageIDs,
    const QList< int > & tagIDs )
```

Remove each tag from a list of tags from a each member of a list of items.

**6.223.2.146 renameAlbum()**

```
void Digikam::CoreDB::renameAlbum (
    int albumID,
    int newAlbumRootId,
    const QString & newRelativePath )
```

Give an existing album a new relativePath and a newAlbumRootId

**6.223.2.147 renameItem()**

```
void Digikam::CoreDB::renameItem (
    qlonglong imageID,
    const QString & newName )
```

Rename the item. Note: we not use here [ImageChangeset](#).

**6.223.2.148 scanAlbums()**

```
AlbumInfo::List Digikam::CoreDB::scanAlbums ( ) const
```

Returns all albums and their attributes in the database

**Returns**

a list of albums and their attributes

**6.223.2.149 scanSearches()**

```
SearchInfo::List Digikam::CoreDB::scanSearches ( ) const
```

Returns all searches from the database

**Returns**

a list of searches from the database

**6.223.2.150 scanTags()**

```
TagInfo::List Digikam::CoreDB::scanTags ( ) const
```

Returns all tags and their attributes in the database

**Returns**

a list of tags and their attributes

**6.223.2.151 setAlbumCaption()**

```
void Digikam::CoreDB::setAlbumCaption (
    int albumID,
    const QString & caption )
```

Set a caption for the album.

## Parameters

<i>albumID</i>	the id of the album
<i>caption</i>	the new caption for the album

**6.223.2.152 setAlbumCategory()**

```
void Digikam::CoreDB::setAlbumCategory (
    int albumID,
    const QString & category )
```

Set a category for the album.

## Parameters

<i>albumID</i>	the id of the album
<i>category</i>	the new category for the album

**6.223.2.153 setAlbumDate()**

```
void Digikam::CoreDB::setAlbumDate (
    int albumID,
    const QDate & date )
```

Set a date for the album.

## Parameters

<i>albumID</i>	the id of the album
<i>date</i>	the date for the album

**6.223.2.154 setAlbumIcon()**

```
void Digikam::CoreDB::setAlbumIcon (
    int albumID,
    qlonglong iconID )
```

Set the icon for the album.

## Parameters

<i>albumID</i>	the id of the album
<i>iconID</i>	the id of the icon file

**6.223.2.155 setAlbumModificationDate()**

```
void Digikam::CoreDB::setAlbumModificationDate (
    int albumID,
    const QDateTime & modificationDate )
```

Set the modification date time for the album.

**Parameters**

<i>albumID</i>	the id of the album
<i>modificationDate</i>	the modification date time for the album

**6.223.2.156 setAlbumRootCaseSensitivity()**

```
void Digikam::CoreDB::setAlbumRootCaseSensitivity (
    int rootId,
    CollectionLocation::CaseSensitivity caseSensitivity )
```

Sets the case sensitivity of the specified album root to a new value.

**6.223.2.157 setAlbumRootLabel()**

```
void Digikam::CoreDB::setAlbumRootLabel (
    int rootId,
    const QString & newLabel )
```

Changes the label of the specified album root

**Parameters**

<i>rootId</i>	the id of the album root
<i>newLabel</i>	new label for the album root

**6.223.2.158 setAlbumRootPath()**

```
void Digikam::CoreDB::setAlbumRootPath (
    int rootId,
    const QString & newPath )
```

Changes the specificPath of the specified album root

**Parameters**

<i>rootId</i>	the id of the album root
<i>newPath</i>	new path for the album root

**6.223.2.159 setAlbumRootType()**

```
void Digikam::CoreDB::setAlbumRootType (
    int rootId,
    CollectionLocation::Type newType )
```

Sets the type of the specified album root to a new value.

**6.223.2.160 setFilterSettings()**

```
void Digikam::CoreDB::setFilterSettings (
    const QStringList & imageFilter,
    const QStringList & videoFilter,
    const QStringList & audioFilter )
```

Sets the main filter settings of the database. Should only be called at schema update.

**6.223.2.161 setImageComment()**

```
int Digikam::CoreDB::setImageComment (
    qlonglong imageID,
    const QString & comment,
    DatabaseComment::Type type,
    const QString & language = QString(),
    const QString & author = QString(),
    const QDateTime & date = QDateTime() ) const
```

Sets the comments for the image. A comment for the image with the same source, language and author will be overwritten.

**Parameters**

<i>imageID</i>	The imageID of the image
<i>comment</i>	The comment string
<i>type</i>	The type of the comment
<i>language</i>	Information about the language of the comment. A null string shall be used if language information is not available from the source, or if the comment is in the default language.
<i>author</i>	Optional information about the author who wrote the comment. If not supported by the source, pass a null string.
<i>date</i>	Optional information about the date when the comment was written. If not supported by the source, pass a null string.

**Returns**

the comment ID of the comment

**6.223.2.162 setImageProperty()**

```
void Digikam::CoreDB::setImageProperty (
    qlonglong imageID,
```

```
const QString & property,  
const QString & value )
```

Sets the property with the given name for the given image to the specified value

#### 6.223.2.163 **setItemAlbum()**

```
void Digikam::CoreDB::setItemAlbum (   
    qlonglong imageID,  
    qlonglong albumId )
```

Updates the album field for the item.

#### Note

: Do not use this to move the item. This function only has the purpose to reuse image infos for restored images from trash.

#### 6.223.2.164 **setItemCopyrightProperty()**

```
void Digikam::CoreDB::setItemCopyrightProperty (   
    qlonglong imageID,  
    const QString & property,  
    const QString & value,  
    const QString & extraValue = QString(),  
    CopyrightPropertyUnique uniqueness = PropertyUnique )
```

Sets the property with the given name for the given image to the specified value and extraValue

#### 6.223.2.165 **setItemHistory()**

```
void Digikam::CoreDB::setItemHistory (   
    qlonglong imageId,  
    const QString & history )
```

Changes (adds or updates) the image history

#### 6.223.2.166 **setItemManualOrder()**

```
void Digikam::CoreDB::setItemManualOrder (   
    qlonglong imageID,  
    qlonglong value )
```

Updates the manualOrder field for the item.

#### 6.223.2.167 **setItemModificationDate()**

```
void Digikam::CoreDB::setItemModificationDate (   
    qlonglong imageID,  
    const QDateTime & modificationDate )
```

Updates the modification date field for the item.

**6.223.2.168 setItemStatus()**

```
void Digikam::CoreDB::setItemStatus (
    qlonglong imageID,
    DatabaseItem::Status status )
```

Updates the status field for the item.

**Note**

: Do not use this to set to the Removed status, see [removeItems\(\)](#).

**6.223.2.169 setSetting()**

```
void Digikam::CoreDB::setSetting (
    const QString & keyword,
    const QString & value )
```

This adds a keyword-value combination to the database Settings table if the keyword already exists, the value will be replaced with the new value.

**Parameters**

<i>keyword</i>	The keyword
<i>value</i>	The value

**6.223.2.170 setTagIcon()**

```
void Digikam::CoreDB::setTagIcon (
    int tagID,
    const QString & iconKDE,
    qlonglong iconID )
```

Set the icon for the tag.

**Parameters**

<i>tagID</i>	the id of the tag
<i>iconKDE</i>	the filename for the kde icon file
<i>iconID</i>	the id of the icon file Note: Only one of the iconKDE or iconID parameters is used. if the iconKDE parameter is empty, then the iconID parameter is used

**6.223.2.171 setTagName()**

```
void Digikam::CoreDB::setTagName (
    int tagID,
    const QString & name )
```

Set a new name for the tag.

## Parameters

<i>tagID</i>	the id of the tag
<i>name</i>	the new name for the tag

**6.223.2.172 setTagParentID()**

```
void Digikam::CoreDB::setTagParentID (
    int tagID,
    int newParentTagID )
```

Set the parent tagid for the tag. This is equivalent to reparenting the tag

## Parameters

<i>tagID</i>	the id of the tag
<i>newParentTagID</i>	the new parentid for the tag

**6.223.2.173 setUserFilterSettings()**

```
void Digikam::CoreDB::setUserFilterSettings (
    const QStringList & imageFilter,
    const QStringList & videoFilter,
    const QStringList & audioFilter )
```

Sets the user-configurable filter settings. The lists shall be as specified for `getFilterSettings`. They may include entries starting with "-", which indicates that this format shall be removed from the list, if it is included in the main settings list.

**6.223.2.174 updateItem()**

```
void Digikam::CoreDB::updateItem (
    qulonglong imageID,
    DatabaseItem::Category category,
    const QDateTime & modificationDate,
    qulonglong fileSize,
    const QString & uniqueHash )
```

Update the fields of the Images table that have changed when the file has been modified on disk.

## Parameters

<i>imageID</i>	the image that has been modified
<i>category</i>	the image category that has been modified
<i>modificationDate</i>	the image time-stamp that has been modified
<i>fileSize</i>	the image file size that has been modified
<i>uniqueHash</i>	the image hash that has been modified



**6.223.2.175 updateSearch()**

```
void Digikam::CoreDB::updateSearch (
    int searchID,
    DatabaseSearch::Type type,
    const QString & name,
    const QString & query )
```

Updates Search with new attributes

**Parameters**

<i>searchID</i>	the id of the search
<i>type</i>	type of the search
<i>name</i>	name of the search
<i>query</i>	database query of the search

**6.223.2.176 vacuum()**

```
void Digikam::CoreDB::vacuum ( )
```

Shrinks the database.

**6.224 Digikam::CoreDbAccess Class Reference****Public Types**

- enum **ApplicationStatus** { **MainApplication** , **DatabaseSlave** }

**Public Member Functions**

- [CoreDbBackend](#) \* [backend](#) () const
- [CoreDbAccess](#) ()
- [CoreDB](#) \* [db](#) () const
- [QString](#) [lastError](#) ()
- void [setLastError](#) (const [QString](#) &error)

**Static Public Member Functions**

- static bool [checkReadyForUse](#) ([InitializationObserver](#) \*const observer=nullptr)
- static void [cleanUpDatabase](#) ()
- static [CoreDbWatch](#) \* [databaseWatch](#) ()
- static void [initDbEngineErrorHandler](#) ([DbEngineErrorHandler](#) \*const errorhandler)
- static [DbEngineParameters](#) [parameters](#) ()
- static void [setParameters](#) (const [DbEngineParameters](#) &parameters)
- static void [setParameters](#) (const [DbEngineParameters](#) &parameters, [ApplicationStatus](#) status)

## Friends

- class **CoreDbAccessUnlock**

### 6.224.1 Detailed Description

The [CoreDbAccess](#) provides access to the database: Create an instance of this class on the stack to retrieve a pointer to the database. While you hold an instance of [CoreDbAccess](#), the database access is locked for other threads, but *not* for other processes. This is due to the fact that while databases allow concurrent access (of course), their client libs may not be thread-safe.

When initializing your application, you need to call two methods:

- in a not-yet-multithreaded context, you need to call `setParameters`
- to make sure that the database is available and the schema is properly initialized, call [checkReadyForUse\(\)](#)

### 6.224.2 Constructor & Destructor Documentation

#### 6.224.2.1 CoreDbAccess()

```
Digikam::CoreDbAccess::CoreDbAccess ( )
```

Create a [CoreDbAccess](#) object for the default database. Note that when initializing your app, `setParameters` need to be called (in a not-yet-multithreaded context) for this to work. If the database is not yet opened, it will be opened. The schema will not be checked, use [checkReadyForUse\(\)](#) for a full opening process including schema update and error messages.

### 6.224.3 Member Function Documentation

#### 6.224.3.1 backend()

```
CoreDbBackend * Digikam::CoreDbAccess::backend ( ) const
```

Retrieve a pointer to the database backend

#### 6.224.3.2 checkReadyForUse()

```
bool Digikam::CoreDbAccess::checkReadyForUse (
    InitializationObserver *const observer = nullptr ) [static]
```

Method to one-time initialize a database when new parameters have been set: Make sure that the database is open, that the schema has properly been initialized. If the parameters were not changed, this method has no effect.

## Returns

if the database is ready for use

### 6.224.3.3 cleanUpDatabase()

```
void Digikam::CoreDbAccess::cleanUpDatabase ( ) [static]
```

Clean up the database access. When this function has been called, the access can be restored by calling `setParameters`. Construction a database access object otherwise after calling this method will crash.

### 6.224.3.4 databaseWatch()

```
CoreDbWatch * Digikam::CoreDbAccess::databaseWatch ( ) [static]
```

Return the [CoreDbWatch](#).

### 6.224.3.5 db()

```
CoreDB * Digikam::CoreDbAccess::db ( ) const
```

Retrieve a pointer to the album database

### 6.224.3.6 initDbEngineErrorHandler()

```
void Digikam::CoreDbAccess::initDbEngineErrorHandler (
    DbEngineErrorHandler *const errorHandler ) [static]
```

[Setup](#) the errors handler instance.

### 6.224.3.7 lastError()

```
QString Digikam::CoreDbAccess::lastError ( )
```

Returns the error message for the last error that occurred, or a null QString of no error occurred.

### 6.224.3.8 parameters()

```
DbEngineParameters Digikam::CoreDbAccess::parameters ( ) [static]
```

Return the default parameters

### 6.224.3.9 setLastError()

```
void Digikam::CoreDbAccess::setLastError (
    const QString & error )
```

Set the "last error" message. This method is not for public use.

### 6.224.3.10 setParameters()

```
void Digikam::CoreDbAccess::setParameters (
    const DbEngineParameters & parameters ) [static]
```

Set the default parameters. Call this function at least once in the starting phase of your application, when no other threads will yet access the database, to initialize DatabaseAccess. After this initial call, it is thread-safe to call this function again. In a subsequent call, if the parameters are identical, nothing is done. If the parameters change, the current database will be closed. When parameters have been set or changed, the new one will be opened on-demand, i.e. when the first [CoreDbAccess](#) object is constructed.

## 6.225 Digikam::CoreDbAccessUnlock Class Reference

### Public Member Functions

- [CoreDbAccessUnlock](#) ()
- [CoreDbAccessUnlock](#) ([CoreDbAccess](#) \*const access)

### 6.225.1 Constructor & Destructor Documentation

#### 6.225.1.1 CoreDbAccessUnlock()

```
Digikam::CoreDbAccessUnlock::CoreDbAccessUnlock ( )
```

Acquire an object of this class if you want to assure that the [CoreDbAccess](#) is *not* held during the lifetime of the object. At creation, the lock is obtained shortly, then all locks are released. At destruction, all locks are acquired again. If you need to access any locked structures during lifetime, acquire a new [CoreDbAccess](#).



- bool `initSchema` (`CoreDbSchemaUpdater *updater`)
- void `recordChangeset` (const `AlbumChangeset &changeset`)
- void `recordChangeset` (const `AlbumRootChangeset &changeset`)
- void `recordChangeset` (const `CollectionImageChangeset &changeset`)
- void `recordChangeset` (const `ImageChangeset &changeset`)
- void `recordChangeset` (const `ImageTagChangeset &changeset`)
- void `recordChangeset` (const `SearchChangeset &changeset`)
- void `recordChangeset` (const `TagChangeset &changeset`)
- void `setCoreDbWatch` (`CoreDbWatch *watch`)

## Public Member Functions inherited from `Digikam::BdEngineBackend`

- QDateTime `asDBDateTime` (const QDateTime &dateTime) const
- `BdEngineBackend` (const QString &backendName, `DbEngineLocking *const locking`)
- `BdEngineBackend` (const QString &backendName, `DbEngineLocking *const locking`, `BdEngineBackendPrivate &dd`)
- `BdEngineBackend::QueryState` `beginTransaction` ()
- bool `checkOrSetWALMode` ()
- void `close` ()
- `BdEngineBackend::QueryState` `commitTransaction` ()
- `DbEngineConfigSettings` `configElement` () const
- bool `connectionErrorHandling` (int retries)
- `DbEngineSqlQuery` `copyQuery` (const `DbEngineSqlQuery &old`)
- DbType `databaseType` () const
- bool `exec` (`DbEngineSqlQuery &query`)
- bool `execBatch` (`DbEngineSqlQuery &query`)
- `QueryState` `execDBAction` (const `DbEngineAction &action`, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- `QueryState` `execDBAction` (const `DbEngineAction &action`, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- `QueryState` `execDBAction` (const QString &action, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- `QueryState` `execDBAction` (const QString &action, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- QSqlQuery `execDBActionQuery` (const `DbEngineAction &action`, const QMap< QString, QVariant > &bindingMap)
- QSqlQuery `execDBActionQuery` (const QString &action, const QMap< QString, QVariant > &bindingMap)
- `QueryState` `execDirectSql` (const QString &query)
- `QueryState` `execDirectSqlWithResult` (const QString &query, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- `DbEngineSqlQuery` `execQuery` (const QString &sql)
- `DbEngineSqlQuery` `execQuery` (const QString &sql, const QList< QVariant > &boundValues)
- `DbEngineSqlQuery` `execQuery` (const QString &sql, const QMap< QString, QVariant > &bindingMap)
- `DbEngineSqlQuery` `execQuery` (const QString &sql, const QVariant &boundValue1)
- `DbEngineSqlQuery` `execQuery` (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2)
- `DbEngineSqlQuery` `execQuery` (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- `DbEngineSqlQuery` `execQuery` (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)
- void `execQuery` (`DbEngineSqlQuery &preparedQuery`, const QList< QVariant > &boundValues)
- void `execQuery` (`DbEngineSqlQuery &preparedQuery`, const QVariant &boundValue1)
- void `execQuery` (`DbEngineSqlQuery &preparedQuery`, const QVariant &boundValue1, const QVariant &boundValue2)

- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)
- [QueryState](#) **execSql** (const QString &sql, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execUpsertDBAction** (const [DbEngineAction](#) &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [QueryState](#) **execUpsertDBAction** (const QString &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [DbEngineAction](#) **getDBAction** (const QString &actionName) const
- [DbEngineSqlQuery](#) **getQuery** ()
- [QueryState](#) **handleQueryResult** ([DbEngineSqlQuery](#) &query, QList< QVariant > \*const values, QVariant \*const lastInsertId)
- bool **isCompatible** (const [DbEngineParameters](#) &parameters)
- bool **isInTransaction** () const
- bool **isOpen** () const
- bool **isReady** () const
- QString **lastError** ()
- QSqlError **lastSQLError** ()
- int **maximumBoundValues** () const
- bool **open** (const [DbEngineParameters](#) &parameters)
- [DbEngineSqlQuery](#) **prepareQuery** (const QString &sql)
- bool **queryErrorHandling** ([DbEngineSqlQuery](#) &query, int retries)
- QList< QVariant > **readToList** ([DbEngineSqlQuery](#) &query)
- void **rollbackTransaction** ()
- void **setDbEngineErrorHandler** ([DbEngineErrorHandler](#) \*const handler)
- void **setForeignKeyChecks** (bool check)

- [Status status](#) () const
- QStringList [tables](#) ()
- bool **transactionErrorHandling** (const QSqlError &[lastError](#), int retries)

### Additional Inherited Members

### Public Types inherited from [Digikam::BdEngineBackend](#)

- enum **DbType** { [SQLite](#) , [MySQL](#) }
- enum **QueryOperationStatus** { [ExecuteNormal](#) , [Wait](#) , [AbortQueries](#) }
- enum [QueryStateEnum](#) { [NoErrors](#) , [SQLError](#) , [ConnectionError](#) }
- enum [Status](#) { [Unavailable](#) , [Open](#) , [OpenSchemaChecked](#) }

### Protected Attributes inherited from [Digikam::BdEngineBackend](#)

- [BdEngineBackendPrivate](#) \*const **d\_ptr** = nullptr

## 6.226.1 Member Function Documentation

### 6.226.1.1 [initSchema\(\)](#)

```
bool Digikam::CoreDbBackend::initSchema (
    CoreDbSchemaUpdater * updater )
```

Initialize the database schema to the current version, carry out upgrades if necessary. Shall only be called from the thread that called [open\(\)](#).

### 6.226.1.2 [recordChangeset\(\)](#)

```
void Digikam::CoreDbBackend::recordChangeset (
    const ImageChangeset & changeset )
```

Notify all listeners of the changeset

### 6.226.1.3 [setCoreDbWatch\(\)](#)

```
void Digikam::CoreDbBackend::setCoreDbWatch (
    CoreDbWatch * watch )
```

Sets the global database watch



## 6.227 Digikam::CoreDbBackendPrivate Class Reference

Inheritance diagram for Digikam::CoreDbBackendPrivate:



### Classes

- class [ChangesetContainer](#)

## Public Member Functions

- **CoreDbBackendPrivate** ([CoreDbBackend](#) \*const backend)
- void **sendToWatch** (const [AlbumChangeset](#) &changeset)
- void **sendToWatch** (const [AlbumRootChangeset](#) &changeset)
- void **sendToWatch** (const [CollectionImageChangeset](#) &changeset)
- void **sendToWatch** (const [ImageChangeset](#) &changeset)
- void **sendToWatch** (const [ImageTagChangeset](#) &changeset)
- void **sendToWatch** (const [SearchChangeset](#) &changeset)
- void **sendToWatch** (const [TagChangeset](#) &changeset)
- void **transactionFinished** () override

## Public Member Functions inherited from [Digikam::BdEngineBackendPrivate](#)

- **BdEngineBackendPrivate** ([BdEngineBackend](#) \*const backend)
- bool **checkOperationStatus** ()
- bool **checkRetrySQLiteLockError** (int retries)
- void **closeDatabaseForThread** ()
- void **connectionErrorAbortQueries** () override
- void **connectionErrorContinueQueries** () override
- QString **connectionName** ()
- QSqlDatabase **createDatabaseConnection** ()
- QSqlError **databaseErrorForThread** ()
- QSqlDatabase **databaseForThread** ()
- void **debugOutputFailedQuery** (const QSqlQuery &query) const
- void **debugOutputFailedTransaction** (const QSqlError &error) const
- bool **decrementTransactionCount** ()
- bool **handleWithErrorHandler** (const [DbEngineSqlQuery](#) \*const query)
- bool **incrementTransactionCount** ()
- void **init** (const QString &connectionName, [DbEngineLocking](#) \*const locking)
- bool **isConnectionError** (const [DbEngineSqlQuery](#) &query) const
- bool **isInMainThread** () const
- bool **isInUIThread** () const
- bool **isSQLiteLockError** (const [DbEngineSqlQuery](#) &query) const
- bool **isSQLiteLockTransactionError** (const QSqlError &lastError) const
- bool **needToConsultUserForError** (const [DbEngineSqlQuery](#) &query) const
- bool **needToHandleWithErrorHandler** (const [DbEngineSqlQuery](#) &query) const
- void **queryOperationWakeAll** ([BdEngineBackend::QueryOperationStatus](#) status)
- bool **reconnectOnError** () const
- bool **resetDatabaseForThread** ()
- void **setDatabaseErrorForThread** (const QSqlError &lastError)
- void **setQueryOperationFlag** ([BdEngineBackend::QueryOperationStatus](#) status)

## Public Attributes

- [ChangesetContainer](#)< [AlbumChangeset](#) > **albumChangesetContainer**
- [ChangesetContainer](#)< [AlbumRootChangeset](#) > **albumRootChangesetContainer**
- [ChangesetContainer](#)< [CollectionImageChangeset](#) > **collectionImageChangesetContainer**
- [ChangesetContainer](#)< [ImageChangeset](#) > **imageChangesetContainer**
- [ChangesetContainer](#)< [ImageTagChangeset](#) > **imageTagChangesetContainer**
- [ChangesetContainer](#)< [SearchChangeset](#) > **searchChangesetContainer**
- [ChangesetContainer](#)< [TagChangeset](#) > **tagChangesetContainer**
- [CoreDbWatch](#) \* **watch** = nullptr

## Public Attributes inherited from [Digikam::BdEngineBackendPrivate](#)

- QString **backendName**
- QWaitCondition **busyWaitCondVar**
- QMutex **busyWaitMutex**
- int **currentValidity** = 0
- [DbEngineErrorHandler](#) \* **errorHandler** = nullptr
- QWaitCondition **errorLockCondVar**
- QMutex **errorLockMutex**
- [BdEngineBackend::QueryOperationStatus](#) **errorLockOperationStatus** = [BdEngineBackend::ExecuteNormal](#)
- bool **isInTransaction** = false
- [DbEngineLocking](#) \* **lock** = nullptr
- [BdEngineBackend::QueryOperationStatus](#) **operationStatus** = [BdEngineBackend::ExecuteNormal](#)
- [DbEngineParameters](#) **parameters**
- [BdEngineBackend](#) \*const **q** = nullptr
- [BdEngineBackend::Status](#) **status** = [BdEngineBackend::Unavailable](#)
- QThreadStorage< [DbEngineThreadData](#) \* > **threadDataStorage**

### 6.227.1 Member Function Documentation

#### 6.227.1.1 transactionFinished()

```
void Digikam::CoreDbBackendPrivate::transactionFinished ( ) [inline], [override], [virtual]
```

Reimplemented from [Digikam::BdEngineBackendPrivate](#).

## 6.228 Digikam::CoreDbBackendPrivate::ChangesetContainer< T > Class Template Reference

### Public Member Functions

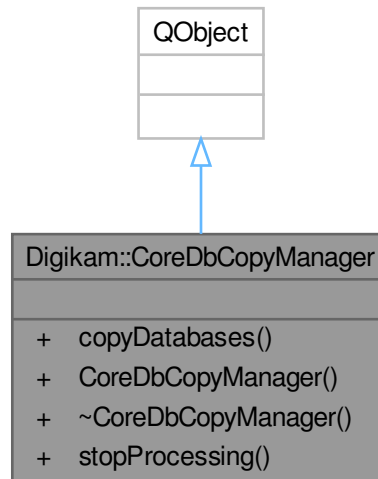
- **ChangesetContainer** ([CoreDbBackendPrivate](#) \*const dd)
- void **recordChangeset** (const T &changeset)
- void **sendOut** ()

### Public Attributes

- QList< T > **changesets**
- [CoreDbBackendPrivate](#) \*const **d** = nullptr

## 6.229 Digikam::CoreDbCopyManager Class Reference

Inheritance diagram for Digikam::CoreDbCopyManager:



### Public Types

- enum **FinishStates** { **success** , **failed** , **canceled** }

### Public Slots

- void **stopProcessing** ()

### Signals

- void **finished** (int finishState, const QString &errorMsg)
- void **smallStepStarted** (int currValue, int maxValue)
- void **stepStarted** (const QString &stepName)

### Public Member Functions

- void **copyDatabases** (const [DbEngineParameters](#) &fromDBParameters, const [DbEngineParameters](#) &toDBParameters)

## 6.230 Digikam::CoreDbDownloadHistory Class Reference

### Static Public Member Functions

- static void **setDownloaded** (const QString &identifier, const QString &name, qulonglong fileSize, const QDateTime &date)
- static [CamItemInfo::DownloadStatus](#) **status** (const QString &identifier, const QString &name, qulonglong fileSize, const QDateTime &date)

## 6.230.1 Member Function Documentation

### 6.230.1.1 setDownloaded()

```
void Digikam::CoreDbDownloadHistory::setDownloaded (
    const QString & identifier,
    const QString & name,
    qulonglong fileSize,
    const QDateTime & date ) [static]
```

Sets the status of the item to Downloaded

### 6.230.1.2 status()

```
CamItemInfo::DownloadStatus Digikam::CoreDbDownloadHistory::status (
    const QString & identifier,
    const QString & name,
    qulonglong fileSize,
    const QDateTime & date ) [static]
```

Queries the status of a download item that is uniquely described by the four parameters. The identifier is recommended to be an MD5 hash of properties describing the camera, if available, and the directory path (though you are free to use all four parameters as you want)

## 6.231 Digikam::CoreDbNameFilter Class Reference

### Public Member Functions

- [CoreDbNameFilter](#) (const QString &filter)
- bool [matches](#) (const QString &name)

### Protected Attributes

- QList< QRegularExpression > [m\\_filterList](#)

## 6.231.1 Constructor & Destructor Documentation

### 6.231.1.1 CoreDbNameFilter()

```
Digikam::CoreDbNameFilter::CoreDbNameFilter (
    const QString & filter ) [explicit]
```

Creates a name filter object with the given filter string. The string is a list of text parts of which one needs to match (file suffixes), separated by ';' characters.

## 6.231.2 Member Function Documentation

### 6.231.2.1 matches()

```
bool Digikam::CoreDbNameFilter::matches (
    const QString & name )
```

Returns if the specified name matches this filter

## 6.232 Digikam::CoreDbOperationGroup Class Reference

### Public Member Functions

- void [allowLift](#) ()
- [CoreDbOperationGroup](#) ()
- [CoreDbOperationGroup](#) ([CoreDbAccess](#) \*const access)
- void [lift](#) ()
- void [resetTime](#) ()
- void [setMaximumTime](#) (int msec)

### 6.232.1 Detailed Description

When you intend to execute a number of write operations to the database, group them while holding a [CoreDbOperationGroup](#). For some database systems (SQLite), keeping a transaction across write operations occurring in short time results in enormous speedup (800x). For system that do not need this optimization, this class is a no-op.

### 6.232.2 Constructor & Destructor Documentation

#### 6.232.2.1 CoreDbOperationGroup() [1/2]

```
Digikam::CoreDbOperationGroup::CoreDbOperationGroup ( )
```

Retrieve a [CoreDbAccess](#) object each time when constructing and destructing.

#### 6.232.2.2 CoreDbOperationGroup() [2/2]

```
Digikam::CoreDbOperationGroup::CoreDbOperationGroup (
    CoreDbAccess *const access ) [explicit]
```

Use an existing [CoreDbAccess](#) object, which must live as long as this object exists.

### 6.232.3 Member Function Documentation

#### 6.232.3.1 allowLift()

```
void Digikam::CoreDbOperationGroup::allowLift ( )
```

Allows to [lift](#)(). The transaction will be lifted if the time set by [setMaximumTime](#)() has expired.

### 6.232.3.2 lift()

```
void Digikam::CoreDbOperationGroup::lift ( )
```

This will - if a transaction is held - commit the transaction and acquire a new one. This may improve concurrent access.

### 6.232.3.3 resetTime()

```
void Digikam::CoreDbOperationGroup::resetTime ( )
```

Resets to 0 the time used by [allowLift\(\)](#)

## 6.233 Digikam::CoreDbPrivilegesChecker Class Reference

### Public Member Functions

- bool **checkPriv** ([CoreDbBackend](#) &dbBackend, const QString &dbActionName)
- bool **checkPrivileges** (QStringList &insufficientRights)
- [CoreDbPrivilegesChecker](#) (const [DbEngineParameters](#) &parameters)

## 6.234 Digikam::CoreDbSchemaUpdater Class Reference

### Public Member Functions

- [CoreDbSchemaUpdater](#) ([CoreDB](#) \*const albumDB, [CoreDbBackend](#) \*const backend, const [DbEngineParameters](#) &parameters)
- const QString **getLastErrorMessage** ( )
- void **setCoreDbAccess** ([CoreDbAccess](#) \*const dbAccess)
- void **setObserver** ([InitializationObserver](#) \*const observer)
- bool **update** ( )
- bool **updateUniqueHash** ( )

### Static Public Member Functions

- static int **filterSettingsVersion** ( )
- static bool **isUniqueHashUpToDate** ( )
- static int **schemaVersion** ( )
- static int **uniqueHashVersion** ( )

## 6.235 Digikam::CoreDbTransaction Class Reference

### Public Member Functions

- [CoreDbTransaction](#) ( )
- [CoreDbTransaction](#) ([CoreDbAccess](#) \*const access)

### 6.235.1 Detailed Description

Convenience class: You can create a [CoreDbTransaction](#) object for a scope for which you want to declare a database commit. Equivalent to calling `beginTransaction` and `commitTransaction` on the album db.

### 6.235.2 Constructor & Destructor Documentation

#### 6.235.2.1 `CoreDbTransaction()` [1/2]

```
Digikam::CoreDbTransaction::CoreDbTransaction ( )
```

Retrieve a [CoreDbAccess](#) object each time when constructing and destructing.

#### 6.235.2.2 `CoreDbTransaction()` [2/2]

```
Digikam::CoreDbTransaction::CoreDbTransaction (
    CoreDbAccess *const access ) [explicit]
```

Use an existing [CoreDbAccess](#) object, which must live as long as this object exists.



## 6.236 Digikam::CoreDbUrl Class Reference

Inheritance diagram for Digikam::CoreDbUrl:



### Public Member Functions

- `QString album () const`
- `QUrl albumRoot () const`

*Album URL.*

- int `albumRootId` () const
- QString `albumRootPath` () const
- bool `areaCoordinates` (double \*lat1, double \*lat2, double \*lon1, double \*lon2) const

*MapImages URL.*

- `CoreDbUrl` ()=default
- **CoreDbUrl** (const `CoreDbUrl` &url)
- `CoreDbUrl` (const QUrl &digikamUrl)
- QDate `endDate` () const
- QUrl `fileUrl` () const
- bool `isAlbumUrl` () const
- bool `isDateUrl` () const
- bool `isMapImagesUrl` () const
- bool `isSearchUrl` () const
- bool `isTagUrl` () const
- QString `name` () const
- `CoreDbUrl` & **operator=** (const `CoreDbUrl` &url)
- `CoreDbUrl` & **operator=** (const QUrl &digikamalbumsUrl)
- bool **operator==** (const QUrl &digikamalbumsUrl) const
- `DbEngineParameters` `parameters` () const
- int `searchId` () const

*Search URL.*

- void `setParameters` (const `DbEngineParameters` &parameters)
- QDate `startDate` () const

*Date URL.*

- int `tagId` () const

*Tag URL.*

- QList< int > `tagIds` () const

**Static Public Member Functions**

- static `CoreDbUrl` `albumUrl` (const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` `dateUrl` (const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` **fromAlbumAndName** (const QString &name, const QString &album, const QUrl &albumRoot, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` **fromAlbumAndName** (const QString &name, const QString &album, const QUrl &albumRoot, int albumRootId, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` **fromAreaRange** (const qreal lat1, const qreal lng1, const qreal lat2, const qreal lng2, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` **fromDateForMonth** (const QDate &date, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`)
- static `CoreDbUrl` **fromDateForYear** (const QDate &date, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`)
- static `CoreDbUrl` **fromDateRange** (const QDate &startDate, const QDate &endDate, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` **fromFileUrl** (const QUrl &fileUrl, const QUrl &albumRoot, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` **fromFileUrl** (const QUrl &fileUrl, const QUrl &albumRoot, int albumRootId, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` **fromTagIds** (const QList< int > &tagIds, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`)
- static `CoreDbUrl` `mapImagesUrl` (const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())
- static `CoreDbUrl` `searchUrl` (int searchId, const `DbEngineParameters` &parameters=`CoreDbAccess::parameters`())

## 6.236.1 Constructor & Destructor Documentation

### 6.236.1.1 CoreDbUrl() [1/2]

```
Digikam::CoreDbUrl::CoreDbUrl (
    const QUrl & digikamUrl ) [explicit]
```

Create a [CoreDbUrl](#) object from a QUrl, to retrieve the information stored

### 6.236.1.2 CoreDbUrl() [2/2]

```
Digikam::CoreDbUrl::CoreDbUrl ( ) [default]
```

Create an invalid database URL

## 6.236.2 Member Function Documentation

### 6.236.2.1 album()

```
QString Digikam::CoreDbUrl::album ( ) const
```

Returns the album: This is the directory hierarchy below the album root. In the example above, the album is "/Summer 2007"

### 6.236.2.2 albumRoot()

```
QUrl Digikam::CoreDbUrl::albumRoot ( ) const
```

The following methods are only applicable for a certain protocol each. If the URL has another protocol, the return value of these methods is undefined. Returns the album root URL of the file or album referenced by this URL In the example above, this is "file:///media/fotos"

### 6.236.2.3 albumRootId()

```
int Digikam::CoreDbUrl::albumRootId ( ) const
```

Returns the album root id

### 6.236.2.4 albumRootPath()

```
QString Digikam::CoreDbUrl::albumRootPath ( ) const
```

Returns the album root path of the file or album referenced by this URL In the example above, this is "/media/fotos"

**6.236.2.5 albumUrl()**

```
CoreDbUrl Digikam::CoreDbUrl::albumUrl (
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create an empty digikamalbums:/ url

**6.236.2.6 areaCoordinates()**

```
bool Digikam::CoreDbUrl::areaCoordinates (
    double * lat1,
    double * lat2,
    double * lon1,
    double * lon2 ) const
```

Returns the coordinates surrounding the map area. Returns true if the string to number conversion was ok.

**6.236.2.7 dateUrl()**

```
CoreDbUrl Digikam::CoreDbUrl::dateUrl (
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create an empty digikamdates:/ url

**6.236.2.8 endDate()**

```
QDate Digikam::CoreDbUrl::endDate ( ) const
```

Return the referenced end date (excluded from the referenced span)

**6.236.2.9 fileUrl()**

```
QUrl Digikam::CoreDbUrl::fileUrl ( ) const
```

Converts this digikamalbums:// URL to a [file://](#) URL

**6.236.2.10 fromAlbumAndName()**

```
CoreDbUrl Digikam::CoreDbUrl::fromAlbumAndName (
    const QString & name,
    const QString & album,
    const QUrl & albumRoot,
    int albumRootId,
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create a digikamalbums:/ url from an album name and an image in this album. If name is empty, the album is referenced. Other parameters as above.

### 6.236.2.11 fromDateForMonth()

```
CoreDbUrl Digikam::CoreDbUrl::fromDateForMonth (
    const QDate & date,
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create a digikamdates:/ url for the month of the given date. (The whole month of the given date will included in the referenced time span)

### 6.236.2.12 fromDateForYear()

```
CoreDbUrl Digikam::CoreDbUrl::fromDateForYear (
    const QDate & date,
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create a digikamdates:/ url for the year of the given date. (The whole year of the given date will included in the referenced time span)

### 6.236.2.13 fromDateRange()

```
CoreDbUrl Digikam::CoreDbUrl::fromDateRange (
    const QDate & startDate,
    const QDate & endDate,
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create a digikamdates:/ url for a specified time span which begin with the start date (inclusive) and ends before the end date (exclusive). To cover the whole year of 1984, you would pass 1/1/1984 and 1/1/1985.

### 6.236.2.14 fromFileUrl()

```
CoreDbUrl Digikam::CoreDbUrl::fromFileUrl (
    const QUrl & fileUrl,
    const QUrl & albumRoot,
    int albumRootId,
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

This class shall facilitate the usage of digikamalbums:/, digikamtags:/, digikamdates:/ and digikamsearch: URLs. It provides functions to set and get the parameters stored in such a URL. (with the exception of the search parameters in a search URL, which are out of the scope of this class.) Create a digikamalbums:/ URL from a [file://](#) URL. The file URL can point to a file or a directory (an album in this case). The additional information stored in the URL need to be supplied as well:

- The album root in which the entity pointed to is stored. This is the left part of the file URL. (if the file is "/media/fotos/Summer 2007/001.jpg", the album root may be "/media/fotos")
- The parameters of the database that is referenced

### 6.236.2.15 fromTagIds()

```
CoreDbUrl Digikam::CoreDbUrl::fromTagIds (
    const QList< int > & tagIds,
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create a digikamtags:/ url from a list of tag IDs, where this list is the tag hierarchy of the referenced tag, with the topmost parent first, and the tag last in the list. An empty list references the root tag.

### 6.236.2.16 isAlbumUrl()

```
bool Digikam::CoreDbUrl::isAlbumUrl ( ) const
```

These test for the protocol of this URL. The protocol string is of course available via protocol().

### 6.236.2.17 mapImagesUrl()

```
CoreDbUrl Digikam::CoreDbUrl::mapImagesUrl (
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create an empty digikammapimages:/ url

### 6.236.2.18 name()

```
QString Digikam::CoreDbUrl::name ( ) const
```

Returns the file name. In the example above, this is "001.jpg"

### 6.236.2.19 parameters()

```
DbEngineParameters Digikam::CoreDbUrl::parameters ( ) const
```

Returns the [DbEngineParameters](#) stored in this URL. Applicable to all protocols.

### 6.236.2.20 searchId()

```
int Digikam::CoreDbUrl::searchId ( ) const
```

Return the id of the search.

### 6.236.2.21 searchUrl()

```
CoreDbUrl Digikam::CoreDbUrl::searchUrl (
    int searchId,
    const DbEngineParameters & parameters = CoreDbAccess::parameters() ) [static]
```

Create a digikamsearch: URL for the search with the given id.

**6.236.2.22 setParameters()**

```
void Digikam::CoreDbUrl::setParameters (
    const DbEngineParameters & parameters )
```

Change the database parameters stored in this URL. Applicable to all protocols.

**6.236.2.23 startDate()**

```
QDate Digikam::CoreDbUrl::startDate ( ) const
```

Return the referenced start date (included in the referenced span)

**6.236.2.24 tagId()**

```
int Digikam::CoreDbUrl::tagId ( ) const
```

Returns the tag ID, or -1 if the root tag is referenced

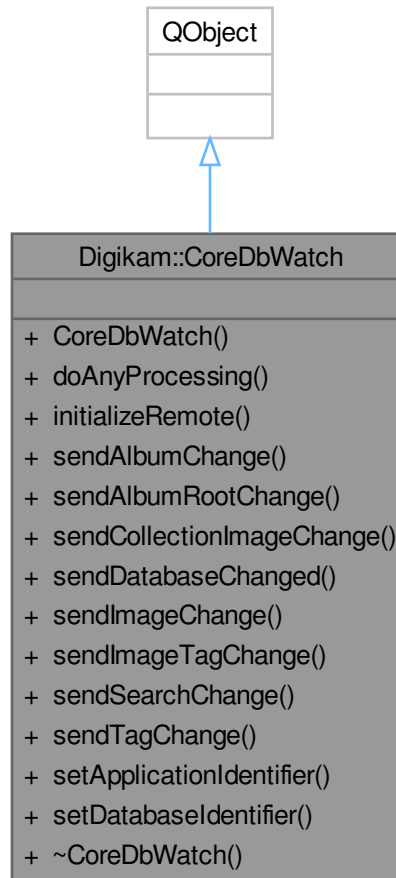
**6.236.2.25 tagIds()**

```
QList< int > Digikam::CoreDbUrl::tagIds ( ) const
```

Returns the tag ids of all tags in the tag path of this tag, the topmost tag in the hierarchy first.

## 6.237 Digikam::CoreDbWatch Class Reference

Inheritance diagram for Digikam::CoreDbWatch:



### Public Types

- enum `DatabaseMode` { `DatabaseMaster` , `DatabaseSlave` }

### Signals

- void `albumChange` (const [AlbumChangeset](#) &changeset)
- void `albumRootChange` (const [AlbumRootChangeset](#) &changeset)
- void `collectionImageChange` (const [CollectionImageChangeset](#) &changeset)
- void `databaseChanged` ()
- void `imageChange` (const [ImageChangeset](#) &changeset)
- void `imageTagChange` (const [ImageTagChangeset](#) &changeset)
- void `searchChange` (const [SearchChangeset](#) &changeset)
- void `tagChange` (const [TagChangeset](#) &changeset)



## Public Member Functions

- void **doAnyProcessing** ()
- void **initializeRemote** (DatabaseMode mode)
- void **sendAlbumChange** (const [AlbumChangeset](#) &changeset)
- void **sendAlbumRootChange** (const [AlbumRootChangeset](#) &changeset)
- void **sendCollectionImageChange** (const [CollectionImageChangeset](#) &changeset)
- void **sendDatabaseChanged** ()
  - library-internal signal-trigger methods*
- void **sendImageChange** (const [ImageChangeset](#) &changeset)
- void **sendImageTagChange** (const [ImageTagChangeset](#) &changeset)
- void **sendSearchChange** (const [SearchChangeset](#) &changeset)
- void **sendTagChange** (const [TagChangeset](#) &changeset)
- void **setApplicationIdentifier** (const QString &identifier)
- void **setDatabaseIdentifier** (const QString &identifier)

## 6.237.1 Member Function Documentation

### 6.237.1.1 databaseChanged

```
void Digikam::CoreDbWatch::databaseChanged ( ) [signal]
```

Retrieve the [CoreDbWatch](#) object from [CoreDbAccess::databaseWatch\(\)](#). This does not describe a change of the contents of a table; rather, it signals that a new database has been loaded. That means all cached content has to be discarded.

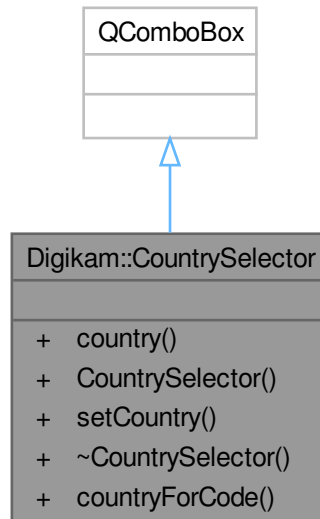
### 6.237.1.2 imageChange

```
void Digikam::CoreDbWatch::imageChange (
    const ImageChangeset & changeset ) [signal]
```

Notifies of changes in the database. Connect to the set of signals that you are interested in.

## 6.238 Digikam::CountrySelector Class Reference

Inheritance diagram for Digikam::CountrySelector:



### Public Member Functions

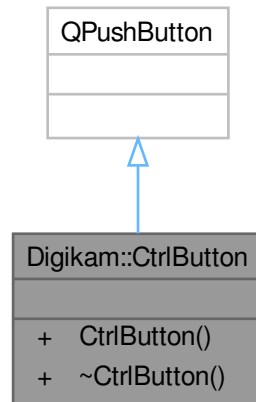
- bool **country** (QString &countryCode, QString &countryName) const
- **CountrySelector** (QWidget \*const parent)
- void **setCountry** (const QString &countryCode)

### Static Public Member Functions

- static QString **countryForCode** (const QString &countryCode)

## 6.239 Digikam::CtrlButton Class Reference

Inheritance diagram for Digikam::CtrlButton:

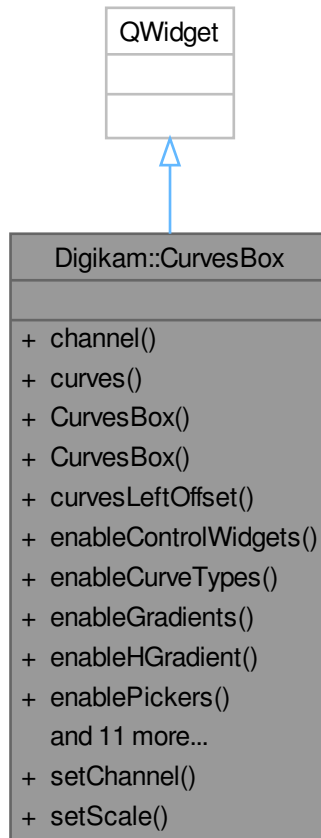


### Public Member Functions

- **CtrlButton** (const QIcon &icon, QWidget \*const parent=nullptr)

## 6.240 Digikam::CurvesBox Class Reference

Inheritance diagram for Digikam::CurvesBox:



### Public Types

- enum `ColorPicker` { `NoPicker` = -1 , `BlackTonal` = 0 , `GrayTonal` , `WhiteTonal` }
- enum `CurvesDrawingType` { `SmoothDrawing` = 0 , `FreeDrawing` }

### Public Slots

- void `setChannel` (`ChannelType` channel)
- void `setScale` ([HistogramScale](#) scale)

### Signals

- void `signalChannelReset` (int)
- void `signalCurvesChanged` ()
- void `signalCurveTypeChanged` (int)
- void `signalPickerChanged` (int)

**Public Member Functions**

- ChannelType **channel** () const
- [ImageCurves](#) \* **curves** () const
- **CurvesBox** (int w, int h, const [DImg](#) &img, QWidget \*const parent=nullptr, bool readOnly=false)
- **CurvesBox** (int w, int h, QWidget \*const parent=nullptr, bool readOnly=false)
- int **curvesLeftOffset** () const
- void **enableControlWidgets** (bool enable)
- void **enableCurveTypes** (bool enable)
- void **enableGradients** (bool enable)
- void **enableHGradient** (bool enable)
- void **enablePickers** (bool enable)
- void **enableResetButton** (bool enable)
- void **enableVGradient** (bool enable)
- int **picker** () const
- void **readCurveSettings** (KConfigGroup &group, const QString &prefix)
- void **reset** ()
- void **resetChannel** (int channel)
- void **resetChannels** ()
- void **resetPickers** ()
- void **setCurveGuide** (const [DColor](#) &color)
- void **writeCurveSettings** (KConfigGroup &group, const QString &prefix)

## 6.241 Digikam::CurvesContainer Class Reference

**Public Member Functions**

- [CurvesContainer](#) ()=default
- **CurvesContainer** (int type, bool sixteenBit)
- void **initialize** ()
- bool **isEmpty** () const
- bool **isStoredLosslessly** () const
- bool **operator==** (const [CurvesContainer](#) &other) const
- void **writeToFilterAction** ([FilterAction](#) &action, const QString &prefix=QString()) const

**Static Public Member Functions**

- static [CurvesContainer](#) **fromFilterAction** (const [FilterAction](#) &action, const QString &prefix=QString())

**Public Attributes**

- int **curvesType** = [ImageCurves::CURVE\\_SMOOTH](#)
- bool **sixteenBit** = false
- QPolygon **values** [ColorChannels]

### 6.241.1 Constructor & Destructor Documentation

#### 6.241.1.1 CurvesContainer()

```
Digikam::CurvesContainer::CurvesContainer ( ) [default]
```

Provides a convenient storage for a curve. Initially, the values are empty. Call [initialize\(\)](#) before adjusting values manually.

## 6.241.2 Member Function Documentation

### 6.241.2.1 initialize()

```
void Digikam::CurvesContainer::initialize ( )
```

Fills the values with a linear curve suitable for type and sixteenBit parameters.

### 6.241.2.2 isEmpty()

```
bool Digikam::CurvesContainer::isEmpty ( ) const
```

An empty container is interpreted as a linear curve. A non-empty container can also be linear; test for `isLinear()` of the resulting [ImageCurves](#). Note: If an [ImageCurves](#) is linear, it will return an empty container.

### 6.241.2.3 isStoredLosslessly()

```
bool Digikam::CurvesContainer::isStoredLosslessly ( ) const
```

Serialize from and to [FilterAction](#). `isStoredLosslessly` returns false if the curve cannot be losslessly stored in XML because it would be too large (free 16 bit). It is then lossily compressed.

## 6.241.3 Member Data Documentation

### 6.241.3.1 curvesType

```
int Digikam::CurvesContainer::curvesType = ImageCurves::CURVE\_SMOOTH
```

Smooth : QPolygon have size of 18 points. Free : QPolygon have size of 255 or 65535 values.

## 6.242 Digikam::CurvesFilter Class Reference

Inheritance diagram for Digikam::CurvesFilter:



### Public Member Functions

- **CurvesFilter** (const [CurvesContainer](#) &settings, [DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, [DImg](#) &destImage, int progressBegin=0, int progressEnd=100)

- **CurvesFilter** ([DImg](#) \*const orgImage, [QObject](#) \*const parent=nullptr, const [CurvesContainer](#) &settings=[CurvesContainer](#)())
- **CurvesFilter** ([QObject](#) \*const parent=nullptr)
- [FilterAction](#) [filterAction](#) () override
- [QString](#) [filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int [stop](#), int [start](#)=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int [version](#))
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const [master](#), int [progressBegin](#)=0, int [progressEnd](#)=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool [emitThem](#))
- void [setPriority](#) ([QThread::Priority](#) [priority](#))
- State [state](#) () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }



## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.242.1 Member Function Documentation

### 6.242.1.1 filterAction()

`FilterAction` Digikam::CurvesFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.242.1.2 filterIdentifier()

`QString` Digikam::CurvesFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

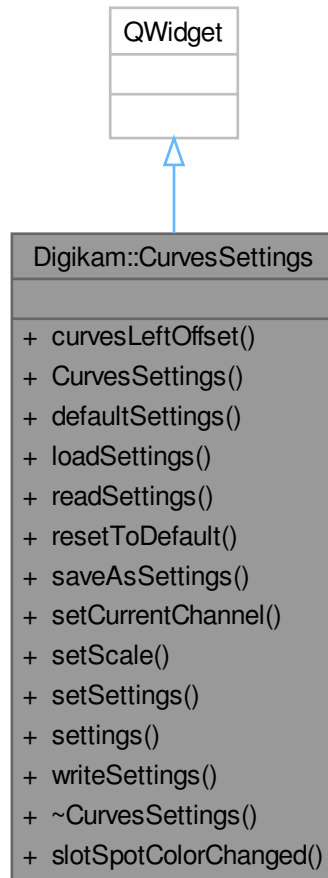
### 6.242.1.3 readParameters()

```
void Digikam::CurvesFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.243 Digikam::CurvesSettings Class Reference

Inheritance diagram for Digikam::CurvesSettings:



### Public Slots

- void `slotSpotColorChanged` (const [Digikam::DColor](#) &color)

### Signals

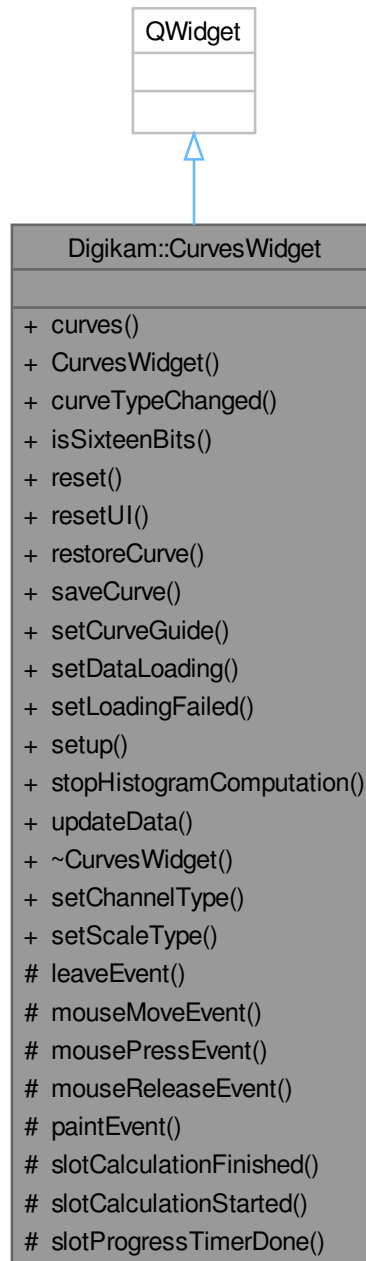
- void `signalChannelReset` (int)
- void `signalPickerChanged` (int)
- void `signalSettingsChanged` ()
- void `signalSpotColorChanged` ()

## Public Member Functions

- int **curvesLeftOffset** () const
- **CurvesSettings** (QWidget \*const parent, [DImg](#) \*const img)
- [CurvesContainer](#) **defaultSettings** () const
- void **loadSettings** ()
- void **readSettings** (KConfigGroup &group)
- void **resetToDefault** ()
- void **saveAsSettings** ()
- void **setCurrentChannel** (ChannelType channel)
- void **setScale** ([HistogramScale](#) type)
- void **setSettings** (const [CurvesContainer](#) &settings)
- [CurvesContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.244 Digikam::CurvesWidget Class Reference

Inheritance diagram for Digikam::CurvesWidget:



### Public Slots

- void **setChannelType** (ChannelType channel)
- void **setScaleType** ([HistogramScale](#) scale)

## Signals

- void **signalCurvesChanged** ()
- void **signalHistogramComputationDone** ()
- void **signalHistogramComputationFailed** ()
- void **signalMouseMoved** (int x, int y)

## Public Member Functions

- [ImageCurves](#) \* **curves** () const
- **CurvesWidget** (int w, int h, QWidget \*const parent, bool readOnly=false)
- void **curveTypeChanged** ()
- bool **isSixteenBits** () const
- void **reset** ()
- void **resetUI** ()
- void **restoreCurve** (const KConfigGroup &group, const QString &prefix)
- void **saveCurve** (KConfigGroup &group, const QString &prefix)
- void **setCurveGuide** (const [DColor](#) &color)
- void **setDataLoading** ()
- void **setLoadingFailed** ()
- void **setup** (int w, int h, bool readOnly)
- void **stopHistogramComputation** ()
- void **updateData** (const [DImg](#) &img)

## Protected Slots

- void **slotCalculationFinished** (bool success)
- void **slotCalculationStarted** ()
- void **slotProgressTimerDone** ()

## Protected Member Functions

- void **leaveEvent** (QEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override

## 6.244.1 Member Function Documentation

### 6.244.1.1 reset()

```
void Digikam::CurvesWidget::reset ( )
```

Resets the ui including the user specified curve.

### 6.244.1.2 resetUI()

```
void Digikam::CurvesWidget::resetUI ( )
```

Resets only the ui and keeps the curve.

### 6.244.1.3 restoreCurve()

```
void Digikam::CurvesWidget::restoreCurve (
    const KConfigGroup & group,
    const QString & prefix )
```

Restores the curve tfrom the given group with prefix as a prefix for the curve point config entries.

## Parameters

<i>group</i>	the group to restore the curve from
<i>prefix</i>	the prefix prepended to the point numbers in the config

**6.244.1.4 saveCurve()**

```
void Digikam::CurvesWidget::saveCurve (
    KConfigGroup & group,
    const QString & prefix )
```

Saves the currently created curve to the given group with prefix as a prefix for the curve point config entries.

## Parameters

<i>group</i>	the group to save the curve to
<i>prefix</i>	the prefix prepended to the point numbers in the config

**6.244.1.5 stopHistogramComputation()**

```
void Digikam::CurvesWidget::stopHistogramComputation ( )
```

Stop current histogram computations.

**6.244.1.6 updateData()**

```
void Digikam::CurvesWidget::updateData (
    const DImg & img )
```

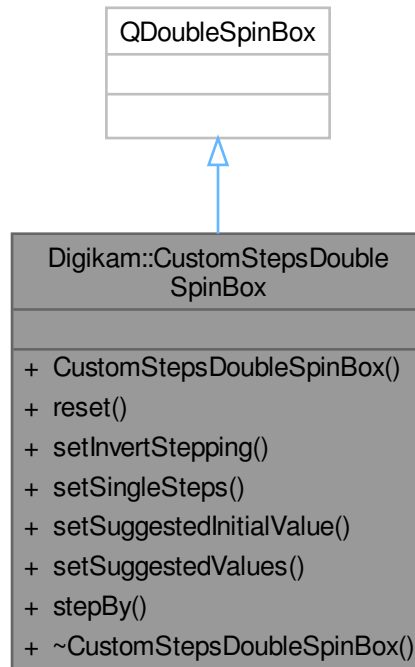
Updates the image data the curve should be used for.

## Parameters

<i>img</i>	image data
------------	------------

## 6.245 Digikam::CustomStepsDoubleSpinBox Class Reference

Inheritance diagram for Digikam::CustomStepsDoubleSpinBox:



### Public Member Functions

- [CustomStepsDoubleSpinBox](#) (`QWidget *const parent=nullptr`)
- void [reset](#) ()
- void **setInvertStepping** (`bool invert`)
- void [setSingleSteps](#) (`double smaller, double larger`)
- void [setSuggestedInitialValue](#) (`double initialValue`)
- void [setSuggestedValues](#) (`const QList< double > &values`)
- void **stepBy** (`int steps`) override

### 6.245.1 Constructor & Destructor Documentation

#### 6.245.1.1 CustomStepsDoubleSpinBox()

```

Digikam::CustomStepsDoubleSpinBox::CustomStepsDoubleSpinBox (
    QWidget *const parent = nullptr ) [explicit]
  
```

This is a normal `QDoubleSpinBox` which allows to customize the stepping behavior, for cases where linear steps are not applicable



## 6.245.2 Member Function Documentation

### 6.245.2.1 reset()

```
void Digikam::CustomStepsDoubleSpinBox::reset ( )
```

Resets to minimum value.

### 6.245.2.2 setSingleSteps()

```
void Digikam::CustomStepsDoubleSpinBox::setSingleSteps (
    double smaller,
    double larger )
```

Allows to set to different default single steps, for the range below `m_values`, the other for above.

### 6.245.2.3 setSuggestedInitialValue()

```
void Digikam::CustomStepsDoubleSpinBox::setSuggestedInitialValue (
    double initialValue )
```

Sets the value that should be set as first value when first moving away from the minimum value.

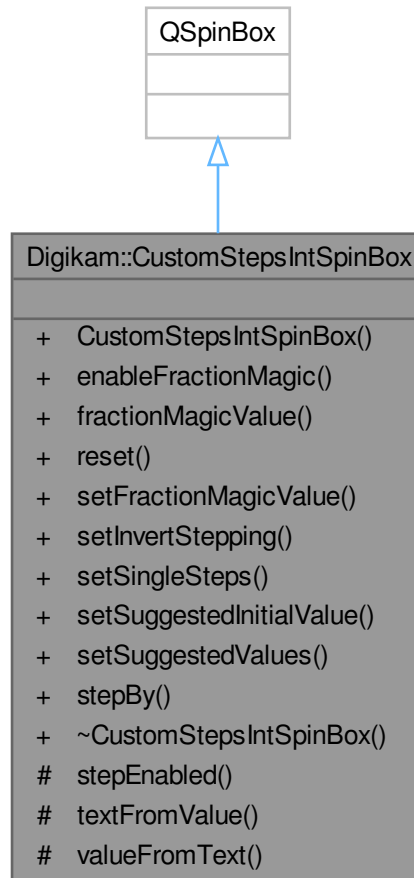
### 6.245.2.4 setSuggestedValues()

```
void Digikam::CustomStepsDoubleSpinBox::setSuggestedValues (
    const QList< double > & values )
```

Set a list of values that are usually applicable for the type of data of the combo box. The user can still type in any other value. Boundaries are not touched. Up or below the min and max values of the list given, default stepping is used.

## 6.246 Digikam::CustomStepsIntSpinBox Class Reference

Inheritance diagram for Digikam::CustomStepsIntSpinBox:



### Public Member Functions

- [CustomStepsIntSpinBox](#) (`QWidget *const parent=nullptr`)
- void [enableFractionMagic](#) (`const QString &prefix`)
- double [fractionMagicValue](#) () const
- void [reset](#) ()
- void [setFractionMagicValue](#) (`double value`)
- void [setInvertStepping](#) (`bool invert`)
- void [setSingleSteps](#) (`int smaller, int larger`)
- void [setSuggestedInitialValue](#) (`int initialValue`)
- void [setSuggestedValues](#) (`const QList< int > &values`)
- void [stepBy](#) (`int steps`) override

## Protected Member Functions

- StepEnabled **stepEnabled** () const override
- QString **textFromValue** (int value) const override
- int **valueFromText** (const QString &text) const override

## 6.246.1 Constructor & Destructor Documentation

### 6.246.1.1 CustomStepsIntSpinBox()

```
Digikam::CustomStepsIntSpinBox::CustomStepsIntSpinBox (  
    QWidget *const parent = nullptr ) [explicit]
```

This is a normal QIntSpinBox which allows to customize the stepping behavior, for cases where linear steps are not applicable

## 6.246.2 Member Function Documentation

### 6.246.2.1 enableFractionMagic()

```
void Digikam::CustomStepsIntSpinBox::enableFractionMagic (  
    const QString & prefix )
```

Call this with a fraction prefix (like "1/") to enable magic handling of the value as fraction denominator.

### 6.246.2.2 fractionMagicValue()

```
double Digikam::CustomStepsIntSpinBox::fractionMagicValue ( ) const
```

value() and setValue() for fraction magic value.

### 6.246.2.3 reset()

```
void Digikam::CustomStepsIntSpinBox::reset ( )
```

Resets to minimum value

### 6.246.2.4 setSingleSteps()

```
void Digikam::CustomStepsIntSpinBox::setSingleSteps (  
    int smaller,  
    int larger )
```

Allows to set to different default single steps, for the range below m\_values, the other for above.

### 6.246.2.5 setSuggestedInitialValue()

```
void Digikam::CustomStepsIntSpinBox::setSuggestedInitialValue (
    int initialValue )
```

Sets the value that should be set as first value when first moving away from the minimum value.

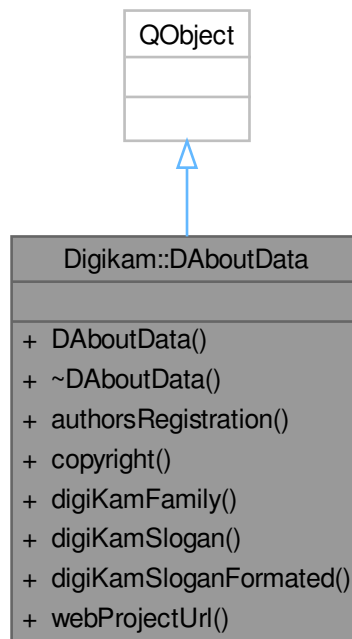
### 6.246.2.6 setSuggestedValues()

```
void Digikam::CustomStepsIntSpinBox::setSuggestedValues (
    const QList< int > & values )
```

Set a list of values that are usually applicable for the type of data of the combo box. The user can still type in any other value. Boundaries are not touched. Up or below the min and max values of the list given, default stepping is used.

## 6.247 Digikam::DAboutData Class Reference

Inheritance diagram for Digikam::DAboutData:



### Public Member Functions

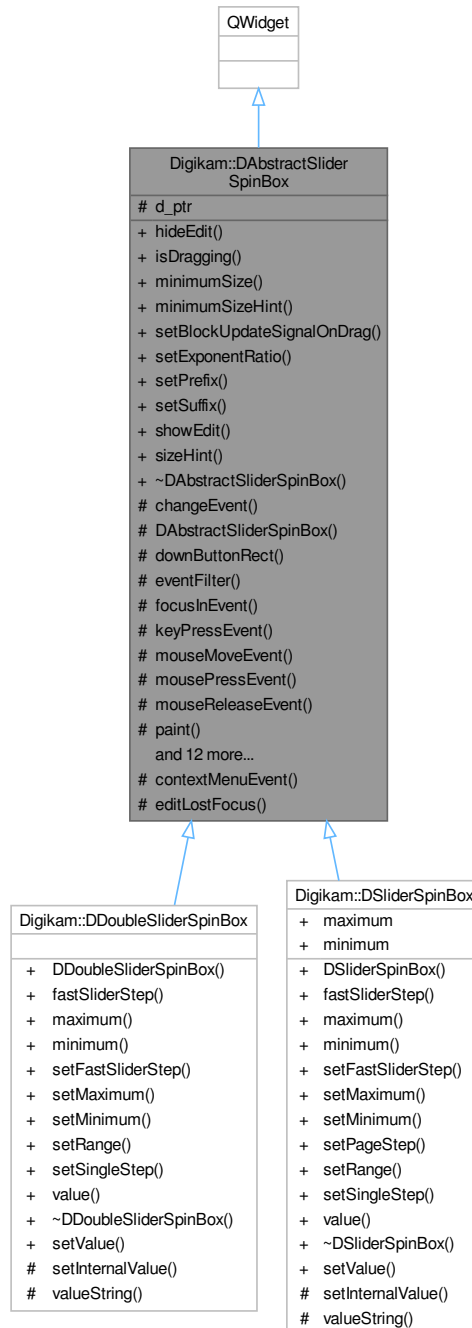
- `DAboutData` (`DXmlGuiWindow` \*const parent)

### Static Public Member Functions

- static void **authorsRegistration** (KAboutData &aboutData)
- static const QString **copyright** ()
- static const QString **digiKamFamily** ()
- static const QString **digiKamSlogan** ()
- static const QString **digiKamSloganFormatted** ()
- static const QUrl **webProjectUrl** ()

## 6.248 Digikam::DAbstractSliderSpinBox Class Reference

Inheritance diagram for Digikam::DAbstractSliderSpinBox:



### Public Member Functions

- void **hideEdit** ()
- bool **isDragging** () const

- virtual QSize **minimumSize** () const
- QSize **minimumSizeHint** () const override
- void **setBlockUpdateSignalOnDrag** (bool block)
- void **setExponentRatio** (double dbl)
- void **setPrefix** (const QString &prefix)
- void **setSuffix** (const QString &suffix)
- void **showEdit** ()
- QSize **sizeHint** () const override

### Protected Slots

- void **contextMenuEvent** (QContextMenuEvent \*event) override
- void **editLostFocus** ()

### Protected Member Functions

- void **changeEvent** (QEvent \*e) override
- **DAbstractSliderSpinBox** (QWidget \*const parent, DAbstractSliderSpinBoxPrivate \*const q)
- QRect **downButtonRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- bool **eventFilter** (QObject \*recv, QEvent \*e) override
- void **focusInEvent** (QFocusEvent \*e) override
- void **keyPressEvent** (QKeyEvent \*e) override
- void **mouseMoveEvent** (QMouseEvent \*e) override
- void **mousePressEvent** (QMouseEvent \*e) override
- void **mouseReleaseEvent** (QMouseEvent \*e) override
- void **paint** (QPainter &painter)
- void **paintBreeze** (QPainter &painter)
- void **paintEvent** (QPaintEvent \*e) override
- void **paintFusion** (QPainter &painter)
- void **paintPlastique** (QPainter &painter)
- QStyleOptionProgressBar **progressBarOptions** () const
- QRect **progressRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- virtual void **setInternalValue** (int value, bool blockUpdateSignal)=0
- QStyleOptionSpinBox **spinBoxOptions** () const
- QRect **upButtonRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- int **valueForX** (int x, Qt::KeyboardModifiers modifiers=Qt::NoModifier) const
- virtual QString **valueString** () const =0
- void **wheelEvent** (QWheelEvent \*e) override

### Protected Attributes

- DAbstractSliderSpinBoxPrivate \*const **d\_ptr**

## 6.248.1 Member Function Documentation

### 6.248.1.1 setBlockUpdateSignalOnDrag()

```
void Digikam::DAbstractSliderSpinBox::setBlockUpdateSignalOnDrag (
    bool block )
```

If set to block, it informs inheriting classes that they shouldn't emit signals if the update comes from a mouse dragging the slider. Set this to true when dragging the slider and updates during the drag are not needed.

### 6.248.1.2 setInternalValue()

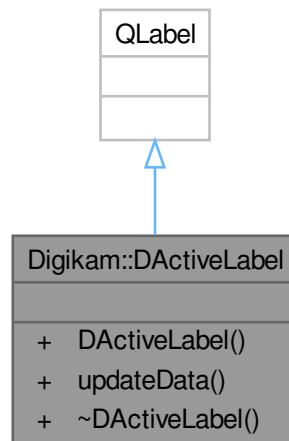
```
virtual void Digikam::DAbstractSliderSpinBox::setInternalValue (
    int value,
    bool blockUpdateSignal ) [protected], [pure virtual]
```

Sets the slider internal value. Inheriting classes should respect blockUpdateSignal so that, in specific cases, we have a performance improvement. See setIgnoreMouseMoveEvents.

Implemented in [Digikam::DSliderSpinBox](#), and [Digikam::DDoubleSliderSpinBox](#).

## 6.249 Digikam::DActiveLabel Class Reference

Inheritance diagram for Digikam::DActiveLabel:



### Public Member Functions

- **DActiveLabel** (const `QUrl` &url=`QUrl()`, const `QString` &imgPath=`QString()`, `QWidget` \*const parent=`nullptr`)
- void **updateData** (const `QUrl` &url, const `QImage` &img)

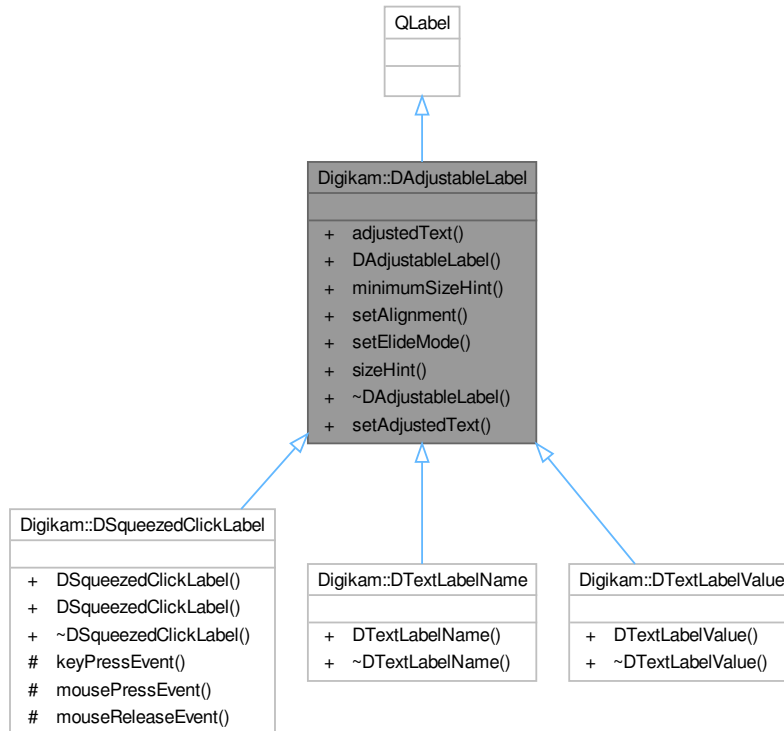
### 6.249.1 Detailed Description

A widget to host an image into a label with an active url which can be open to default web browser using simple mouse click.



## 6.250 Digikam::DAdjustableLabel Class Reference

Inheritance diagram for Digikam::DAdjustableLabel:



### Public Slots

- void **setAdjustedText** (const QString &text=QString())

### Public Member Functions

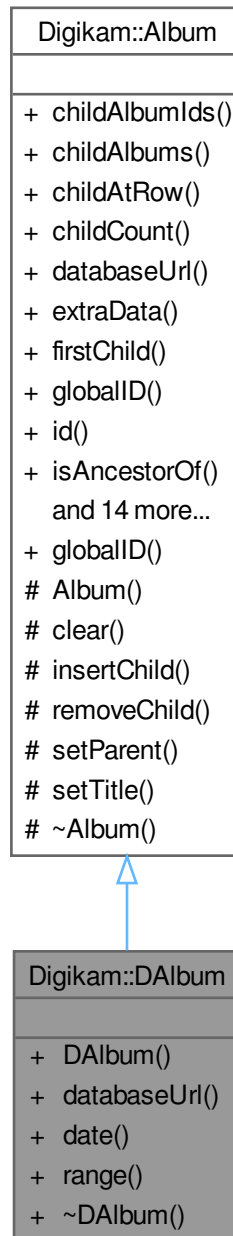
- QString **adjustedText** () const
- **DAdjustableLabel** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setAlignment** (Qt::Alignment align)
- void **setElideMode** (Qt::TextElideMode mode)
- QSize **sizeHint** () const override

### 6.250.1 Detailed Description

A label to show text adjusted to widget size

## 6.251 Digikam::DAAlbum Class Reference

Inheritance diagram for Digikam::DAAlbum:



### Public Types

- enum **Range** { **Month** = 0 , **Year** }

## Public Types inherited from Digikam::Album

- enum `Type` {  
    `PHYSICAL` = 0 , `TAG` , `DATE` , `SEARCH` ,  
    `FACE` }

## Public Member Functions

- **DAAlbum** (const `QDate` &date, bool root=false, Range range=Month)
- `CoreDbUrl databaseUrl` () const override
- `QDate date` () const
- Range `range` () const

## Public Member Functions inherited from Digikam::Album

- `QList< int >` `childAlbumIds` (bool recursive=false)
- `AlbumList` `childAlbums` (bool recursive=false)
- `Album *` `childAtRow` (int row) const
- int `childCount` () const
- void \* `extraData` (const void \*const key) const
- `Album *` `firstChild` () const
- int `globalID` () const
- int `id` () const
- bool `isAncestorOf` (`Album *`const album) const
- bool `isRoot` () const
- bool `isTrashAlbum` () const
- bool `isUsedByLabelsTree` () const
- `Album *` `lastChild` () const
- `Album *` `next` () const
- `Album *` `parent` () const
- void `prepareForDeletion` ()
- `Album *` `prev` () const
- void `removeExtraData` (const void \*const key)
- int `rowFromAlbum` () const
- void `setExtraData` (const void \*const key, void \*const value)
- void `setUsedByLabelsTree` (bool isUsed)
- `QString` `title` () const
- `Type` `type` () const

## Friends

- class `AlbumManager`

## Additional Inherited Members

## Static Public Member Functions inherited from Digikam::Album

- static int `globalID` (`Type` type, int id)  
*Produces the global id.*

## Protected Member Functions inherited from [Digikam::Album](#)

- [Album](#) ([Album::Type](#) type, int id, bool root)
- void [clear](#) ()
- void [insertChild](#) ([Album](#) \*const child)
- void [removeChild](#) ([Album](#) \*const child)
- void [setParent](#) ([Album](#) \*const [parent](#))
- void [setTitle](#) (const QString &title)
- virtual [~Album](#) ()

### 6.251.1 Detailed Description

A Date [Album](#) representation

### 6.251.2 Member Function Documentation

#### 6.251.2.1 [databaseUrl\(\)](#)

```
CoreDbUrl Digikam::DAlbum::databaseUrl ( ) const [override], [virtual]
```

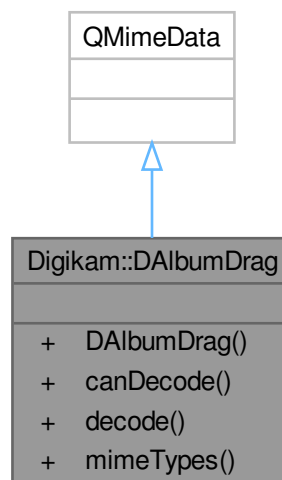
#### Returns

the kde url of the album

Implements [Digikam::Album](#).

## 6.252 [Digikam::DAlbumDrag](#) Class Reference

Inheritance diagram for [Digikam::DAlbumDrag](#):



### Public Member Functions

- **DAAlbumDrag** (const QUrl &databaseUrl, int albumid, const QUrl &fileUrl=QUrl())

### Static Public Member Functions

- static bool **canDecode** (const QMimeData \*e)
- static bool **decode** (const QMimeData \*e, QList< QUrl > &urls, int &albumID)
- static QStringList **mimeTypes** ()

## 6.252.1 Detailed Description

Provides a drag object for an album

When an album is moved through drag'n'drop an object of this class is created.

## 6.253 Digikam::DAAlbumInfo Class Reference

### Public Member Functions

- QString **albumPath** () const
- QString **caption** () const
- **DAAlbumInfo** (const [DInfoInterface::DInfoMap](#) &)
- QDate **date** () const
- QString **path** () const
- QString **title** () const

## 6.254 Digikam::DArrowClickLabel Class Reference

Inheritance diagram for Digikam::DArrowClickLabel:



### Signals

- void **leftClicked** ()

### Public Member Functions

- Qt::ArrowType **arrowType** () const
- **DArrowClickLabel** (QWidget \*const parent=nullptr)
- void **setArrowType** (Qt::ArrowType arrowType)
- QSize **sizeHint** () const override

### Protected Member Functions

- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- void **paintEvent** (QPaintEvent \*event) override

### Protected Attributes

- Qt::ArrowType **m\_arrowType** = Qt::DownArrow
- int **m\_margin** = 2
- int **m\_size** = 8

## 6.255 Digikam::DatabaseBlob Class Reference

### Public Types

- enum { **Version** = 1 }

### Public Member Functions

- void [read](#) (const QByteArray &array, [Haar::SignatureData](#) &data)
- QByteArray **write** (const [Haar::SignatureData](#) &data)

### 6.255.1 Detailed Description

This class encapsulates the Haar signature in a QByteArray that can be stored as a BLOB in the database.

Reading and writing is done in a platform-independent manner, which induces a certain overhead, but which is necessary IMO.

### 6.255.2 Member Function Documentation

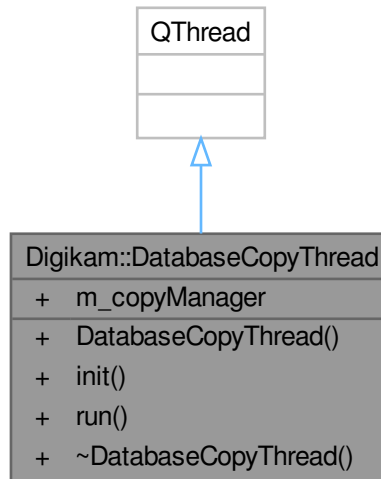
#### 6.255.2.1 read()

```
void Digikam::DatabaseBlob::read (
    const QByteArray & array,
    Haar::SignatureData & data )
```

Read the QByteArray into the [Haar::SignatureData](#).

## 6.256 Digikam::DatabaseCopyThread Class Reference

Inheritance diagram for Digikam::DatabaseCopyThread:



### Public Member Functions

- **DatabaseCopyThread** (QWidget \*const parent)
- void **init** (const [DbEngineParameters](#) &fromDatabaseSettingsWidget, const [DbEngineParameters](#) &toDatabaseSettingsWidget)
- void **run** () override

### Public Attributes

- [CoreDbCopyManager](#) **m\_copyManager**

## 6.257 Digikam::DatabaseFields::DatabaseFieldsEnumIterator<FieldName> Class Template Reference

### Public Member Functions

- bool **atEnd** () const
- FieldName **operator\*** () const
- void **operator++** ()



### 6.257.1 Detailed Description

```
template<typename FieldName>
class Digikam::DatabaseFields::DatabaseFieldsEnumIterator< FieldName >
```

You can iterate over each of the Enumerations defined above: ImagesIterator, ImageMetadataIterator etc. for  
(ImagesIterator it ; !it.atEnd() ; ++it) {...}

## 6.258 Digikam::DatabaseFields::DatabaseFieldsEnumIteratorSetOnly< FieldName > Class Template Reference

### Public Member Functions

- bool **atEnd** () const
- **DatabaseFieldsEnumIteratorSetOnly** (const FieldName setValues)
- FieldName **operator\*** () const
- void **operator++** ()

### 6.258.1 Detailed Description

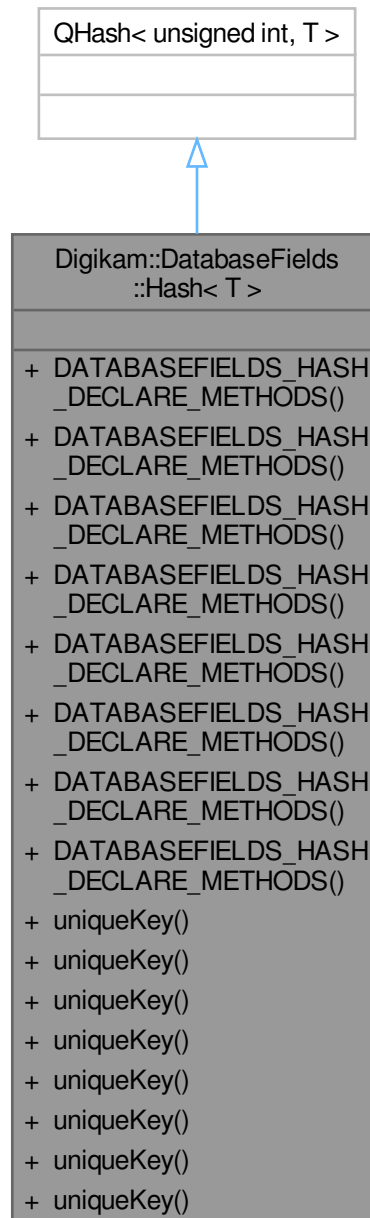
```
template<typename FieldName>
class Digikam::DatabaseFields::DatabaseFieldsEnumIteratorSetOnly< FieldName >
```

An iterator that iterates only over the flags which are set

## 6.259 Digikam::DatabaseFields::FieldMetaInfo< FieldName > Class Template Reference

## 6.260 Digikam::DatabaseFields::Hash< T > Class Template Reference

Inheritance diagram for Digikam::DatabaseFields::Hash< T >:



### Public Member Functions

- `DATABASEFIELDS_HASH_DECLARE_METHODS` (CustomEnum, uniqueKey)

- **DATABASEFIELDS\_HASH\_DECLARE\_METHODS** (ImageHistoryInfo, uniqueKey)
- **DATABASEFIELDS\_HASH\_DECLARE\_METHODS** (ImageMetadata, uniqueKey)
- **DATABASEFIELDS\_HASH\_DECLARE\_METHODS** (Images, uniqueKey)
- **DATABASEFIELDS\_HASH\_DECLARE\_METHODS** (ItemComments, uniqueKey)
- **DATABASEFIELDS\_HASH\_DECLARE\_METHODS** (ItemInformation, uniqueKey)
- **DATABASEFIELDS\_HASH\_DECLARE\_METHODS** (ItemPositions, uniqueKey)
- **DATABASEFIELDS\_HASH\_DECLARE\_METHODS** (VideoMetadata, uniqueKey)

### Static Public Member Functions

- static unsigned int **uniqueKey** (CustomEnum f)
- static unsigned int **uniqueKey** (ImageHistoryInfo f)
- static unsigned int **uniqueKey** (ImageMetadata f)
- static unsigned int **uniqueKey** (Images f)
- static unsigned int **uniqueKey** (ItemComments f)
- static unsigned int **uniqueKey** (ItemInformation f)
- static unsigned int **uniqueKey** (ItemPositions f)
- static unsigned int **uniqueKey** (VideoMetadata f)

### 6.260.1 Detailed Description

```
template<class T>
class Digikam::DatabaseFields::Hash< T >
```

This class provides a hash on all DatabaseFields enums, allowing to use the enum values as independent keys. You can use the class like a normal QHash with the value type defined by you, and as keys the members of the DatabaseFields enums. You can only use single enum members as keys, not or'ed numbers. You can use one custom enum, cast to DatabaseFields::CustomEnum, which can have at most 26 flag values ( $1 \ll 0$  to  $1 \ll 26$ ). Pass this as the optional second template parameter.

## 6.261 Digikam::DatabaseFields::Set Class Reference

### Public Member Functions

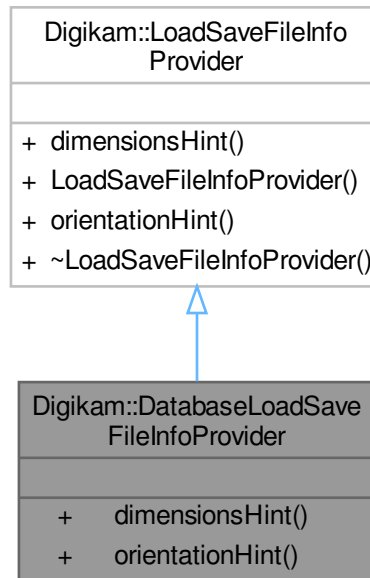
- void **initialize** ()
- bool **operator&** (const [Set](#) &other)
- CustomEnum **operator&** (CustomEnum f) const
- CustomEnum & **operator=** (const CustomEnum &f)
- CustomEnum **operator^** (CustomEnum f) const
- CustomEnum & **operator^=** (CustomEnum f)
- CustomEnum **operator|** (CustomEnum f) const
- CustomEnum & **operator|=** (CustomEnum f)
- [Set](#) & **setFields** (const [Set](#) &otherSet)

### 6.261.1 Detailed Description

This class provides a set of all DatabaseFields enums, without resorting to a QSet.

## 6.262 Digikam::DatabaseLoadSaveFileInfoProvider Class Reference

Inheritance diagram for Digikam::DatabaseLoadSaveFileInfoProvider:



### Public Member Functions

- QSize [dimensionsHint](#) (const QString &path) override
- int [orientationHint](#) (const QString &path) override

### 6.262.1 Member Function Documentation

#### 6.262.1.1 dimensionsHint()

```
QSize Digikam::DatabaseLoadSaveFileInfoProvider::dimensionsHint (
    const QString & path ) [override], [virtual]
```

Gives a hint at the size of the image. This can be used to supersede the Exif information in the file.

Implements [Digikam::LoadSaveFileInfoProvider](#).

#### 6.262.1.2 orientationHint()

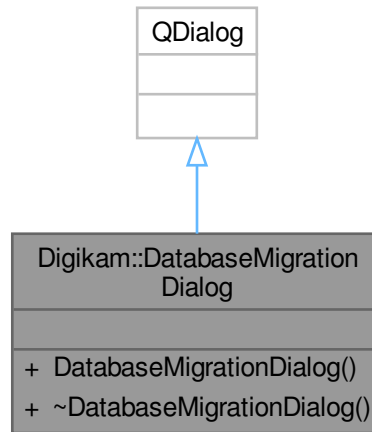
```
int Digikam::DatabaseLoadSaveFileInfoProvider::orientationHint (
    const QString & path ) [override], [virtual]
```

Gives a hint at the orientation of the image. This can be used to supersede the Exif information in the file. Will not be used if DMetadata::ORIENTATION\_UNSPECIFIED (default value)

Implements [Digikam::LoadSaveFileInfoProvider](#).

## 6.263 Digikam::DatabaseMigrationDialog Class Reference

Inheritance diagram for Digikam::DatabaseMigrationDialog:



### Public Member Functions

- **DatabaseMigrationDialog** (QWidget \*const parent)

## 6.264 Digikam::DatabaseOption Class Reference

Inheritance diagram for Digikam::DatabaseOption:



### Protected Member Functions

- QString `parseOperation` (`ParseSettings &settings`, const `QRegularExpressionMatch &match`) override

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void [setDescription](#) (const QString &desc)
- void [setIcon](#) (const QString &pixmap)
- void [setRegExp](#) (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum [IconType](#) { [Action](#) = 0 , [Dialog](#) }

## Signals inherited from [Digikam::Rule](#)

- void [signalTokenTriggered](#) (const QString &)

## Public Member Functions inherited from [Digikam::Option](#)

- [Option](#) (const QString &name, const QString &description)
- [Option](#) (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString [description](#) () const
- QPixmap [icon](#) (Rule::IconType type=Rule::Action) const
- bool [isValid](#) () const
- [ParseResults](#) [parse](#) ([ParseSettings](#) &settings)
- QRegularExpression & [regExp](#) () const
- QPushButton \* [registerButton](#) (QWidget \*parent)
- QAction \* [registerMenu](#) (QMenu \*parent)
- virtual void [reset](#) ()
- [Rule](#) (const QString &name)
- [Rule](#) (const QString &name, const QString &icon)
- TokenList & [tokens](#) () const
- bool [useTokenMenu](#) () const

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString [escapeToken](#) (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void [slotTokenTriggered](#) (const QString &)

### 6.264.1 Member Function Documentation

#### 6.264.1.1 [parseOperation\(\)](#)

```
QString Digikam::DatabaseOption::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [virtual]
```

TODO: describe me

## Parameters

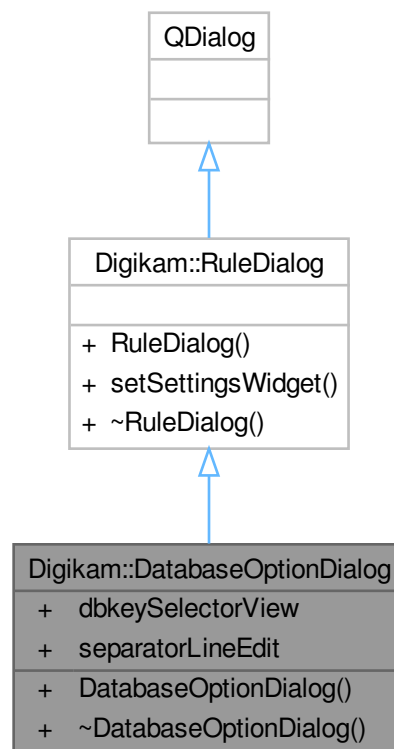
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Option](#).

## 6.265 Digikam::DatabaseOptionDialog Class Reference

Inheritance diagram for Digikam::DatabaseOptionDialog:



### Public Member Functions

- `DatabaseOptionDialog` ([Rule](#) \*const parent)



## Public Member Functions inherited from [Digikam::RuleDialog](#)

- **RuleDialog** ([Rule](#) \*const parent)
- void **setSettingsWidget** (QWidget \*const settingsWidget)

## Public Attributes

- [DbKeySelectorView](#) \* **dbkeySelectorView** = nullptr
- QLineEdit \* **separatorLineEdit** = nullptr

## 6.266 Digikam::DatabasePage Class Reference

Inheritance diagram for Digikam::DatabasePage:



### Public Member Functions

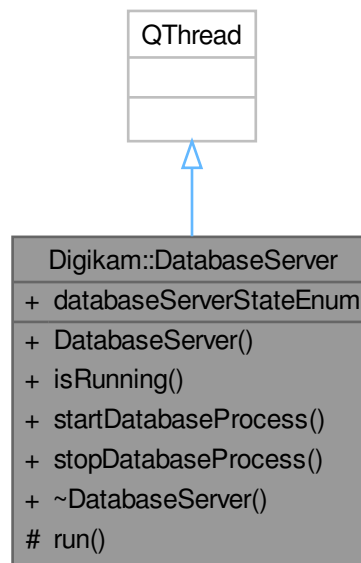
- bool **checkSettings** ()
- **DatabasePage** (QWizard \*const dlg)
- **DbEngineParameters** **getDbEngineParameters** () const
- void **saveSettings** ()
- void **setDatabasePath** (const QString &path)

## Public Member Functions inherited from Digikam::DWizardPage

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)
- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.267 Digikam::DatabaseServer Class Reference

Inheritance diagram for Digikam::DatabaseServer:



### Public Types

- enum **DatabaseServerStateEnum** { **started** , **running** , **notRunning** , **stopped** }

### Signals

- void **done** ()

### Public Member Functions

- **DatabaseServer** (const [DbEngineParameters](#) &params, [DatabaseServerStarter](#) \*const parent=[DatabaseServerStarter::instance](#))
- bool [isRunning](#) () const
- [DatabaseServerError](#) [startDatabaseProcess](#) ()
- void [stopDatabaseProcess](#) ()

### Public Attributes

- DatabaseServerStateEnum [databaseServerStateEnum](#)

### Protected Member Functions

- void [run](#) () override

## 6.267.1 Member Function Documentation

### 6.267.1.1 isRunning()

```
bool Digikam::DatabaseServer::isRunning ( ) const
```

Returns true if the server process is running.

### 6.267.1.2 startDatabaseProcess()

```
DatabaseServerError Digikam::DatabaseServer::startDatabaseProcess ( )
```

Starts the database management server.

### 6.267.1.3 stopDatabaseProcess()

```
void Digikam::DatabaseServer::stopDatabaseProcess ( )
```

Terminates the databaser server process.

## 6.268 Digikam::DatabaseServerError Class Reference

### Public Types

- enum [DatabaseServerErrorEnum](#) { [NoErrors](#) = 0 , [NotSupported](#) , [StartError](#) }

### Public Member Functions

- **DatabaseServerError** (const [DatabaseServerError](#) &dbServerError)
- **DatabaseServerError** ([DatabaseServerErrorEnum](#) errorType=[NoErrors](#), const QString &errorText=QString())
- QString [getErrorText](#) () const
- [DatabaseServerErrorEnum](#) [getErrorType](#) () const
- void [setErrorText](#) (const QString &errorText)
- void [setErrorType](#) ([DatabaseServerErrorEnum](#) errorType)

## 6.268.1 Member Enumeration Documentation

### 6.268.1.1 DatabaseServerErrorEnum

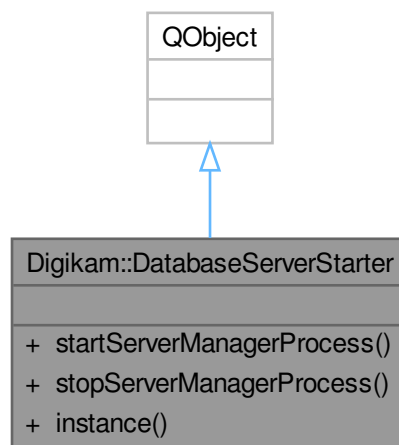
```
enum Digikam::DatabaseServerError::DatabaseServerErrorEnum
```

## Enumerator

NoErrors	No errors occurred while starting the database server
NotSupported	The requested database type is not supported.
StartError	A error has occurred while starting the database server executable.

## 6.269 Digikam::DatabaseServerStarter Class Reference

Inheritance diagram for Digikam::DatabaseServerStarter:



### Public Member Functions

- `DatabaseServerError startServerManagerProcess` (const `DbEngineParameters` &parameters) const
- void `stopServerManagerProcess` ()

### Static Public Member Functions

- static `DatabaseServerStarter * instance` ()

### Friends

- class `DatabaseServerStarterCreator`

## 6.269.1 Member Function Documentation

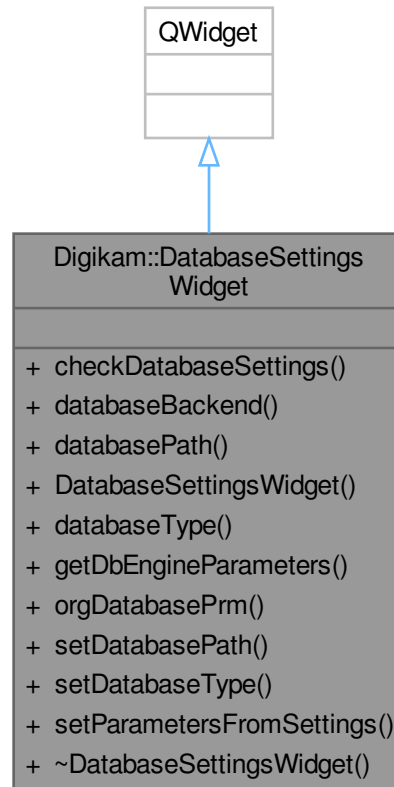
### 6.269.1.1 instance()

```
DatabaseServerStarter * Digikam::DatabaseServerStarter::instance ( ) [static]
```

Global instance of internal server starter. All accessor methods are thread-safe.

## 6.270 Digikam::DatabaseSettingsWidget Class Reference

Inheritance diagram for Digikam::DatabaseSettingsWidget:



### Classes

- class [Private](#)

### Public Types

- enum `DatabaseType` { `SQLite = 0` , `MySqlInternal = 1` , `MySqlServer = 2` }

### Public Member Functions

- bool `checkDatabaseSettings` ()
- QString `databaseBackend` () const
- QString `databasePath` () const
- `DatabaseSettingsWidget` (`QWidget *const parent=nullptr`)
- int `databaseType` () const
- `DbEngineParameters` `getDbEngineParameters` () const
- `DbEngineParameters` `orgDatabasePrm` () const
- void `setDatabasePath` (const QString &path)
- void `setDatabaseType` (int type)
- void `setParametersFromSettings` (const [ApplicationSettings](#) \*const settings, const bool &migration=false)

## 6.270.1 Member Function Documentation

### 6.270.1.1 checkDatabaseSettings()

```
bool Digikam::DatabaseSettingsWidget::checkDatabaseSettings ( )
```

For Sqlite or MysqlInternal, check properties of local path to store database files. For MysqlServer, check the network connection and database names.

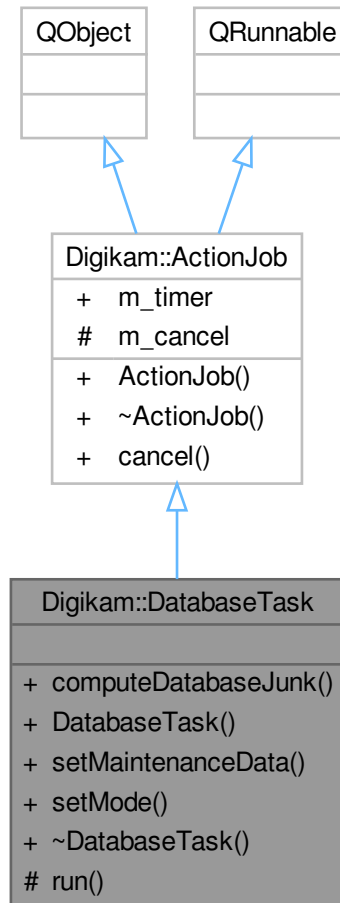
## 6.271 Digikam::DatabaseSettingsWidget::Private Class Reference

### Public Attributes

- QLineEdit \* **connectOpts** = nullptr
- [DBinarySearch](#) \* **dbBinariesWidget** = nullptr
- QGroupBox \* **dbDetailsBox** = nullptr
- QLineEdit \* **dbNameCore** = nullptr
- QLineEdit \* **dbNameFace** = nullptr
- QLineEdit \* **dbNameSimilarity** = nullptr
- [DFileSelector](#) \* **dbNameThumbs** = nullptr
- QGroupBox \* **dbNoticeBox** = nullptr
- [DFileSelector](#) \* **dbPathEdit** = nullptr
- QLabel \* **dbPathLabel** = nullptr
- QLabel \* **dbThumbsLabel** = nullptr
- QComboBox \* **dbType** = nullptr
- QMap< int, int > **dbTypeMap**
- QGroupBox \* **expertSettings** = nullptr
- QLineEdit \* **hostName** = nullptr
- QSpinBox \* **hostPort** = nullptr
- QGroupBox \* **ignoreDirectoriesBox** = nullptr
- QLineEdit \* **ignoreDirectoriesEdit** = nullptr
- QLabel \* **ignoreDirectoriesLabel** = nullptr
- [MysqlAdminBinary](#) **mysqlAdminBin**
- [DVBox](#) \* **mysqlCmdBox** = nullptr
- [MysqlInitBinary](#) **mysqlInitBin**
- [MysqlServerBinary](#) **mysqlServerBin**
- [MysqlUpgradeBinary](#) **mysqlUpgradeBin**
- [DbEngineParameters](#) **orgPrms**
- QLineEdit \* **password** = nullptr
- QTextBrowser \* **sqlInit** = nullptr
- QTabWidget \* **tab** = nullptr
- QLineEdit \* **userName** = nullptr
- QLabel \* **walLabel** = nullptr
- QCheckBox \* **walModeCheck** = nullptr

## 6.272 Digikam::DatabaseTask Class Reference

Inheritance diagram for Digikam::DatabaseTask:



### Public Types

- enum **Mode** {
  - Unknown**
  - ComputeDatabaseJunk**
  - CleanCoreDb**
  - CleanThumbsDb**
  - CleanRecognitionDb**
  - CleanSimilarityDb**
  - ShrinkDatabases**

### Signals

- void [signalAddItemsToProcess](#) (int count)
- void **signalData** (const QList< qlonglong > &staleImagelds, const QList< int > &staleThumblds, const QList< [Identity](#) > &staleIdentities, const QList< qlonglong > &staleSimilarityImagelds)
- void **signalFinished** ()
- void **signalFinished** (bool done, bool errorFree)



## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Member Functions

- void **computeDatabaseJunk** (bool thumbsDb=false, bool facesDb=false, bool similarityDb=false)
- void **setMaintenanceData** ([MaintenanceData](#) \*const data=nullptr)
- void **setMode** (Mode mode)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void **run** () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.272.1 Member Function Documentation

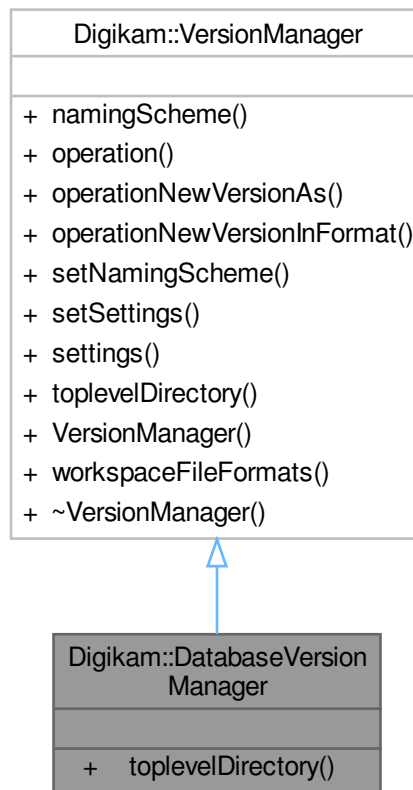
### 6.272.1.1 [signalAddItemsToProcess](#)

```
void Digikam::DatabaseTask::signalAddItemsToProcess (  
    int count ) [signal]
```

Signal to emit the count of additional items to process.

## 6.273 Digikam::DatabaseVersionManager Class Reference

Inheritance diagram for Digikam::DatabaseVersionManager:



### Public Member Functions

- `QString toplevelDirectory (const QString &path)` override

### Public Member Functions inherited from [Digikam::VersionManager](#)

- `VersionNamingScheme * namingScheme ()` const
- `VersionFileOperation operation (FileNameType request, const VersionFileInfo &loadedFile, const DImageHistory &initialResolvedHistory, const DImageHistory &currentHistory)`
- `VersionFileOperation operationNewVersionAs (const VersionFileInfo &loadedFile, const VersionFileInfo &saveLocation, const DImageHistory &initialResolvedHistory, const DImageHistory &currentHistory)`
- `VersionFileOperation operationNewVersionInFormat (const VersionFileInfo &loadedFile, const QString &format, const DImageHistory &initialResolvedHistory, const DImageHistory &currentHistory)`
- void `setNamingScheme (VersionNamingScheme *scheme)`
- void `setSettings (const VersionManagerSettings &settings)`
- `VersionManagerSettings settings ()` const
- virtual `QStringList workspaceFileFormats ()` const

### Additional Inherited Members

### Public Types inherited from [Digikam::VersionManager](#)

- enum `FileNameType` { `CurrentVersionName` , `NewVersionName` }

## 6.273.1 Member Function Documentation

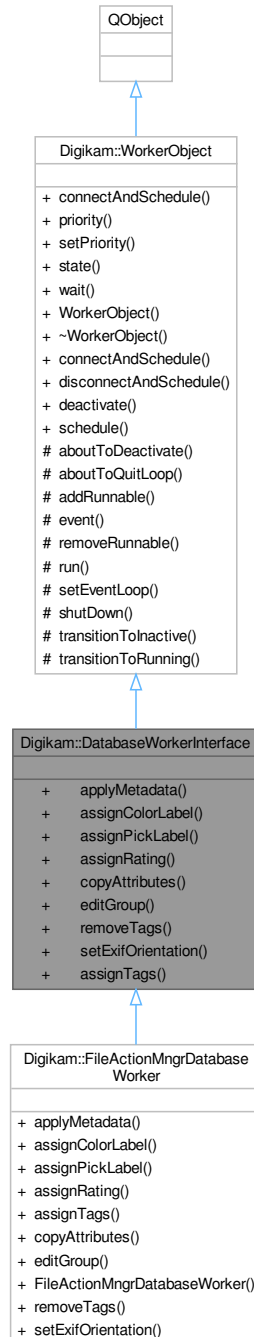
### 6.273.1.1 toplevelDirectory()

```
QString Digikam::DatabaseVersionManager::toplevelDirectory (
    const QString & path ) [inline], [override], [virtual]
```

Reimplemented from [Digikam::VersionManager](#).

## 6.274 Digikam::DatabaseWorkerInterface Class Reference

Inheritance diagram for Digikam::DatabaseWorkerInterface:



### Public Slots

- virtual void **assignTags** (const [FileActionItemInfoList](#) &, const `QList< int >` &)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

## Signals

- void [writeMetadata](#) ([FileActionItemInfoList](#) infos, int flag)
- void [writeMetadataToFiles](#) ([FileActionItemInfoList](#) infos)
- void [writeOrientationToFiles](#) ([FileActionItemInfoList](#) infos, int orientation)

## Signals inherited from [Digikam::WorkerObject](#)

- void [finished](#) ()
- void [started](#) ()

## Public Member Functions

- virtual void [applyMetadata](#) (const [FileActionItemInfoList](#) &, [DisjointMetadata](#) \*)
- virtual void [assignColorLabel](#) (const [FileActionItemInfoList](#) &, int)
- virtual void [assignPickLabel](#) (const [FileActionItemInfoList](#) &, int)
- virtual void [assignRating](#) (const [FileActionItemInfoList](#) &, int)
- virtual void [copyAttributes](#) (const [FileActionItemInfoList](#) &, const [QStringList](#) &)
- virtual void [editGroup](#) (int, const [ItemInfo](#) &, const [FileActionItemInfoList](#) &)
- virtual void [removeTags](#) (const [FileActionItemInfoList](#) &, const [QList](#)< int > &)
- virtual void [setExifOrientation](#) (const [FileActionItemInfoList](#) &, int)

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const char \*method, [Qt::](#)↔[ConnectionType](#) type=[Qt::AutoConnection](#)) const
- [QThread::Priority](#) [priority](#) () const
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- void [wait](#) ()
- [WorkerObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, [Qt::](#)[ConnectionType](#) type=[Qt::AutoConnection](#))
- static bool [disconnectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## 6.275 Digikam::DatabaseWriter Class Reference

Inheritance diagram for Digikam::DatabaseWriter:



### Public Slots

- void **process** (const FacePipelineExtendedPackage::Ptr &package)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

## Signals

- void **processed** (const [FacePipelineExtendedPackage::Ptr](#) &package)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Public Member Functions

- **DatabaseWriter** ([FacePipeline::WriteMode](#) wmode, [FacePipeline::Private](#) \*const dd)

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const char \*method, [Qt::](#)↳ [ConnectionType](#) type=[Qt::AutoConnection](#)) const
- [QThread::Priority](#) **priority** () const
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Protected Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- [FacePipeline::WriteMode](#) **mode** = [FacePipeline::NormalWrite](#)
- [ThumbnailLoadThread](#) \* **thumbnailLoadThread** = nullptr

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, [Qt::ConnectionType](#) type=[Qt::AutoConnection](#))
- static bool **disconnectAndSchedule** (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

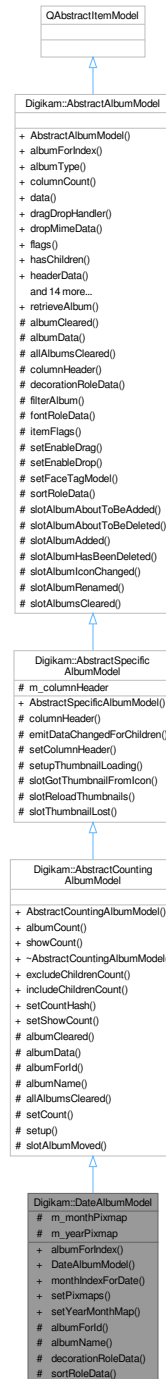


## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## 6.276 Digikam::DateAlbumModel Class Reference

Inheritance diagram for Digikam::DateAlbumModel:



### Public Slots

- void **setYearMonthMap** (const QMap< YearMonth, int > &yearMonthMap)

## Public Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [excludeChildrenCount](#) (const QModelIndex &index)
- void [includeChildrenCount](#) (const QModelIndex &index)
- void [setCountHash](#) (const QHash< int, int > &idCountHash)
- void [setShowCount](#) (bool show)

*Call to enable or disable showing the count. Default is false.*

## Public Member Functions

- [DAlbum](#) \* [albumForIndex](#) (const QModelIndex &index) const
- [DateAlbumModel](#) (QObject \*const parent=nullptr)
- QModelIndex [monthIndexForDate](#) (const QDate &date) const
- void [setPixmaps](#) (const QPixmap &forYearAlbums, const QPixmap &forMonthAlbums)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- [AbstractCountingAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), QObject \*const parent=nullptr)  
*Supports displaying a count alongside the album name in DisplayRole.*
- virtual int [albumCount](#) ([Album](#) \*album) const
- bool [showCount](#) () const

## Public Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- [AbstractSpecificAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), QObject \*const parent=nullptr)  
*Abstract base class, do not instantiate.*

## Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- [AbstractAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), QObject \*const parent=nullptr)
- [Album](#) \* [albumForIndex](#) (const QModelIndex &index) const
- [Album::Type](#) [albumType](#) () const
- int [columnCount](#) (const QModelIndex &parent=QModelIndex()) const override
- QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const override
- [AlbumModelDragDropHandler](#) \* [dragDropHandler](#) () const
- bool [dropMimeData](#) (const QMimeData \*data, Qt::DropAction action, int row, int column, const QModelIndex &parent) override
- Qt::ItemFlags [flags](#) (const QModelIndex &index) const override
- bool [hasChildren](#) (const QModelIndex &parent=QModelIndex()) const override
- QVariant [headerData](#) (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- QModelIndex [index](#) (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex [indexForAlbum](#) ([Album](#) \*album) const
- bool [isFaceTagModel](#) () const
- QMimeData \* [mimeData](#) (const QModelIndexList &indexes) const override
- QStringList [mimeTypes](#) () const override
- QModelIndex [parent](#) (const QModelIndex &index) const override
- [Album](#) \* [rootAlbum](#) () const
- [RootAlbumBehavior](#) [rootAlbumBehavior](#) () const
- QModelIndex [rootAlbumIndex](#) () const
- int [rowCount](#) (const QModelIndex &parent=QModelIndex()) const override
- void [setDragDropHandler](#) ([AlbumModelDragDropHandler](#) \*handler)
- void [setDropIndex](#) (const QModelIndex &index)
- Qt::DropActions [supportedDropActions](#) () const override

### Protected Member Functions

- Album \* albumForId (int id) const override  
*need to implement in subclass*
- QString albumName (Album \*a) const override  
*Can reimplement in subclass.*
- QVariant decorationRoleData (Album \*a) const override  
*For subclassing convenience: A part of the implementation of data()*
- QVariant sortRoleData (Album \*a) const override  
*For subclassing convenience: A part of the implementation of data()*

### Protected Member Functions inherited from Digikam::AbstractCountingAlbumModel

- void albumCleared (Album \*album) override  
*Notification when an entry is removed.*
- QVariant albumData (Album \*a, int role) const override  
*Reimplemented from parent classes.*
- void allAlbumsCleared () override  
*Notification when all entries are removed.*
- void setCount (Album \*album, int count)  
*If you do not use setCountHash, excludeChildrenCount and includeChildrenCount, you can set a count here.*
- void setup ()

### Protected Member Functions inherited from Digikam::AbstractSpecificAlbumModel

- QString columnHeader () const override  
*For subclassing convenience: A part of the implementation of headerData()*
- void emitDataChangedForChildren (Album \*album)
- void setColumnHeader (const QString &header)
- void setupThumbnailLoading ()  
*You need to call this from your constructor if you intend to load the thumbnail facilities of this class.*

### Protected Member Functions inherited from Digikam::AbstractAlbumModel

- virtual bool filterAlbum (Album \*album) const
- virtual QVariant fontRoleData (Album \*a) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual Qt::ItemFlags itemFlags (Album \*album) const  
*For subclassing convenience: A part of the implementation of itemFlags()*
- void setEnableDrag (bool enable)
- void setEnableDrop (bool enable)
- void setFaceTagModel (bool enable)

### Protected Attributes

- QPixmap m\_monthPixmap
- QPixmap m\_yearPixmap

## Protected Attributes inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString `m_columnHeader`

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractAlbumModel](#)

- enum `AlbumDataRole` {  
    `AlbumTitleRole` = Qt::UserRole , `AlbumTypeRole` = Qt::UserRole + 1 , `AlbumPointerRole` = Qt::UserRole + 2  
    , `AlbumIdRole` = Qt::UserRole + 3 ,  
    `AlbumGlobalIdRole` = Qt::UserRole + 4 , `AlbumSortRole` = Qt::UserRole + 5 }
- enum `RootAlbumBehavior` { `IncludeRootAlbum` , `IgnoreRootAlbum` }

## Signals inherited from [Digikam::AbstractCountingAlbumModel](#)

- void `signalUpdateAlbumCount` (`Album *album`)

## Signals inherited from [Digikam::AbstractAlbumModel](#)

- void `rootAlbumAvailable` ()

## Static Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- static `Album * retrieveAlbum` (const `QModelIndex &index`)

## Protected Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void `slotAlbumMoved` (`Album *album`)

## Protected Slots inherited from [Digikam::AbstractSpecificAlbumModel](#)

- void `slotGotThumbnailFromIcon` (`Album *album`, const `QPixmap &thumbnail`)
- void `slotReloadThumbnails` ()
- void `slotThumbnailLost` (`Album *album`)

## Protected Slots inherited from [Digikam::AbstractAlbumModel](#)

- void `slotAlbumAboutToBeAdded` (`Album *album`, `Album *parent`, `Album *prev`)
- void `slotAlbumAboutToBeDeleted` (`Album *album`)
- void `slotAlbumAdded` (`Album *`)
- void `slotAlbumHasBeenDeleted` (`Album *album`)
- void `slotAlbumIconChanged` (`Album *album`)
- void `slotAlbumRenamed` (`Album *album`)
- void `slotAlbumsCleared` ()

## 6.276.1 Detailed Description

A model for date based albums.

## 6.276.2 Constructor & Destructor Documentation

### 6.276.2.1 DateAlbumModel()

```
Digikam::DateAlbumModel::DateAlbumModel (  
    QObject *const parent = nullptr ) [explicit]
```

Constructor.

## Parameters

<i>parent</i>	the parent for Qt's parent child mechanism
---------------	--

## 6.276.3 Member Function Documentation

### 6.276.3.1 albumForId()

```
Album * Digikam::DateAlbumModel::albumForId (
    int id ) const [override], [protected], [virtual]
```

Implements [Digikam::AbstractCountingAlbumModel](#).

### 6.276.3.2 albumName()

```
QString Digikam::DateAlbumModel::albumName (
    Album * a ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractCountingAlbumModel](#).

### 6.276.3.3 decorationRoleData()

```
QVariant Digikam::DateAlbumModel::decorationRoleData (
    Album * a ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).

### 6.276.3.4 monthIndexForDate()

```
QModelIndex Digikam::DateAlbumModel::monthIndexForDate (
    const QDate & date ) const
```

Finds an album index based on a date. The given date is therefore normalized to year-month-form. The day is ignored. This means the returned index always points to a month [DAlbum](#).

## Parameters

<i>date</i>	the date to search for (year and month)
-------------	---

## Returns

model index corresponding to the album with the given date or an empty index if not found

### 6.276.3.5 setPixmap()

```
void Digikam::DateAlbumModel::setPixmap (
```

```
const QPixmap & forYearAlbums,  
const QPixmap & forMonthAlbums )
```

Set pixmaps for the DecorationRole

### 6.276.3.6 sortRoleData()

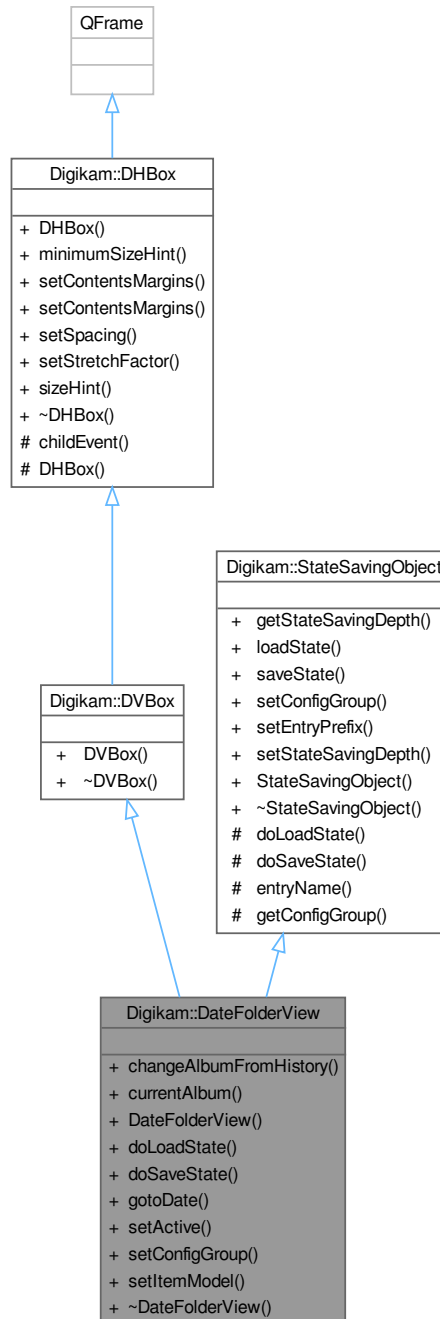
```
QVariant Digikam::DateAlbumModel::sortRoleData (  
    Album * a ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).



## 6.277 Digikam::DateFolderView Class Reference

Inheritance diagram for Digikam::DateFolderView:



### Public Member Functions

- void **changeAlbumFromHistory** (`DAAlbum *const album`)
- `AlbumPointer`< `DAAlbum` > **currentAlbum** () const

- **DateFolderView** (QWidget \*const parent, [DateAlbumModel](#) \*const dateAlbumModel)
- void **doLoadState** () override
- void **doSaveState** () override
- void **gotoDate** (const QDate &dt)
- void **setActive** (const bool val)
- void **setConfigGroup** (const KConfigGroup &group) override
- void **setItemModel** ([ItemFilterModel](#) \*const model)

### Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

### Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString **entryName** (const QString &base) const
- KConfigGroup **getConfigGroup** () const

## 6.277.1 Member Function Documentation

### 6.277.1.1 doLoadState()

```
void Digikam::DateFolderView::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.277.1.2 doSaveState()

```
void Digikam::DateFolderView::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.277.1.3 setConfigGroup()

```
void Digikam::DateFolderView::setConfigGroup (
    const KConfigGroup & group ) [override], [virtual]
```

Sets a dedicated config group that will be used to store and reload the state from. If this method is not called, a group based on the object name is used.

You can re-implement this method to pass the group set here to child objects. Don't forget to call this method in your implementation.

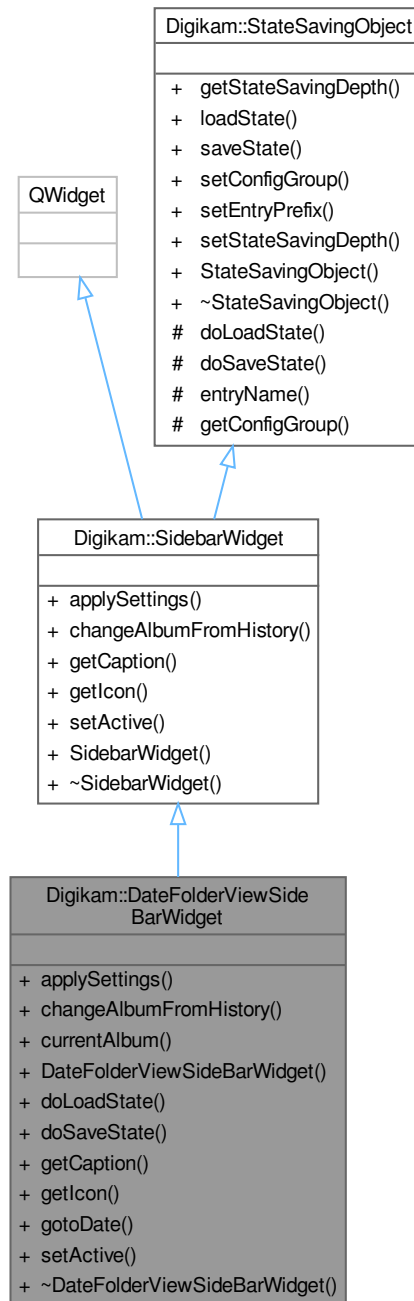
#### Parameters

<i>group</i>	config group to use for state saving and restoring
--------------	--

Reimplemented from [Digikam::StateSavingObject](#).

## 6.278 Digikam::DateFolderViewSideBarWidget Class Reference

Inheritance diagram for Digikam::DateFolderViewSideBarWidget:



### Public Member Functions

- void [applySettings](#) () override
- void [changeAlbumFromHistory](#) (const QList< [Album](#) \* > &album) override

- [AlbumPointer](#)< [DAAlbum](#) > **currentAlbum** () const
- **DateFolderViewSideBarWidget** (QWidget \*const parent, [DateAlbumModel](#) \*const model, [ItemAlbumFilterModel](#) \*const imageFilterModel)
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- void **gotoDate** (const QDate &date)
- void [setActive](#) (bool active) override

### Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

### Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.278.1 Member Function Documentation

### 6.278.1.1 [applySettings\(\)](#)

```
void Digikam::DateFolderViewSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).

### 6.278.1.2 `changeAlbumFromHistory()`

```
void Digikam::DateFolderViewSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.278.1.3 `doLoadState()`

```
void Digikam::DateFolderViewSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.278.1.4 `doSaveState()`

```
void Digikam::DateFolderViewSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.278.1.5 `getCaption()`

```
const QString Digikam::DateFolderViewSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.278.1.6 `getIcon()`

```
const QIcon Digikam::DateFolderViewSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.278.1.7 `setActive()`

```
void Digikam::DateFolderViewSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.279 Digikam::DateFormat Class Reference

### Public Types

- typedef QPair< QString, QVariant > **DateFormatDescriptor**
- typedef QList< DateFormatDescriptor > **DateFormatMap**
- enum **Type** {  
    **Standard** = 0 , **ISO** , **FullText** , **UnixTimeStamp** ,  
    **Custom** }

### Public Member Functions

- QVariant **format** (const QString &identifier)
- QVariant **format** (Type type)
- QString **identifier** (Type type)
- DateFormatMap & **map** ()
- Type **type** (const QString &identifier)

## 6.280 Digikam::DateOption Class Reference

Inheritance diagram for Digikam::DateOption:



### Protected Member Functions

- QString `parseOperation` (`ParseSettings &settings`, const `QRegularExpressionMatch &match`) override



## Protected Member Functions inherited from [Digikam::Rule](#)

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void [setDescription](#) (const QString &desc)
- void [setIcon](#) (const QString &pixmap)
- void [setRegExp](#) (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum [IconType](#) { [Action](#) = 0 , [Dialog](#) }

## Signals inherited from [Digikam::Rule](#)

- void [signalTokenTriggered](#) (const QString &)

## Public Member Functions inherited from [Digikam::Option](#)

- [Option](#) (const QString &name, const QString &description)
- [Option](#) (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString [description](#) () const
- QPixmap [icon](#) (Rule::IconType type=Rule::Action) const
- bool [isValid](#) () const
- [ParseResults](#) [parse](#) ([ParseSettings](#) &settings)
- QRegularExpression & [regExp](#) () const
- QPushButton \* [registerButton](#) (QWidget \*parent)
- QAction \* [registerMenu](#) (QMenu \*parent)
- virtual void [reset](#) ()
- [Rule](#) (const QString &name)
- [Rule](#) (const QString &name, const QString &icon)
- TokenList & [tokens](#) () const
- bool [useTokenMenu](#) () const

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString [escapeToken](#) (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void [slotTokenTriggered](#) (const QString &)

### 6.280.1 Member Function Documentation

#### 6.280.1.1 [parseOperation\(\)](#)

```
QString Digikam::DateOption::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [virtual]
```

TODO: describe me

## Parameters

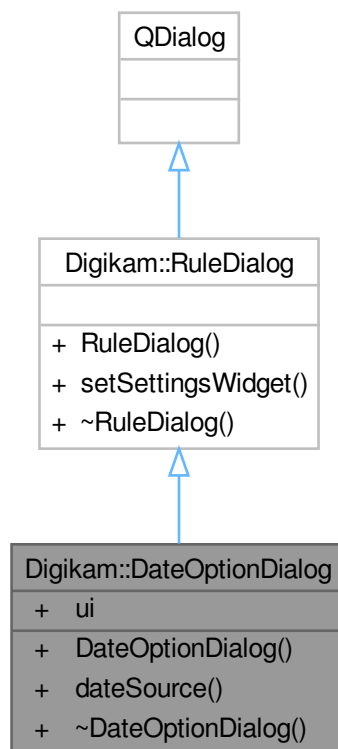
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Option](#).

## 6.281 Digikam::DateOptionDialog Class Reference

Inheritance diagram for Digikam::DateOptionDialog:



## Public Types

- enum `DateSource` { `FromImage = 0` , `CurrentDateTime` , `FixedDateTime` }

### Public Member Functions

- **DateOptionDialog** ([Rule](#) \*parent)
- DataSource **dataSource** () const

### Public Member Functions inherited from [Digikam::RuleDialog](#)

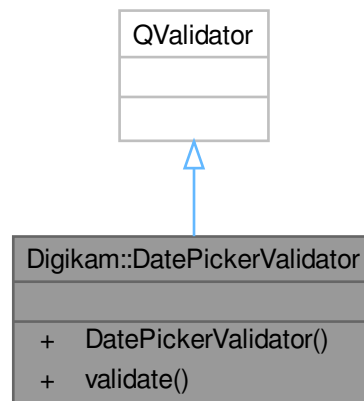
- **RuleDialog** ([Rule](#) \*const parent)
- void **setSettingsWidget** (QWidget \*const settingsWidget)

### Public Attributes

- Ui::DateOptionDialogWidget \*const **ui** = nullptr

## 6.282 Digikam::DatePickerValidator Class Reference

Inheritance diagram for Digikam::DatePickerValidator:

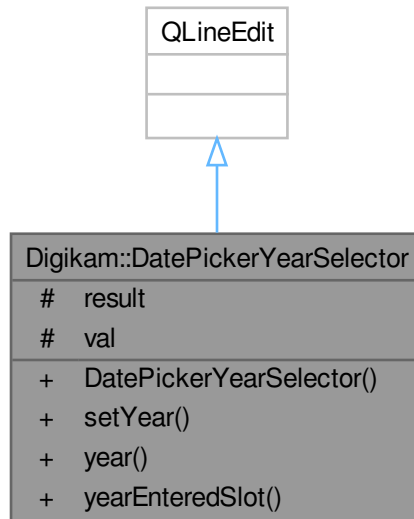


### Public Member Functions

- **DatePickerValidator** ([DDatePicker](#) \*const parent)
- [State](#) **validate** (QString &text, int &) const override

## 6.283 Digikam::DatePickerYearSelector Class Reference

Inheritance diagram for Digikam::DatePickerYearSelector:



### Public Slots

- void **yearEnteredSlot** ()

### Signals

- void **closeMe** (int)

### Public Member Functions

- [DatePickerYearSelector](#) (const QDate &currentDate, QWidget \*const parent=nullptr)
- void **setYear** (int year)
- int **year** () const

### Protected Attributes

- int **result** = 0
- QIntValidator \* **val** = nullptr

## 6.283.1 Constructor & Destructor Documentation

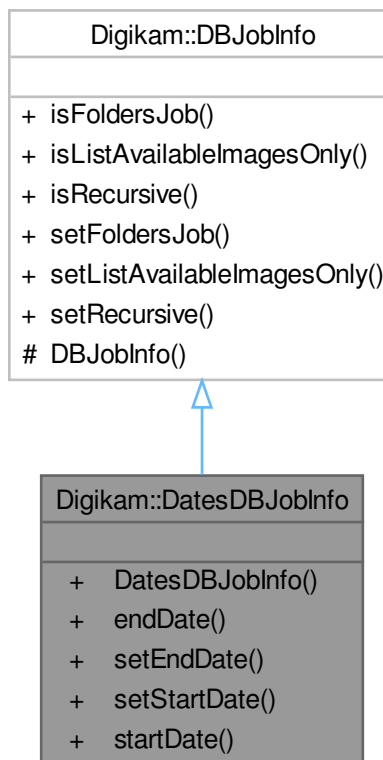
### 6.283.1.1 DatePickerYearSelector()

```
Digikam::DatePickerYearSelector::DatePickerYearSelector (
    const QDate & currentDate,
    QWidget *const parent = nullptr ) [explicit]
```

NOTE: Week numbers are defined by ISO 8601 See [https://en.wikipedia.org/wiki/Week#Week\\_numbering](https://en.wikipedia.org/wiki/Week#Week_numbering) for details

## 6.284 Digikam::DatesDBJobInfo Class Reference

Inheritance diagram for Digikam::DatesDBJobInfo:



### Public Member Functions

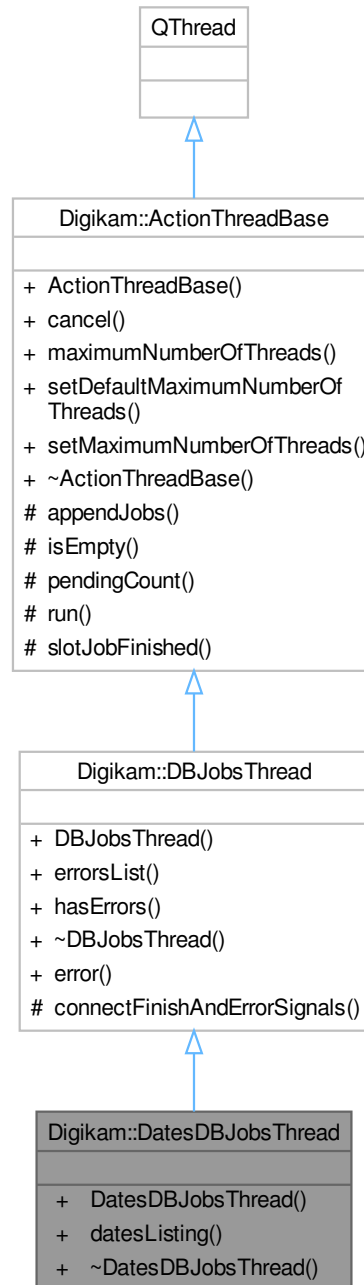
- QDate **endDate** () const
- void **setEndDate** (const QDate &date)
- void **setStartDate** (const QDate &date)
- QDate **startDate** () const

**Public Member Functions inherited from [Digikam::DBJobInfo](#)**

- bool **isFoldersJob** () const
- bool **isListAvailableImagesOnly** () const
- bool **isRecursive** () const
- void **setFoldersJob** ()
- void **setListAvailableImagesOnly** ()
- void **setRecursive** ()

## 6.285 Digikam::DatesDBJobsThread Class Reference

Inheritance diagram for Digikam::DatesDBJobsThread:



### Signals

- void **foldersData** (const QHash< QDateTime, int > &)

## Signals inherited from [Digikam::DBJobsThread](#)

- void **data** (const QList< [ItemLISTERRecord](#) > &records)
- void **finished** ()

## Public Member Functions

- **DatesDBJobsThread** (QObject \*const parent)
- void [datesListing](#) (const [DatesDBJobInfo](#) &info)  
*Starts dates listing and scanning.*

## Public Member Functions inherited from [Digikam::DBJobsThread](#)

- **DBJobsThread** (QObject \*const parent)
- QList< QString > & [errorsList](#) ()  
*A method to get all errors reported from jobs.*
- bool [hasErrors](#) ()  
*hasErrors: a method to check for jobs errors*

## Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void [cancel](#) (bool isCancel=true)
- int [maximumNumberOfThreads](#) () const
- void [setDefaultMaximumNumberOfThreads](#) ()
- void [setMaximumNumberOfThreads](#) (int n)

## Additional Inherited Members

## Public Slots inherited from [Digikam::DBJobsThread](#)

- void [error](#) (const QString &errString)  
*Appends the error string to m\_errorsList.*

## Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void [slotJobFinished](#) ()

## Protected Member Functions inherited from [Digikam::DBJobsThread](#)

- void [connectFinishAndErrorSignals](#) (DBJob \*const j)  
*Connects the signals of job to the signals of the thread.*

## Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void [appendJobs](#) (const [ActionJobCollection](#) &jobs)
- bool [isEmpty](#) () const
- int [pendingCount](#) () const
- void [run](#) () override

## 6.285.1 Member Function Documentation

### 6.285.1.1 [datesListing\(\)](#)

```
void Digikam::DatesDBJobsThread::datesListing (
    const DatesDBJobInfo & info )
```

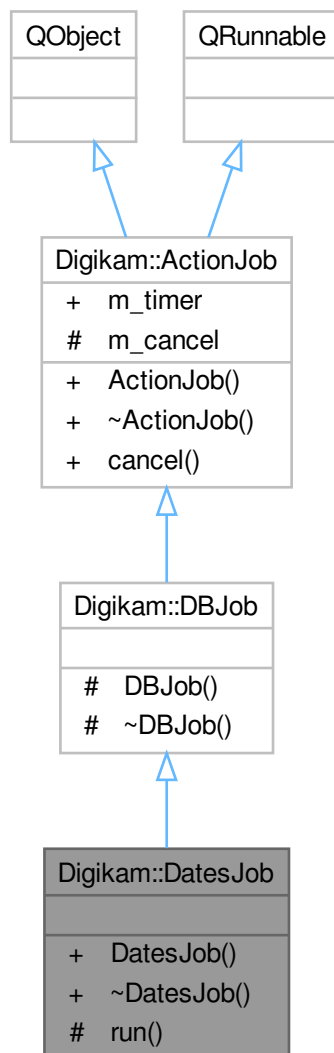


## Parameters

<i>info</i>	represents the dates job info
-------------	-------------------------------

## 6.286 Digikam::DatesJob Class Reference

Inheritance diagram for Digikam::DatesJob:



### Signals

- void **foldersData** (const QMap< QDateTime, int > &datesStatMap)

### Signals inherited from [Digikam::DBJob](#)

- void **data** (const QList< [ItemLISTERRecord](#) > &records)
- void **error** (const QString &err)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Member Functions

- **DatesJob** (const [DatesDBJobInfo](#) &jobInfo)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Protected Member Functions

- void [run](#) () override

### Additional Inherited Members

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Public Attributes inherited from [Digikam::ActionJob](#)

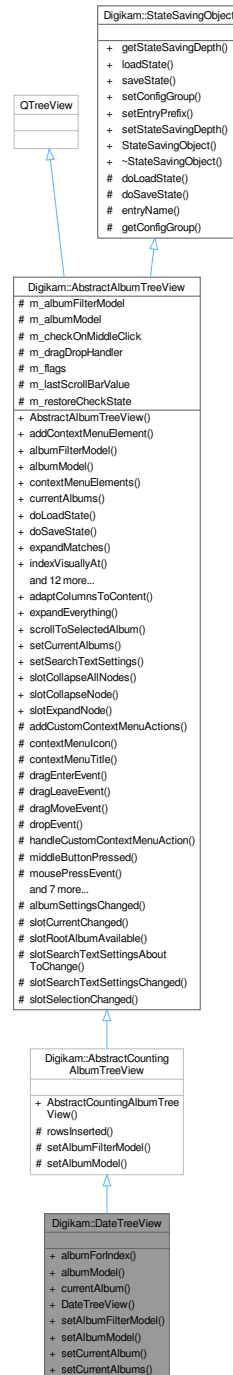
- QElapsedTimer [m\\_timer](#)

### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.287 Digikam::DateTreeView Class Reference

Inheritance diagram for Digikam::DateTreeView:



### Public Slots

- void **setCurrentAlbum** (int dateId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< Album \* > &albums, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()
  - slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()
  - slotExpandNode - expands recursively selected nodes*

## Public Member Functions

- [DAlbum](#) \* [albumForIndex](#) (const QModelIndex &index) const
- [DateAlbumModel](#) \* [albumModel](#) () const
- [DAlbum](#) \* [currentAlbum](#) () const
- [DateTreeView](#) (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([DateAlbumModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void [addContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractSpecificAlbumModel](#) \* [albumModel](#) () const
- QList< [ContextMenuElement](#) \* > [contextMenuElements](#) () const
- template<class A >
  - QList< A \* > [currentAlbums](#) ()
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [expandMatches](#) (const QModelIndex &index)
- QModelIndex [indexVisuallyAt](#) (const QPoint &p)
- void [removeContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > [selectedItems](#) ()
  - selectedItems() -*
- void [setAlbumManagerCurrentAlbum](#) (const bool setCurrentAlbum)
- void [setContextMenuIcon](#) (const QPixmap &pixmap)
- void [setContextMenuTitle](#) (const QString &title)
- void [setEnabledContextMenu](#) (const bool enable)
- void [setExpandNewCurrentItem](#) (const bool doThat)
- void [setExpandOnSingleClick](#) (const bool doThat)
- void [setSelectAlbumOnClick](#) (const bool selectOnClick)
- void [setSelectOnContextMenu](#) (const bool select)
- bool [viewportEvent](#) (QEvent \*event) override

## Public Member Functions inherited from Digikam::StateSavingObject

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

## Public Types inherited from Digikam::AbstractAlbumTreeView

- enum [Flag](#) { [CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) , [AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from Digikam::StateSavingObject

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Signals inherited from Digikam::AbstractAlbumTreeView

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< [Album](#) \* > &selectedAlbums)

## Protected Slots inherited from Digikam::AbstractAlbumTreeView

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Member Functions inherited from Digikam::AbstractCountingAlbumTreeView

- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

## Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album)
- virtual QPixmap [contextMenuIcon](#) () const
- virtual QString [contextMenuTitle](#) () const
- void [dragEnterEvent](#) (QDragEnterEvent \*e) override
- void [dragLeaveEvent](#) (QDragLeaveEvent \*e) override
- void [dragMoveEvent](#) (QDragMoveEvent \*e) override
- void [dropEvent](#) (QDropEvent \*e) override
- virtual void [handleCustomContextMenuAction](#) (QAction \*action, const [AlbumPointer](#)< [Album](#) > &album)
- virtual void [middleButtonPressed](#) ([Album](#) \*a)
- void [mousePressEvent](#) (QMouseEvent \*e) override

*Other helper methods.*

- virtual QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)
- void [rowsAboutToBeRemoved](#) (const QModelIndex &parent, int start, int end) override
- void [rowsInserted](#) (const QModelIndex &index, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
- virtual bool [showContextMenuAt](#) (QContextMenuEvent \*event, [Album](#) \*albumForEvent)
- void [startDrag](#) (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

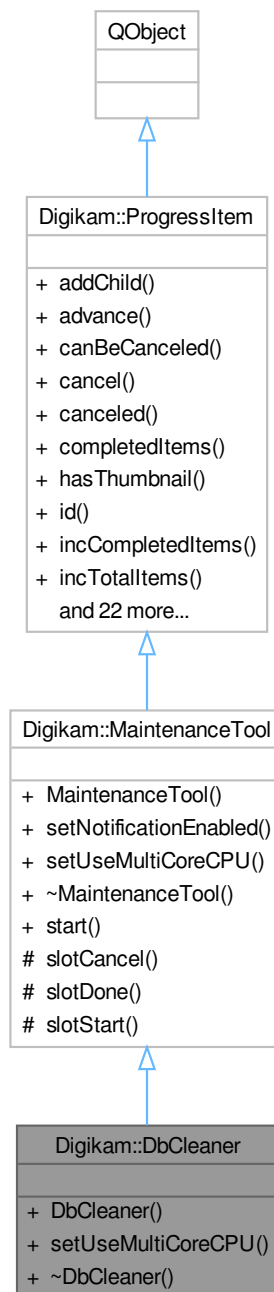
- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* [m\\_albumFilterModel](#) = nullptr
- [AbstractSpecificAlbumModel](#) \* [m\\_albumModel](#) = nullptr
- bool [m\\_checkOnMiddleClick](#) = false
- [AlbumModelDragDropHandler](#) \* [m\\_dragDropHandler](#) = nullptr
- Flags [m\\_flags](#) = DefaultFlags
- int [m\\_lastScrollBarValue](#) = 0
- bool [m\\_restoreCheckState](#) = false

## 6.288 Digikam::DbCleaner Class Reference

Inheritance diagram for Digikam::DbCleaner:



### Public Member Functions

- **DbCleaner** (bool cleanThumbsDb=false, bool cleanFacesDb=false, bool cleanSimilarityDb=false, bool shrinkDatabases=false, [ProgressItem](#) \*const parent=nullptr)
- void [setUseMultiCoreCPU](#) (bool b) override

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- **MaintenanceTool** (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- **ProgressItem** ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool [canBeCanceled](#), bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void [setLabel](#) (const QString &v)
- void [setProgress](#) (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void [setShowAtStart](#) (bool [showAtStart](#))
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void [setStatus](#) (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void [setThumbnail](#) (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void [setUsesBusyIndicator](#) (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool [showAtStart](#) () const
- const QString & [status](#) () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool [usesBusyIndicator](#) () const



## Additional Inherited Members

### Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

### Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

### Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const [QString](#) &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const [QString](#) &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const [QString](#) &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const [QPixmap](#) &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

### Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.288.1 Member Function Documentation

### 6.288.1.1 [setUseMultiCoreCPU\(\)](#)

```
void Digikam::DbCleaner::setUseMultiCoreCPU (
    bool ) [override], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented from [Digikam::MaintenanceTool](#).

## 6.289 Digikam::DbEngineAccess Class Reference

### Static Public Member Functions

- static bool [checkReadyForUse](#) (QString &error)

### 6.289.1 Detailed Description

The [DbEngineAccess](#) class provides access to the database: Create an instance of this class on the stack to retrieve a pointer to the database.

### 6.289.2 Member Function Documentation

#### 6.289.2.1 checkReadyForUse()

```
bool Digikam::DbEngineAccess::checkReadyForUse (
    QString & error ) [static]
```

Checks the availability of drivers. Must be used in children class. Return true if low level drivers are ready to use, else false with an error string of the problem.

## 6.290 Digikam::DbEngineAction Class Reference

### Public Attributes

- QList< [DbEngineActionElement](#) > **dbActionElements**
- QString **mode**
- QString **name**

## 6.291 Digikam::DbEngineActionElement Class Reference

### Public Attributes

- QString **mode**
- int **order** = 0
- QString **statement**

## 6.292 Digikam::DbEngineActionType Class Reference

### Public Member Functions

- **DbEngineActionType** (const [DbEngineActionType](#) &actionType)
- QVariant [getActionValue](#) ()
- bool [isValue](#) () const
- void [setActionValue](#) (const QVariant &actionValue)
- void [setValue](#) (bool [isValue](#))

### Static Public Member Functions

- static [DbEngineActionType](#) **fieldEntry** (const QVariant &actionValue)
- static [DbEngineActionType](#) **value** (const QVariant &value)

#### 6.292.1 Detailed Description

The [DbEngineActionType](#) is used by the [BdEngineBackend](#) to wrap another data object within an sql statement and controls whether it should be used as field entry or as value (prepared to an sql statement with positional binding).

#### 6.292.2 Member Function Documentation

##### 6.292.2.1 `getActionValue()`

```
QVariant Digikam::DbEngineActionType::getActionValue ( )
```

Returns the wrapped object.

##### 6.292.2.2 `isValue()`

```
bool Digikam::DbEngineActionType::isValue ( ) const
```

Returns true, if the entry is an value element. Returns false, if the entry should be used as field entry.

##### 6.292.2.3 `setActionValue()`

```
void Digikam::DbEngineActionType::setActionValue (
    const QVariant & actionValue )
```

Sets the wrapped object.

##### 6.292.2.4 `setValue()`

```
void Digikam::DbEngineActionType::setValue (
    bool isValue )
```

Sets the DBAction mode: true, if the entry is an value element. false, if the entry should be used as field entry.

## 6.293 Digikam::DbEngineConfig Class Reference

### Static Public Member Functions

- static bool **checkReadyForUse** ()
- static [DbEngineConfigSettings](#) **element** (const QString &databaseType)
- static QString **errorMessage** ()

## 6.294 Digikam::DbEngineConfigSettings Class Reference

### Public Attributes

- QString **connectOptions**
- QString **databaseID**
- QString **databaseName**
- QString **hostName**
- QString **password**
- QString **port**
- QMap< QString, [DbEngineAction](#) > **sqlStatements**
- QString **userName**

## 6.295 Digikam::DbEngineConfigSettingsLoader Class Reference

### Public Member Functions

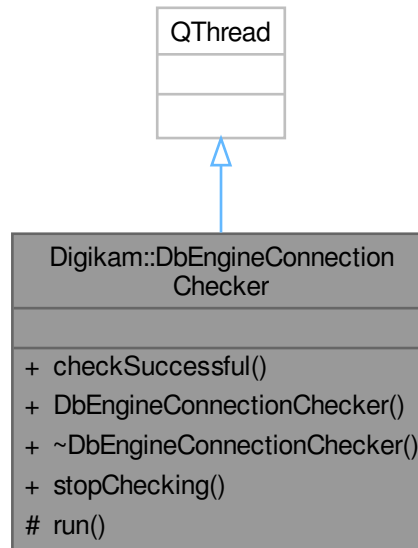
- **DbEngineConfigSettingsLoader** (const QString &filepath, int xmlVersion)
- bool **readConfig** (const QString &filepath, int xmlVersion)
- [DbEngineConfigSettings](#) **readDatabase** (const QDomElement &databaseElement)
- void **readDBActions** (const QDomElement &sqlStatementElements, [DbEngineConfigSettings](#) &config←  
Element)

### Public Attributes

- QMap< QString, [DbEngineConfigSettings](#) > **databaseConfigs**
- QString **errorMessage**
- bool **isValid** = false

## 6.296 Digikam::DbEngineConnectionChecker Class Reference

Inheritance diagram for Digikam::DbEngineConnectionChecker:



### Public Slots

- void **stopChecking** ()

### Signals

- void **done** ()
- void **failedAttempt** ()

### Public Member Functions

- bool **checkSuccessful** () const
- **DbEngineConnectionChecker** (const [DbEngineParameters](#) &parameters)

### Protected Member Functions

- void **run** () override

## 6.297 Digikam::DbEngineErrorAnswer Class Reference

Inheritance diagram for Digikam::DbEngineErrorAnswer:



### Public Member Functions

- virtual void `connectionErrorAbortQueries ()=0`
- virtual void `connectionErrorContinueQueries ()=0`

## 6.297.1 Member Function Documentation

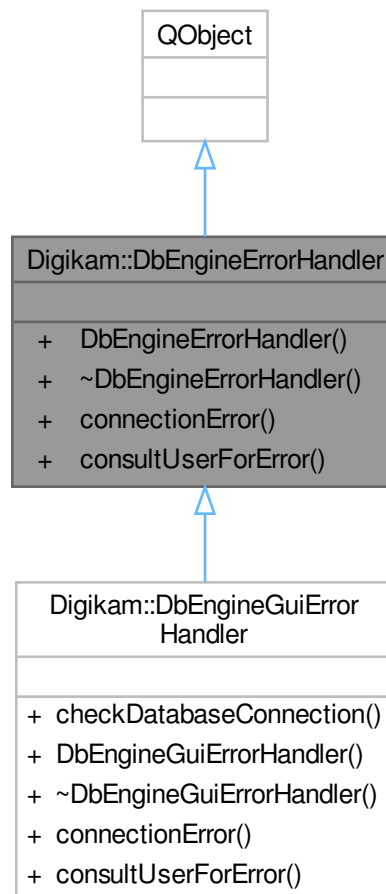
### 6.297.1.1 connectionErrorContinueQueries()

```
virtual void Digikam::DbEngineErrorAnswer::connectionErrorContinueQueries ( ) [pure virtual]
```

Implemented in [Digikam::BdEngineBackendPrivate](#).

## 6.298 Digikam::DbEngineErrorHandler Class Reference

Inheritance diagram for Digikam::DbEngineErrorHandler:



### Public Slots

- virtual void [connectionError](#) ([DbEngineErrorAnswer](#) \*answer, const QSqlError &error, const QString &query)=0
- virtual void [consultUserForError](#) ([DbEngineErrorAnswer](#) \*answer, const QSqlError &error, const QString &query)=0

## 6.298.1 Member Function Documentation

### 6.298.1.1 connectionError

```
virtual void Digikam::DbEngineErrorHandler::connectionError (
    DbEngineErrorAnswer * answer,
    const QSqlError & error,
    const QString & query ) [pure virtual], [slot]
```

In the situation of a connection error, all threads will be waiting with their queries and this method is called. This method can display an error dialog and try to repair the connection. It must then call either `connectionErrorContinueQueries()` or `connectionErrorAbortQueries()`. The method is guaranteed to be invoked in the UI thread.

### 6.298.1.2 consultUserForError

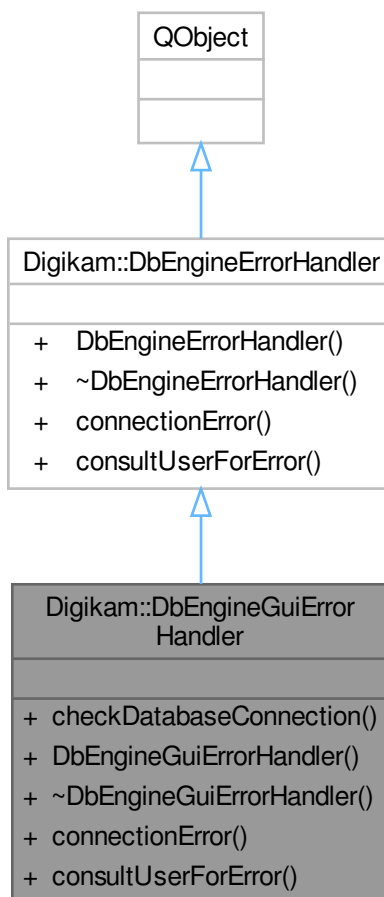
```
virtual void Digikam::DbEngineErrorHandler::consultUserForError (
    DbEngineErrorAnswer * answer,
    const QSqlError & error,
    const QString & query ) [pure virtual], [slot]
```

In the situation of an error requiring user intervention or information, all threads will be waiting with their queries and this method is called. This method can display an error dialog. It must then call either `connectionErrorContinueQueries()` or `connectionErrorAbortQueries()`. The method is guaranteed to be invoked in the UI thread.



## 6.299 Digikam::DbEngineGuiErrorHandler Class Reference

Inheritance diagram for Digikam::DbEngineGuiErrorHandler:



### Public Slots

- void **connectionError** ([DbEngineErrorAnswer](#) \*answer, const QSqlError &error, const QString &query) override
- void **consultUserForError** ([DbEngineErrorAnswer](#) \*answer, const QSqlError &error, const QString &query) override

### Public Slots inherited from [Digikam::DbEngineErrorHandler](#)

- virtual void [connectionError](#) ([DbEngineErrorAnswer](#) \*answer, const QSqlError &error, const QString &query)=0
- virtual void [consultUserForError](#) ([DbEngineErrorAnswer](#) \*answer, const QSqlError &error, const QString &query)=0

## Public Member Functions

- bool **checkDatabaseConnection** ()
- **DbEngineGuiErrorHandler** (const [DbEngineParameters](#) &parameters)

## 6.300 Digikam::DbEngineLocking Class Reference

### Public Attributes

- int **lockCount** = 0  
*create a recursive mutex*
- QRecursiveMutex **mutex**

## 6.301 Digikam::DbEngineParameters Class Reference

### Public Member Functions

- **DbEngineParameters** (const QString &\_type, const QString &\_databaseNameCore, const QString &\_connectOptions=QString(), const QString &\_hostName=QString(), int \_port=-1, bool \_walMode=false, bool \_internalServer=false, const QString &\_userName=QString(), const QString &\_password=QString(), const QString &\_databaseNameThumbnails=QString(), const QString &\_databaseNameFace=QString(), const QString &\_databaseNameSimilarity=QString(), const QString &\_internalServerDBPath=QString(), const QString &\_internalServerMysqlInitCmd=QString(), const QString &\_internalServerMysqlAdminCmd=QString(), const QString &\_internalServerMysqlServerCmd=QString(), const QString &\_internalServerMysqlUpgradeCmd=QString())
- [DbEngineParameters](#) (const QUrl &url)
- [DbEngineParameters](#) **faceParameters** () const
- QString **getCoreDatabaseNameOrDir** () const
- QString **getFaceDatabaseNameOrDir** () const
- QString **getSimilarityDatabaseNameOrDir** () const
- QString **getThumbsDatabaseNameOrDir** () const
- QByteArray **hash** () const
- void **insertInUrl** (QUrl &url) const
- QString **internalServerPath** () const
- bool **isMySQL** () const
- bool **isSQLite** () const
- bool **isValid** () const
- void **legacyAndDefaultChecks** (const QString &suggestedPath=QString())
- bool **operator!=** (const [DbEngineParameters](#) &other) const
- bool **operator==** (const [DbEngineParameters](#) &other) const
- void **readFromConfig** (const QString &configGroup=QString())
- void **removeLegacyConfig** ()
- void **setCoreDatabasePath** (const QString &folderOrFileOrName)
- void **setFaceDatabasePath** (const QString &folderOrFileOrName)
- void **setInternalServerPath** (const QString &path)
- void **setSimilarityDatabasePath** (const QString &folderOrFileOrName)
- void **setThumbsDatabasePath** (const QString &folderOrFileOrName)
- [DbEngineParameters](#) **similarityParameters** () const
- QString **SQLiteDatabaseFile** () const
- [DbEngineParameters](#) **thumbnailParameters** () const
- void **writeToConfig** (const QString &configGroup=QString()) const

### Static Public Member Functions

- static QString **coreDatabaseDirectorySQLite** (const QString &path)
- static QString **coreDatabaseFileSQLite** (const QString &folderOrFile)
- static QString **defaultMysqlAdminCmd** ()
- static QString **defaultMysqlInitCmd** ()
- static QString **defaultMysqlServerCmd** ()
- static QString **defaultMysqlUpgradeCmd** ()
- static DbEngineParameters **defaultParameters** (const QString &databaseType)
- static QString **faceDatabaseDirectorySQLite** (const QString &path)
- static QString **faceDatabaseFileSQLite** (const QString &folderOrFile)
- static QString **MySQLDatabaseType** ()
- static DbEngineParameters **parametersForSQLite** (const QString &databaseFile)
- static DbEngineParameters **parametersForSQLiteDefaultFile** (const QString &directory)
- static DbEngineParameters **parametersFromConfig** (const QString &configGroup=QString())
- static void **removeFromUrl** (QUrl &url)
- static QString **serverPrivatePath** ()
- static QString **similarityDatabaseDirectorySQLite** (const QString &path)
- static QString **similarityDatabaseFileSQLite** (const QString &folderOrFile)
- static QString **SQLiteDatabaseType** ()
- static QString **thumbnailDatabaseDirectorySQLite** (const QString &path)
- static QString **thumbnailDatabaseFileSQLite** (const QString &folderOrFile)

### Public Attributes

- QString **connectOptions**
- QString **databaseNameCore**
- QString **databaseNameFace**
- QString **databaseNameSimilarity**
- QString **databaseNameThumbnails**
- QString **databaseType**
- QString **hostName**
- bool **internalServer** = false
- QString **internalServerDBPath**
- QString **internalServerMysqlAdminCmd**
- QString **internalServerMysqlInitCmd**
- QString **internalServerMysqlServerCmd**
- QString **internalServerMysqlUpgradeCmd**
- QString **password**
- int **port** = -1
- QString **userName**
- bool **walMode** = false

### 6.301.1 Detailed Description

This class encapsulates all parameters needed to establish a connection to a database (inspired by the API of Qt::Sql). The values can be read from and written to a QUrl.

## 6.301.2 Constructor & Destructor Documentation

### 6.301.2.1 DbEngineParameters()

```
Digikam::DbEngineParameters::DbEngineParameters (
    const QUrl & url ) [explicit]
```

QUrl helpers.

## 6.301.3 Member Function Documentation

### 6.301.3.1 defaultMysqlAdminCmd()

```
QString Digikam::DbEngineParameters::defaultMysqlAdminCmd ( ) [static]
```

Return the default Mysql server administration name (Internal server only).

### 6.301.3.2 defaultMysqlInitCmd()

```
QString Digikam::DbEngineParameters::defaultMysqlInitCmd ( ) [static]
```

Return the default Mysql initialization command name (Internal server only).

### 6.301.3.3 defaultMysqlServerCmd()

```
QString Digikam::DbEngineParameters::defaultMysqlServerCmd ( ) [static]
```

Return the default Mysql server command name (Internal server only).

### 6.301.3.4 defaultMysqlUpgradeCmd()

```
QString Digikam::DbEngineParameters::defaultMysqlUpgradeCmd ( ) [static]
```

Return the default Mysql upgrade command name (Internal server only).

### 6.301.3.5 defaultParameters()

```
DbEngineParameters Digikam::DbEngineParameters::defaultParameters (
    const QString & databaseType ) [static]
```

Return a set of default parameters for the given type. For Mysql, it return internal server configuration.

### 6.301.3.6 faceParameters()

```
DbEngineParameters Digikam::DbEngineParameters::faceParameters ( ) const
```

Replaces databaseName with databaseNameFace.

### 6.301.3.7 getCoreDatabaseNameOrDir()

```
QString Digikam::DbEngineParameters::getCoreDatabaseNameOrDir ( ) const
```

#### Note

In case of SQLite, the database name typically is a file. For non-SQLite, this simply handle the database name.

### 6.301.3.8 hash()

```
QByteArray Digikam::DbEngineParameters::hash ( ) const
```

Creates a unique hash of the values stored in this object.

### 6.301.3.9 isValid()

```
bool Digikam::DbEngineParameters::isValid ( ) const
```

Performs basic checks that the parameters are not empty and have the information required for the databaseType.

### 6.301.3.10 parametersForSQLite()

```
DbEngineParameters Digikam::DbEngineParameters::parametersForSQLite (
    const QString & databaseFile ) [static]
```

Convenience methods to create a [DbEngineParameters](#) object for an SQLITE database specified by the local file path.

### 6.301.3.11 readFromConfig()

```
void Digikam::DbEngineParameters::readFromConfig (
    const QString & configGroup = QString() )
```

Read and write parameters from config. You can specify the group, or use the default value.

### 6.301.3.12 serverPrivatePath()

```
QString Digikam::DbEngineParameters::serverPrivatePath ( ) [static]
```

Return the hidden path from home directory to store private data used by internal Mysql server.

### 6.301.3.13 setCoreDatabasePath()

```
void Digikam::DbEngineParameters::setCoreDatabasePath (
    const QString & folderOrFileOrName )
```

Use these methods if you set a file or a folder.

### 6.301.3.14 setInternalServerPath()

```
void Digikam::DbEngineParameters::setInternalServerPath (
    const QString & path )
```

For Mysql internal server: manage the database path to store database files.

### 6.301.3.15 similarityParameters()

```
DbEngineParameters Digikam::DbEngineParameters::similarityParameters ( ) const
```

Replaces databaseName with databaseNameFace.

### 6.301.3.16 SQLiteDatabaseType()

```
QString Digikam::DbEngineParameters::SQLiteDatabaseType ( ) [static]
```

Returns the databaseType designating the said database. If you have a [DbEngineParameters](#) object already, you can use `isSQLite()` as well. These strings are identical to the driver identifiers in the Qt SQL module.

### 6.301.3.17 thumbnailParameters()

```
DbEngineParameters Digikam::DbEngineParameters::thumbnailParameters ( ) const
```

Replaces databaseName with databaseNameThumbnails.

## 6.301.4 Member Data Documentation

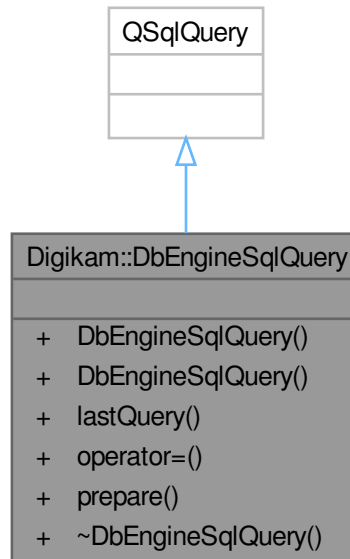
### 6.301.4.1 internalServerMysqlInitCmd

```
QString Digikam::DbEngineParameters::internalServerMysqlInitCmd
```

Settings stored in config file and used only with internal server at runtime to start server instance or init database tables.

## 6.302 Digikam::DbEngineSqlQuery Class Reference

Inheritance diagram for Digikam::DbEngineSqlQuery:



### Public Member Functions

- **DbEngineSqlQuery** (const QSqlDatabase &db)
- **DbEngineSqlQuery** (const QSqlQuery &other)
- QString **lastQuery** () const
- **DbEngineSqlQuery** & **operator=** (const **DbEngineSqlQuery** &other)
- bool **prepare** (const QString &query)

## 6.303 Digikam::DbEngineThreadData Class Reference

### Public Member Functions

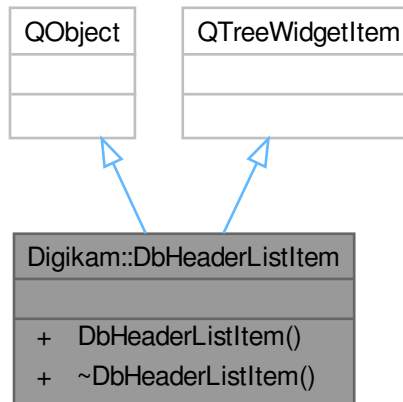
- void **closeDatabase** ()

### Public Attributes

- QString **connectionName**
- QSqlError **lastError**
- int **transactionCount** = 0
- int **valid** = 0

## 6.304 Digikam::DbHeaderListItem Class Reference

Inheritance diagram for Digikam::DbHeaderListItem:



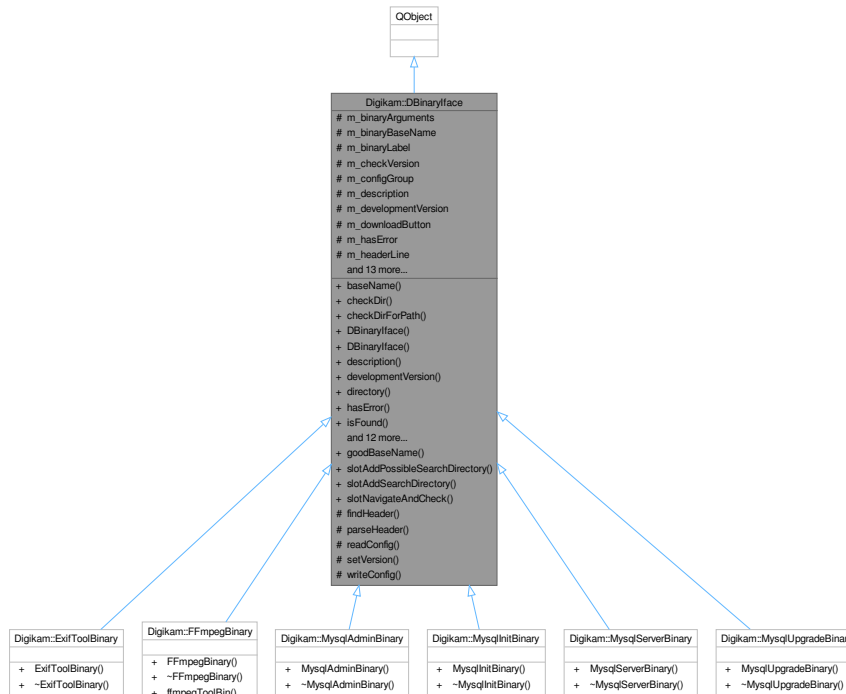
### Public Member Functions

- **DbHeaderListItem** (QTreeWidgetItem \*parent, const QString &key)



## 6.305 Digikam::DBinaryface Class Reference

Inheritance diagram for Digikam::DBinaryface:



### Public Slots

- virtual void **slotAddPossibleSearchDirectory** (const QString &dir)
- virtual void **slotAddSearchDirectory** (const QString &dir)
- virtual void **slotNavigateAndCheck** ()

### Signals

- void **signalBinaryValid** ()
- void **signalSearchDirectoryAdded** (const QString &dir)

### Public Member Functions

- virtual QString **baseName** () const
- virtual bool **checkDir** ()
- virtual bool **checkDirForPath** (const QString &path)
- **DBinaryface** (const QString &binaryName, const QString &minimalVersion, const QString &header, const int headerLine, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- **DBinaryface** (const QString &binaryName, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- const QString & **description** () const
- bool **developmentVersion** () const

- virtual QString **directory** () const
- bool **hasError** () const
- bool **isFound** () const
- bool **isValid** () const
- virtual QString **minimalVersion** () const
- virtual QString **path** () const
- virtual QString **path** (const QString &dir) const
- virtual QString **projectName** () const
- virtual bool **recheckDirectories** ()
- virtual void **setup** (const QString &prev=QString())
- virtual QUrl **url** () const
- const QString & **version** () const
- bool **versionIsRight** () const
- bool **versionIsRight** (const float) const

### Static Public Member Functions

- static QString **goodBaseName** (const QString &b)

### Protected Member Functions

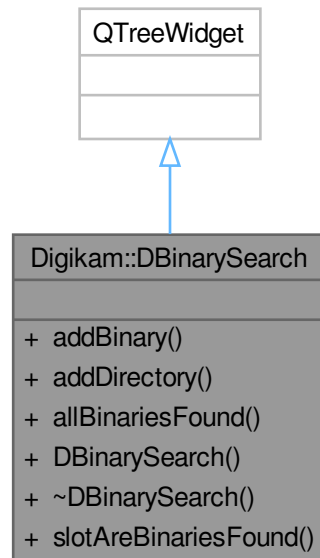
- QString **findHeader** (const QStringList &output, const QString &header) const
- virtual bool **parseHeader** (const QString &output)
- virtual QString **readConfig** ()
- void **setVersion** (QString &version)
- virtual void **writeConfig** ()

### Protected Attributes

- const QStringList **m\_binaryArguments**
- const QString **m\_binaryBaseName**
- QLabel \* **m\_binaryLabel** = nullptr
- const bool **m\_checkVersion**
- const QString **m\_configGroup**
- QString **m\_description**
- bool **m\_developmentVersion** = false
- QLabel \* **m\_downloadButton** = nullptr
- bool **m\_hasError** = false
- const int **m\_headerLine**
- const QString **m\_headerStarts**
- bool **m\_isFound** = false
- QLineEdit \* **m\_lineEdit** = nullptr
- const QString **m\_minimalVersion**
- QPushButton \* **m\_pathButton** = nullptr
- QString **m\_pathDir** = QLatin1String("")
- QFrame \* **m\_pathWidget** = nullptr
- const QString **m\_projectName**
- QSet< QString > **m\_searchPaths**
- QLabel \* **m\_statusIcon** = nullptr
- const QUrl **m\_url**
- QString **m\_version** = QLatin1String("")
- QLabel \* **m\_versionLabel** = nullptr

## 6.306 Digikam::DBinarySearch Class Reference

Inheritance diagram for Digikam::DBinarySearch:



### Public Types

- enum `ColumnType` {  
    **Status** = 0 , **Binary** , **Version** , **Button** ,  
    **Link** }

### Public Slots

- void `slotAreBinariesFound ()`

### Signals

- void `signalAddDirectory` (const QString &dir)
- void `signalAddPossibleDirectory` (const QString &dir)
- void `signalBinariesFound` (bool)

### Public Member Functions

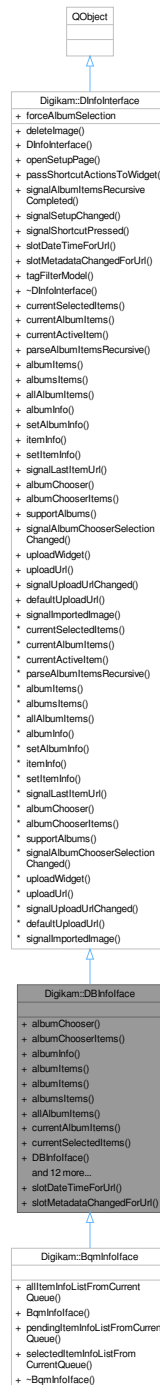
- void `addBinary` ([DBinaryIface](#) &binary)
- void `addDirectory` (const QString &dir)
- bool `allBinariesFound ()`
- `DBinarySearch` (QWidget \*const parent)

### 6.306.1 Detailed Description

This class has nothing to do with a binary search, it is a widget to search for binaries.

## 6.307 Digikam::DBInfoface Class Reference

Inheritance diagram for Digikam::DBInfoface:



**Public Slots**

- void **slotDateTimeForUrl** (const QUrl &url, const QDateTime &dt, bool updModDate) override
- void **slotMetadataChangedForUrl** (const QUrl &url) override

**Public Member Functions**

- QWidget \* **albumChooser** (QWidget \*const parent) const override  
*Albums chooser view methods (to use items from albums before to process).*
- **DAlbumIDs albumChooserItems** () const override
- **DInfoMap albumInfo** (int) const override
- QList< QUrl > **albumItems** (**Album** \*const album) const
- QList< QUrl > **albumItems** (int id) const override
- QList< QUrl > **albumsItems** (const **DAlbumIDs** &) const override
- QList< QUrl > **allAlbumItems** () const override
- QList< QUrl > **currentAlbumItems** () const override
- QList< QUrl > **currentSelectedItems** () const override  
*Low level items and albums methods.*
- **DBInterface** (QObject \*const parent, const QList< QUrl > &lst=QList< QUrl >(), const **OperationType** type=**UnspecifiedOps**)
- QUrl **defaultUploadUrl** () const override  
*Url to upload new items without to use album selector.*
- void **deleteImage** (const QUrl &url) override  
*Manipulate with item.*
- **DInfoMap itemInfo** (const QUrl &) const override
- void **openSetupPage** (SetupPage page) override  
*Open configuration dialog page.*
- void **parseAlbumItemsRecursive** () override
- QMap< QString, QString > **passShortcutActionsToWidget** (QWidget \*const wdg) const override  
*Pass extra shortcut actions to widget and return prefixes of shortcuts.*
- void **setItemInfo** (const QUrl &, const **DInfoMap** &) override
- bool **supportAlbums** () const override
- QAbstractItemModel \* **tagFilterModel** () override  
*Return an instance of tag filter model if host application support this feature, else null pointer.*
- QUrl **uploadUrl** () const override
- QWidget \* **uploadWidget** (QWidget \*const parent) const override  
*Album selector view methods (to upload items from an external place).*

**Public Member Functions inherited from Digikam::DInfoInterface**

- **DInfoInterface** (QObject \*const parent)
- Q\_SIGNAL void **signalAlbumItemsRecursiveCompleted** (const QList< QUrl > &imageList)
- Q\_SIGNAL void **signalSetupChanged** ()
- Q\_SIGNAL void **signalShortcutPressed** (const QString &shortcut, int val)
- virtual Q\_SLOT void **slotDateTimeForUrl** (const QUrl &url, const QDateTime &dt, bool updModDate)  
*Slot to call when date time stamp from item is changed.*
- virtual Q\_SLOT void **slotMetadataChangedForUrl** (const QUrl &url)  
*Slot to call when something in metadata from item is changed.*
- virtual QUrl **currentActiveItem** () const
- virtual void **setAlbumInfo** (int, const **DInfoMap** &) const
- Q\_SIGNAL void **signalLastItemUrl** (const QUrl &)
- Q\_SIGNAL void **signalAlbumChooserSelectionChanged** ()
- Q\_SIGNAL void **signalUploadUrlChanged** ()
- Q\_SIGNAL void **signalImportedImage** (const QUrl &)

## Additional Inherited Members

### Public Types inherited from [Digikam::DInfoInterface](#)

- typedef `QList< int >` **DAlbumIDs**  
*List of [Album](#) ids.*
- typedef `QMap< QString, QVariant >` **DInfoMap**  
*Map of properties name and value.*
- enum **SetupPage** { `ExifToolPage = 0` , `ImageQualityPage` }

### Public Attributes inherited from [Digikam::DInfoInterface](#)

- bool **forceAlbumSelection** = false

## 6.307.1 Member Function Documentation

### 6.307.1.1 albumChooser()

```
QWidget * Digikam::DBInfoIface::albumChooser (
    QWidget *const parent ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.2 albumChooserItems()

```
DBInfoIface::DAlbumIDs Digikam::DBInfoIface::albumChooserItems ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.3 albumInfo()

```
DBInfoIface::DInfoMap Digikam::DBInfoIface::albumInfo (
    int gid ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.4 albumItems()

```
QList< QUrl > Digikam::DBInfoIface::albumItems (
    int id ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.5 albumsItems()

```
QList< QUrl > Digikam::DBInfoInterface::albumsItems (
    const DAlbumIDs & lst ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.6 allAlbumItems()

```
QList< QUrl > Digikam::DBInfoInterface::allAlbumItems ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.7 currentAlbumItems()

```
QList< QUrl > Digikam::DBInfoInterface::currentAlbumItems ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.8 currentSelectedItems()

```
QList< QUrl > Digikam::DBInfoInterface::currentSelectedItems ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.9 defaultUploadUrl()

```
QUrl Digikam::DBInfoInterface::defaultUploadUrl ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.10 deleteImage()

```
void Digikam::DBInfoInterface::deleteImage (
    const QUrl & url ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.307.1.11 itemInfo()

```
DInfoInterface::DInfoMap Digikam::DBInfoInterface::itemInfo (
    const QUrl & url ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

#### 6.307.1.12 openSetupPage()

```
void Digikam::DBInfoIface::openSetupPage (
    SetupPage page ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

#### 6.307.1.13 parseAlbumItemsRecursive()

```
void Digikam::DBInfoIface::parseAlbumItemsRecursive ( ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

#### 6.307.1.14 passShortcutActionsToWidget()

```
QMap< QString, QString > Digikam::DBInfoIface::passShortcutActionsToWidget (
    QWidget *const ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

#### 6.307.1.15 setItemInfo()

```
void Digikam::DBInfoIface::setItemInfo (
    const QUrl & url,
    const DInfoMap & map ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

#### 6.307.1.16 supportAlbums()

```
bool Digikam::DBInfoIface::supportAlbums ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

#### 6.307.1.17 tagFilterModel()

```
QAbstractItemModel * Digikam::DBInfoIface::tagFilterModel ( ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

#### 6.307.1.18 uploadUrl()

```
QUrl Digikam::DBInfoIface::uploadUrl ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).



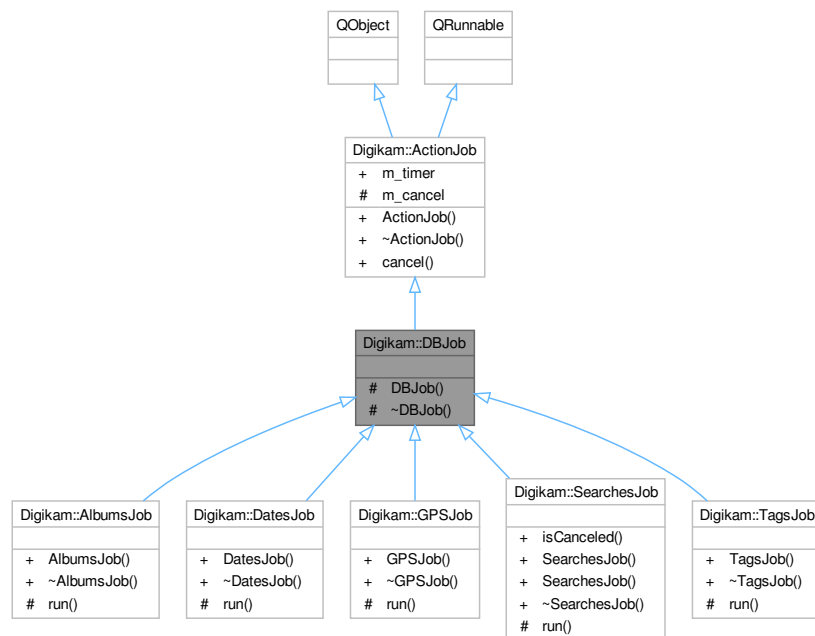
## 6.307.1.19 uploadWidget()

```
QWidget * Digikam::DBInfoIface::uploadWidget (
    QWidget *const parent ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

## 6.308 Digikam::DBJob Class Reference

Inheritance diagram for Digikam::DBJob:



## Signals

- void **data** (const QList< [ItemLISTERRecord](#) > &records)
- void **error** (const QString &err)

Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Additional Inherited Members

Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Member Functions inherited from Digikam::ActionJob

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Public Attributes inherited from Digikam::ActionJob

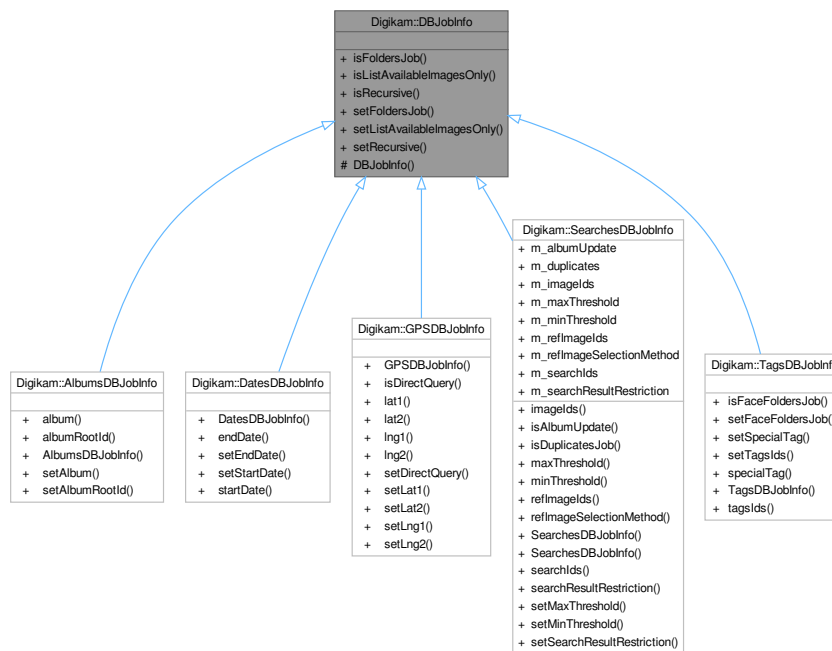
- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from Digikam::ActionJob

- bool [m\\_cancel](#) = false

## 6.309 Digikam::DBJobInfo Class Reference

Inheritance diagram for Digikam::DBJobInfo:

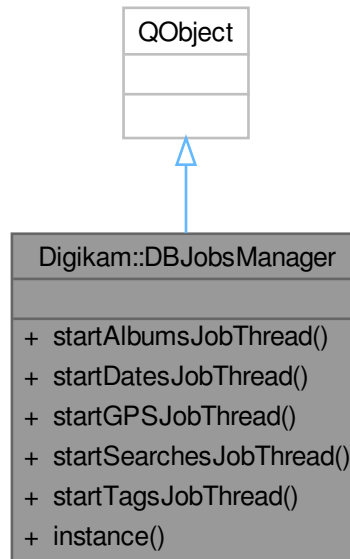


## Public Member Functions

- bool **isFoldersJob** () const
- bool **isListAvailableImagesOnly** () const
- bool **isRecursive** () const
- void **setFoldersJob** ()
- void **setListAvailableImagesOnly** ()
- void **setRecursive** ()

## 6.310 Digikam::DBJobsManager Class Reference

Inheritance diagram for Digikam::DBJobsManager:



### Public Member Functions

- [AlbumsDBJobsThread](#) \* [startAlbumsJobThread](#) (const [AlbumsDBJobInfo](#) &jInfo)  
*startAlbumsJobThread: creates and starts Albums Job Thread*
- [DatesDBJobsThread](#) \* [startDatesJobThread](#) (const [DatesDBJobInfo](#) &jInfo)  
*startDatesJobThread: creates and starts Dates Job Thread*
- [GPSDBJobsThread](#) \* [startGPSJobThread](#) (const [GPSDBJobInfo](#) &jInfo)  
*startGPSJobThread: creates and starts GPS Job Thread*
- [SearchesDBJobsThread](#) \* [startSearchesJobThread](#) (const [SearchesDBJobInfo](#) &jInfo)  
*startSearchesJobThread: creates and starts Searches Job Thread*
- [TagsDBJobsThread](#) \* [startTagsJobThread](#) (const [TagsDBJobInfo](#) &jInfo)  
*startTagsJobThread: creates and starts Tag Job Thread*

### Static Public Member Functions

- static [DBJobsManager](#) \* [instance](#) ()  
*instance: returns DBJobsManager singleton*

### Friends

- class [DBJobsManagerCreator](#)

## 6.310.1 Member Function Documentation

### 6.310.1.1 instance()

```
DBJobsManager * Digikam::DBJobsManager::instance ( ) [static]
```

#### Returns

[DBJobsManager](#) global instance

### 6.310.1.2 startAlbumsJobThread()

```
AlbumsDBJobsThread * Digikam::DBJobsManager::startAlbumsJobThread (
    const AlbumsDBJobInfo & jInfo )
```

#### Parameters

<i>jInfo</i>	holds job info about the DB job
--------------	---------------------------------

#### Returns

[AlbumsDBJobsThread](#) instance for signal/slot connection

### 6.310.1.3 startDatesJobThread()

```
DatesDBJobsThread * Digikam::DBJobsManager::startDatesJobThread (
    const DatesDBJobInfo & jInfo )
```

#### Parameters

<i>jInfo</i>	holds job info about the DB job
--------------	---------------------------------

#### Returns

[DatesDBJobsThread](#) instance for signal/slot connection

### 6.310.1.4 startGPSJobThread()

```
GPSDBJobsThread * Digikam::DBJobsManager::startGPSJobThread (
    const GPSDBJobInfo & jInfo )
```

#### Parameters

<i>jInfo</i>	holds job info about the DB job
--------------	---------------------------------

**Returns**

[GPSDBJobsThread](#) instance for signal/slot connection

**6.310.1.5 startSearchesJobThread()**

```
SearchesDBJobsThread * Digikam::DBJobsManager::startSearchesJobThread (
    const SearchesDBJobInfo & jInfo )
```

**Parameters**

<i>jInfo</i>	holds job info about the DB job
--------------	---------------------------------

**Returns**

[SearchesDBJobsThread](#) instance for signal/slot connection

**6.310.1.6 startTagsJobThread()**

```
TagsDBJobsThread * Digikam::DBJobsManager::startTagsJobThread (
    const TagsDBJobInfo & jInfo )
```

**Parameters**

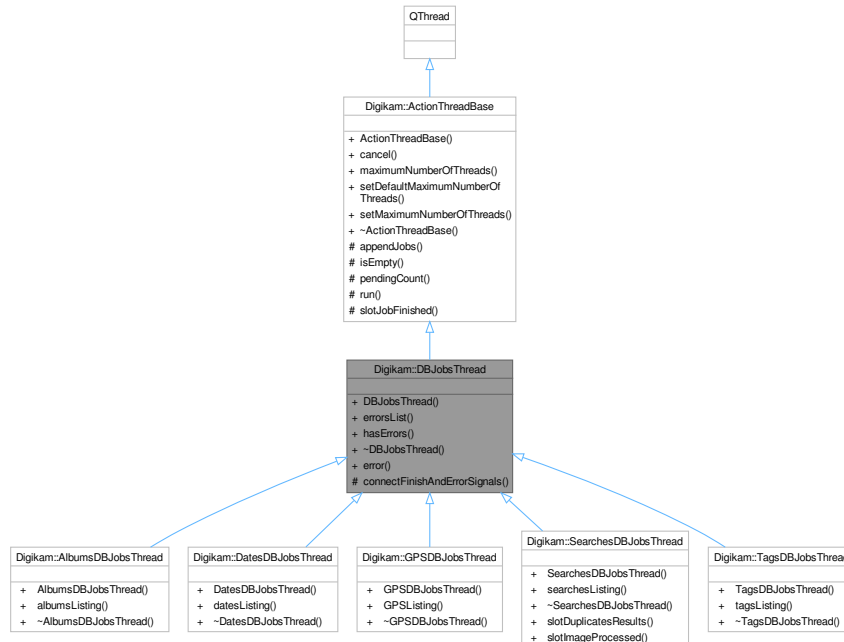
<i>jInfo</i>	holds job info about the DB job
--------------	---------------------------------

**Returns**

[TagsDBJobsThread](#) instance for signal/slot connection

## 6.311 Digikam::DBJobsThread Class Reference

Inheritance diagram for Digikam::DBJobsThread:



### Public Slots

- void **error** (const QString &errString)  
*Appends the error string to m\_errorsList.*

### Signals

- void **data** (const QList< [ItemListerRecord](#) > &records)
- void **finished** ()

### Public Member Functions

- **DBJobsThread** (QObject \*const parent)
- QList< QString > & **errorsList** ()  
*A method to get all errors reported from jobs.*
- bool **hasErrors** ()  
*hasErrors: a method to check for jobs errors*

### Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

## Protected Member Functions

- void [connectFinishAndErrorSignals](#) (DBJob \*const j)  
*Connects the signals of job to the signals of the thread.*

## Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void [appendJobs](#) (const [ActionJobCollection](#) &jobs)
- bool [isEmpty](#) () const
- int [pendingCount](#) () const
- void [run](#) () override

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void [slotJobFinished](#) ()

## 6.311.1 Member Function Documentation

### 6.311.1.1 [connectFinishAndErrorSignals\(\)](#)

```
void Digikam::DBJobsThread::connectFinishAndErrorSignals (
    DBJob *const j ) [protected]
```

#### Parameters

<i>j</i>	Job that wanted to be connected
----------	---------------------------------

### 6.311.1.2 [error](#)

```
void Digikam::DBJobsThread::error (
    const QString & errString ) [slot]
```

#### Parameters

<i>errString</i>	error string reported from the job
------------------	------------------------------------

### 6.311.1.3 [errorsList\(\)](#)

```
QList< QString > & Digikam::DBJobsThread::errorsList ( )
```

#### Returns

String list with errors

### 6.311.1.4 hasErrors()

```
bool Digikam::DBJobsThread::hasErrors ( )
```

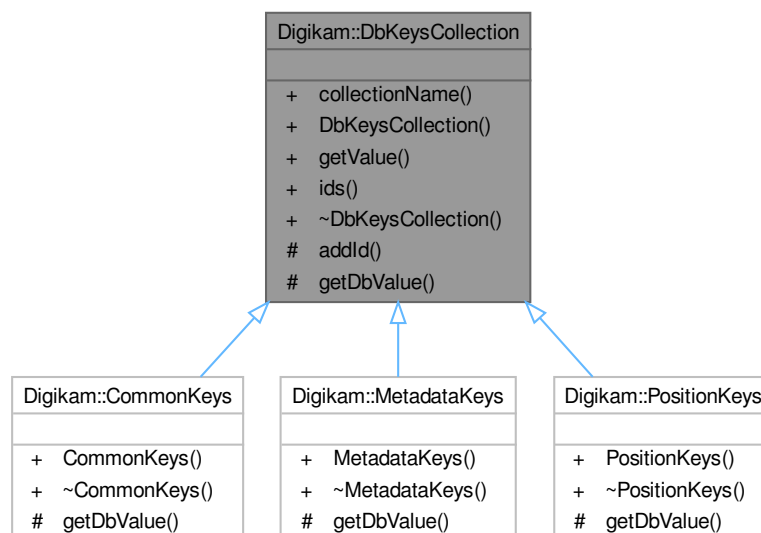
#### Returns

bool: true if the error list is not empty

## 6.312 Digikam::DbKeysCollection Class Reference

A class for managing / grouping database keys.

Inheritance diagram for Digikam::DbKeysCollection:



### Public Member Functions

- `QString` [collectionName](#) () const
- `DbKeysCollection` (const `QString` &n)
- `QString` [getValue](#) (const `QString` &key, `ParseSettings` &settings)
- `DbKeyIdsMap` [ids](#) () const

### Protected Member Functions

- void [addId](#) (const `QString` &id, const `QString` &description)
- virtual `QString` [getDbValue](#) (const `QString` &key, `ParseSettings` &settings)=0



## 6.312.1 Detailed Description

This class manages database keys and provides methods to get the appropriate value from the database.

## 6.312.2 Constructor & Destructor Documentation

### 6.312.2.1 DbKeysCollection()

```
Digikam::DbKeysCollection::DbKeysCollection (
    const QString & n ) [explicit]
```

Default constructor.

Parameters

<i>n</i>	collection name
----------	-----------------

## 6.312.3 Member Function Documentation

### 6.312.3.1 addId()

```
void Digikam::DbKeysCollection::addId (
    const QString & id,
    const QString & description ) [protected]
```

Add an ID to the key collection.

Parameters

<i>id</i>	the id of the database key
<i>description</i>	a short description of the database key

### 6.312.3.2 collectionName()

```
QString Digikam::DbKeysCollection::collectionName ( ) const
```

Get the name of the DbKeysCollection

Returns

the name of the collection

### 6.312.3.3 getDbValue()

```
virtual QString Digikam::DbKeysCollection::getDbValue (
    const QString & key,
    ParseSettings & settings ) [protected], [pure virtual]
```

Abstract method for retrieving the value from the database for the given key.

This method has to be implemented by all child classes. It is called by the [getValue\(\)](#) method.

**Parameters**

<i>key</i>	the key representing the value in the database
<i>settings</i>	the ParseSettings object holding all relevant information about the image.

**Returns**

the value of the given database key

**See also**

[DbKeysCollection::getValue\(\)](#)

Implemented in [Digikam::CommonKeys](#), [Digikam::MetadataKeys](#), and [Digikam::PositionKeys](#).

**6.312.3.4 getValue()**

```
QString Digikam::DbKeysCollection::getValue (
    const QString & key,
    ParseSettings & settings )
```

Get a value from the database.

**Parameters**

<i>key</i>	the key representing the value in the database
<i>settings</i>	the ParseSettings object holding all relevant information about the image.

**Returns**

the value of the given database key

**6.312.3.5 ids()**

```
DbKeyIdMap Digikam::DbKeysCollection::ids ( ) const
```

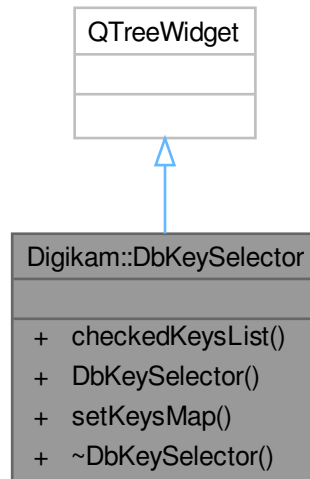
Get all IDs associated with this key collection.

**Returns**

a map of all associated ids and their description

## 6.313 Digikam::DbKeySelector Class Reference

Inheritance diagram for Digikam::DbKeySelector:

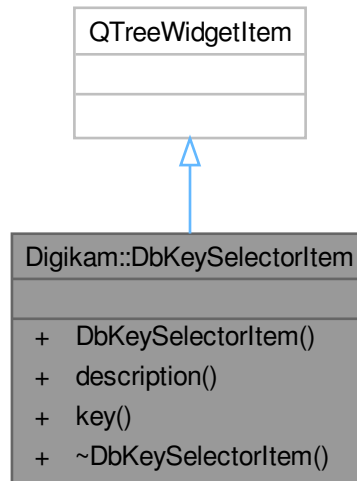


### Public Member Functions

- `QStringList checkedKeysList ()`
- `DbKeySelector (QWidget *const parent)`
- `void setKeysMap (const DbOptionKeysMap &map)`

## 6.314 Digikam::DbKeySelectorItem Class Reference

Inheritance diagram for Digikam::DbKeySelectorItem:

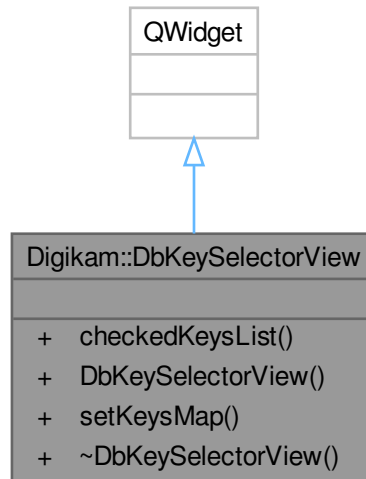


### Public Member Functions

- **DbKeySelectorItem** ([DbHeaderListItem](#) \*const parent, const QString &title, const QString &desc)
- QString **description** () const
- QString **key** () const

## 6.315 Digikam::DbKeySelectorView Class Reference

Inheritance diagram for Digikam::DbKeySelectorView:

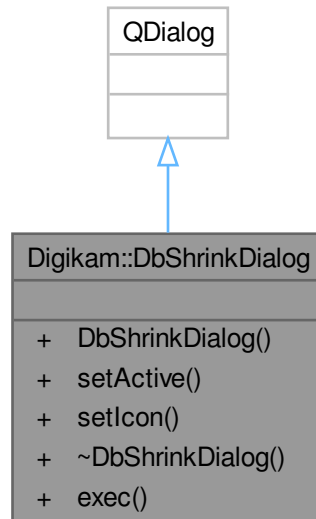


### Public Member Functions

- `QStringList checkedKeysList () const`
- `DbKeySelectorView (QWidget *const parent)`
- `void setKeysMap (const DbOptionKeysMap &map)`

## 6.316 Digikam::DbShrinkDialog Class Reference

Inheritance diagram for Digikam::DbShrinkDialog:



### Public Slots

- int **exec** () override

### Public Member Functions

- **DbShrinkDialog** (QWidget \*const parent)
- void **setActive** (const int pos)
- void **setIcon** (const int pos, const QIcon &icon)

## 6.317 Digikam::DBStatDlg Class Reference

Inheritance diagram for Digikam::DBStatDlg:



### Public Member Functions

- **DBStatDlg** (QWidget \*const parent)

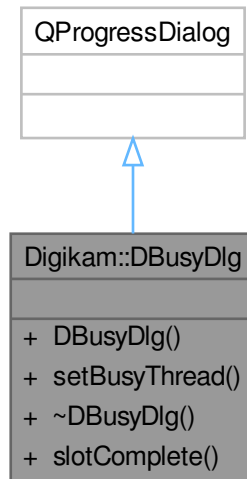
### Public Member Functions inherited from [Digikam::InfoDlg](#)

- QDialogButtonBox \* **buttonBox** () const
- **InfoDlg** (QWidget \*const parent)
- QTreeWidget \* **listView** () const
- QWidget \* **mainWidget** () const
- virtual void **setInfoMap** (const QMap< QString, QString > &list)
- QTabWidget \* **tableView** () const



## 6.318 Digikam::DBusyDlg Class Reference

Inheritance diagram for Digikam::DBusyDlg:



### Public Slots

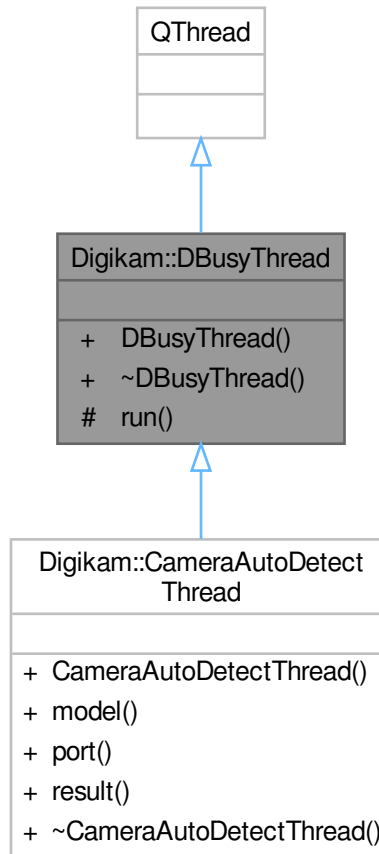
- void **slotComplete** ()

### Public Member Functions

- **DBusyDlg** (const QString &txt, QWidget \*const parent=nullptr)
- void **setBusyThread** ([DBusyThread](#) \*const thread)

## 6.319 Digikam::DBusyThread Class Reference

Inheritance diagram for Digikam::DBusyThread:



### Signals

- void **signalComplete** ()

### Public Member Functions

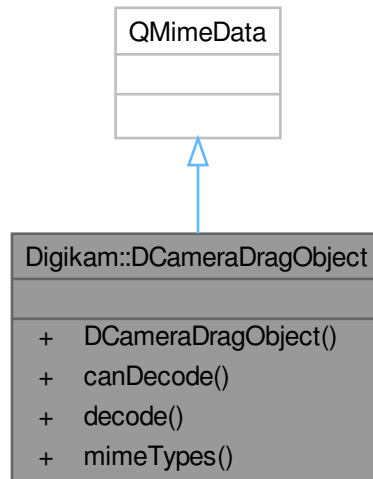
- **DBusyThread** (QObject \*const parent)

### Protected Member Functions

- void **run** () override  
*Reimplement this method with your code to run in a separate thread.*

## 6.320 Digikam::DCameraDragObject Class Reference

Inheritance diagram for Digikam::DCameraDragObject:



### Public Member Functions

- `DCameraDragObject` (const [CameraType](#) &ctype)

### Static Public Member Functions

- static bool **canDecode** (const `QMimeData *e`)
- static bool **decode** (const `QMimeData *e`, [CameraType](#) &ctype)
- static `QStringList` **mimeTypees** ()

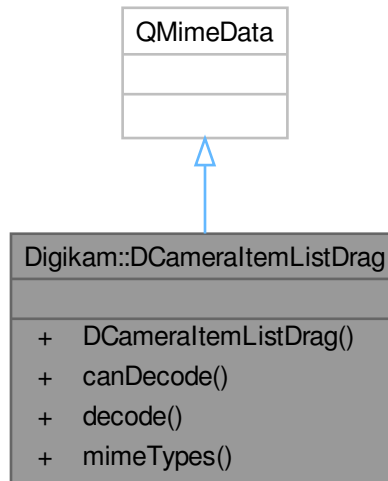
### 6.320.1 Detailed Description

Provides a drag object for a camera object

When a camera object is moved through drag'n'drop an object of this class is created.

## 6.321 Digikam::DCameraltemListDrag Class Reference

Inheritance diagram for Digikam::DCameraltemListDrag:



### Public Member Functions

- **DCameraltemListDrag** (const QStringList &cameraltemPaths)

### Static Public Member Functions

- static bool **canDecode** (const QMimeData \*e)
- static bool **decode** (const QMimeData \*e, QStringList &cameraltemPaths)
- static QStringList **mimeTypes** ()

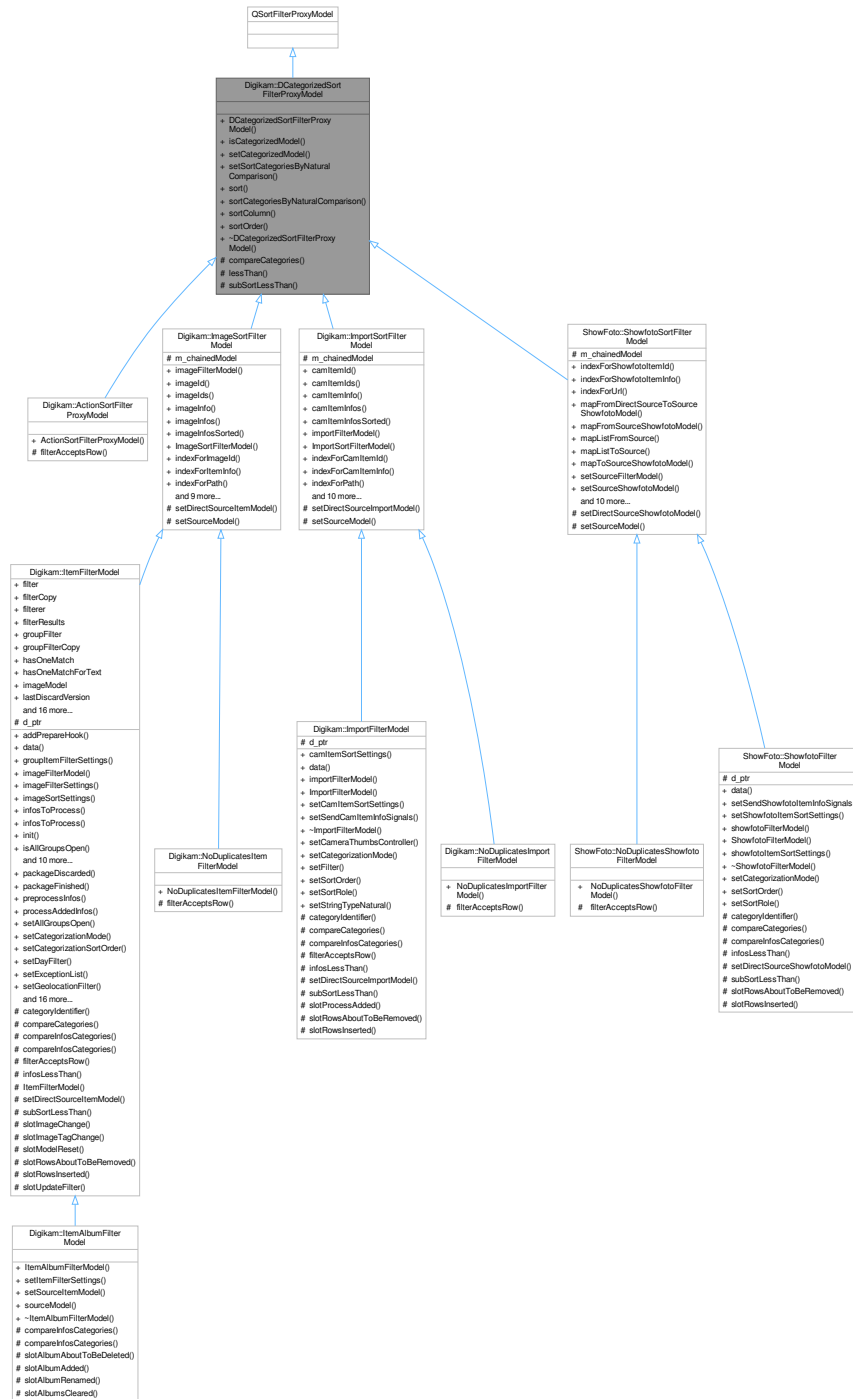
### 6.321.1 Detailed Description

Provides a drag object for a list of camera items

When a camera item is moved through drag'n'drop an object of this class is created.

## 6.322 Digikam::DCategorizedSortFilterProxyModel Class Reference

Inheritance diagram for Digikam::DCategorizedSortFilterProxyModel:



### Classes

- class [Private](#)

## Public Types

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## Public Member Functions

- **DCategorizedSortFilterProxyModel** (QObject \*const parent=nullptr)
- bool [isCategorizedModel](#) () const
- void [setCategorizedModel](#) (bool categorizedModel)
- void [setSortCategoriesByNaturalComparison](#) (bool [sortCategoriesByNaturalComparison](#))
- void [sort](#) (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool [sortCategoriesByNaturalComparison](#) () const
- int [sortColumn](#) () const
- Qt::SortOrder [sortOrder](#) () const

## Protected Member Functions

- virtual int [compareCategories](#) (const QModelIndex &left, const QModelIndex &right) const
- bool [lessThan](#) (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool [subSortLessThan](#) (const QModelIndex &left, const QModelIndex &right) const

### 6.322.1 Detailed Description

This class lets you categorize a view. It is meant to be used along with [DCategorizedView](#) class.

In general terms all you need to do is to reimplement [subSortLessThan\(\)](#) and [compareCategories\(\)](#) methods. In order to make categorization work, you need to also call [setCategorizedModel\(\)](#) class to enable it, since the categorization is disabled by default.

### 6.322.2 Member Enumeration Documentation

#### 6.322.2.1 AdditionalRoles

```
enum Digikam::DCategorizedSortFilterProxyModel::AdditionalRoles
```

#### Enumerator

CategoryDisplayRole	<p><b>Note</b></p> <p>use <code>printf "0x%08X\n" ((\$RANDOM*\$RANDOM))</code> to define additional roles. This role is used for asking the category to a given index</p>
CategorySortRole	<p>This role is used for sorting categories. You can return a string or a long long value. Strings will be sorted alphabetically while long long will be sorted by their value. Please note that this value won't be shown on the view, is only for sorting purposes. What will be shown as "Category" on the view will be asked with the role <a href="#">CategoryDisplayRole</a>.</p>

## 6.322.3 Member Function Documentation

### 6.322.3.1 compareCategories()

```
int Digikam::DCategorizedSortFilterProxyModel::compareCategories (
    const QModelIndex & left,
    const QModelIndex & right ) const [protected], [virtual]
```

This method compares the category of the `left` index with the category of the `right` index.

Internally and if not reimplemented, this method will ask for `left` and `right` models for role `CategorySortRole`. In order to correctly sort categories, the `data()` method of the model should return a `qulonglong` (or numeric) value, or a `QString` object. `QString` objects will be sorted with `QString::localeAwareCompare` if `sortCategoriesByNaturalComparison()` is true.

#### Note

Please have present that: `QString(QChar(QChar::ObjectReplacementCharacter)) > QString(QChar(QChar::ReplacementCharacter)) > [ all possible strings ] > QString();`

This means that `QString()` will be sorted the first one, while `QString(QChar(QChar::ObjectReplacementCharacter))` and `QString(QChar(QChar::ReplacementCharacter))` will be sorted in last position.

#### Warning

Please note that `data()` method of the model should return always information of the same type. If you return a `QString` for an index, you should return always `QStrings` for all indexes for role `CategorySortRole` in order to correctly sort categories. You can't mix by returning a `QString` for one index, and a `qulonglong` for other.

#### Note

If you need a more complex layout, you will have to reimplement this method.

#### Returns

A negative value if the category of `left` should be placed before the category of `right`. 0 if `left` and `right` are on the same category, and a positive value if the category of `left` should be placed after the category of `right`.

Reimplemented in [Digikam::ItemFilterModel](#), [ShowFoto::ShowfotoFilterModel](#), and [Digikam::ImportFilterModel](#).

### 6.322.3.2 isCategorizedModel()

```
bool Digikam::DCategorizedSortFilterProxyModel::isCategorizedModel ( ) const
```

#### Returns

whether the model is categorized or not. Disabled by default.

### 6.322.3.3 lessThan()

```
bool Digikam::DCategorizedSortFilterProxyModel::lessThan (
    const QModelIndex & left,
    const QModelIndex & right ) const [override], [protected]
```

Overridden from `QSortFilterProxyModel`. If you are subclassing [DCategorizedSortFilterProxyModel](#), you will probably not need to reimplement this method.

It calls [compareCategories\(\)](#) to sort by category. If the both items are in the same category (i.e. `compareCategories` returns 0), then `subSortLessThan` is called.

#### Returns

Returns true if the item `left` is less than the item `right` when sorting.

#### Warning

You usually won't need to reimplement this method when subclassing from [DCategorizedSortFilterProxyModel](#).

### 6.322.3.4 setCategorizedModel()

```
void Digikam::DCategorizedSortFilterProxyModel::setCategorizedModel (
    bool categorizedModel )
```

Enables or disables the categorization feature.

#### Parameters

<i>categorizedModel</i>	whether to enable or disable the categorization feature.
-------------------------	--

### 6.322.3.5 setSortCategoriesByNaturalComparison()

```
void Digikam::DCategorizedSortFilterProxyModel::setSortCategoriesByNaturalComparison (
    bool sortCategoriesByNaturalComparison )
```

Set if the sorting using `CategorySortRole` will use a natural comparison in the case that strings were returned. If enabled, `QCollator` will be used for sorting.

#### Parameters

<i>sortCategoriesByNaturalComparison</i>	whether to sort using a natural comparison or not.
--	--

### 6.322.3.6 sort()

```
void Digikam::DCategorizedSortFilterProxyModel::sort (
    int column,
    Qt::SortOrder order = Qt::AscendingOrder ) [override]
```



Overridden from `QSortFilterProxyModel`. Sorts the source model using `column` for the given `order`.

### 6.322.3.7 `sortCategoriesByNaturalComparison()`

```
bool Digikam::DCategorizedSortFilterProxyModel::sortCategoriesByNaturalComparison ( ) const
```

#### Returns

whether it is being used a natural comparison for sorting. Enabled by default.

### 6.322.3.8 `sortColumn()`

```
int Digikam::DCategorizedSortFilterProxyModel::sortColumn ( ) const
```

#### Returns

the column being used for sorting.

### 6.322.3.9 `sortOrder()`

```
Qt::SortOrder Digikam::DCategorizedSortFilterProxyModel::sortOrder ( ) const
```

#### Returns

the sort order being used for sorting.

### 6.322.3.10 `subSortLessThan()`

```
bool Digikam::DCategorizedSortFilterProxyModel::subSortLessThan (
    const QModelIndex & left,
    const QModelIndex & right ) const [protected], [virtual]
```

This method has a similar purpose as [lessThan\(\)](#) has on `QSortFilterProxyModel`. It is used for sorting items that are in the same category.

#### Returns

Returns true if the item `left` is less than the item `right` when sorting.

Reimplemented in [Digikam::ItemFilterModel](#), [ShowFoto::ShowfotoFilterModel](#), and [Digikam::ImportFilterModel](#).

## 6.323 Digikam::DCategorizedSortFilterProxyModel::Private Class Reference

### Public Attributes

- bool **categorizedModel** = false
- QCollator **collator**
- bool **sortCategoriesByNaturalComparison** = true
- int **sortColumn** = 0
- Qt::SortOrder **sortOrder** = Qt::AscendingOrder

## 6.324 Digikam::DCategorizedView Class Reference

Item view for listing items.



## Public Member Functions

- virtual QModelIndexList [categorizedIndexesIn](#) (const QRect &rect) const
- virtual QModelIndex [categoryAt](#) (const QPoint &point) const
- [DCategoryDrawer](#) \* [categoryDrawer](#) () const
- virtual QItemSelectionRange [categoryRange](#) (const QModelIndex &index) const
- virtual QRect [categoryVisualRect](#) (const QModelIndex &index) const
- [DQCategorizedView](#) (QWidget \*const parent=nullptr)
- QModelIndex [indexAt](#) (const QPoint &point) const override
- void [setCategoryDrawer](#) ([DCategoryDrawer](#) \*categoryDrawer)
- void [setDrawDraggedItems](#) (bool drawDraggedItems)
- void [setGridSize](#) (const QSize &size)
- void [setModel](#) (QAbstractItemModel \*model) override
- QRect [visualRect](#) (const QModelIndex &index) const override

## Protected Slots

- void [currentChanged](#) (const QModelIndex &current, const QModelIndex &previous) override
- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- virtual void [rowsInsertedArtificial](#) (const QModelIndex &parent, int start, int end)
- virtual void [rowsRemoved](#) (const QModelIndex &parent, int start, int end)
- virtual void [slotLayoutChanged](#) ()
- void [updateGeometries](#) () override

## Protected Member Functions

- void [dragLeaveEvent](#) (QDragLeaveEvent \*event) override
- void [dragMoveEvent](#) (QDragMoveEvent \*event) override
- void [dropEvent](#) (QDropEvent \*event) override
- void [leaveEvent](#) (QEvent \*event) override
- void [mouseMoveEvent](#) (QMouseEvent \*event) override
- void [mousePressEvent](#) (QMouseEvent \*event) override
- void [mouseReleaseEvent](#) (QMouseEvent \*event) override
- QModelIndex [moveCursor](#) (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void [paintEvent](#) (QPaintEvent \*event) override
- void [resizeEvent](#) (QResizeEvent \*event) override
- void [setSelection](#) (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void [startDrag](#) (Qt::DropActions supportedActions) override

### 6.324.1 Detailed Description

[DQCategorizedView](#) allows you to use it as it were a QListView. Subclass [DQCategorizedSortFilterProxyModel](#) to provide category information for items.

### 6.324.2 Member Function Documentation

#### 6.324.2.1 [categorizedIndexesIn\(\)](#)

```
QModelIndexList DigiKam::DQCategorizedView::categorizedIndexesIn (
    const QRect & rect ) const [virtual]
```

This method will return all indexes whose visual rect intersects `rect`.

## Parameters

<i>rect</i>	rectangle to test intersection with
-------------	-------------------------------------

## Note

Returns an empty list if the view is not categorized.

**6.324.2.2 categoryAt()**

```
QModelIndex Digikam::DCategorizedView::categoryAt (
    const QPoint & point ) const [virtual]
```

This method will return the first index of the category in the region of which *point* is found.

## Note

Returns QModelIndex() if the view is not categorized.

**6.324.2.3 categoryRange()**

```
QItemSelectionRange Digikam::DCategorizedView::categoryRange (
    const QModelIndex & index ) const [virtual]
```

This method returns the range of indexes contained in the category in which *index* is sorted.

## Note

Returns an empty range if the view is no categorized.

**6.324.2.4 categoryVisualRect()**

```
QRect Digikam::DCategorizedView::categoryVisualRect (
    const QModelIndex & index ) const [virtual]
```

This method will return the visual rect of the header of the category in which *index* is sorted.

## Note

Returns QRect() if the view is not categorized.

**6.324.2.5 setDrawDraggedItems()**

```
void Digikam::DCategorizedView::setDrawDraggedItems (
    bool drawDraggedItems )
```

Switch on drawing of dragged items. Default: on. While dragging over the view, dragged items will be drawn transparently following the mouse cursor.

## Parameters

<code>drawDraggedItems</code>	if <code>true</code> , dragged items will be drawn
-------------------------------	--

## 6.325 Digikam::DCategorizedView::Private Class Reference

### Classes

- class [ElementInfo](#)

*Attributes.*

### Public Member Functions

- const QRect & [cacheCategory](#) (const QString &category)
- const QRect & [cachedRectCategory](#) (const QString &category)
- const QRect & [cachedRectIndex](#) (const QModelIndex &index)
- const QRect & [cacheIndex](#) (const QModelIndex &index)
- int [categoryUpperBound](#) ([SparseModelIndexVector](#) &modelIndexList, int begin, int averageSize=0)
- QRect [categoryVisualRect](#) (const QString &category)
- QSize [contentsSize](#) ()
- void [drawDraggedItems](#) ()
- void [drawDraggedItems](#) (QPainter \*painter)
- void [drawNewCategory](#) (const QModelIndex &index, int sortRole, const QStyleOption &option, QPainter \*painter)
- const QModelIndexList & [intersectionSet](#) (const QRect &rect)
- **Private** ([DCategorizedView](#) \*const [listView](#))
- QItemSelection [selectionForRect](#) (const QRect &rect)
- void [updateScrollbars](#) ()
- QRect [visualCategoryRectInViewport](#) (const QString &category) const
- QRect [visualRect](#) (const QModelIndex &index)
- QRect [visualRectInViewport](#) (const QModelIndex &index) const

### Public Attributes

- QSize **biggestItemSize** = QSize(0, 0)
- QStringList **categories**
- QHash< QString, QVector< int > > **categoriesIndexes**
- QHash< QString, QRect > **categoriesPosition**
- [DCategoryDrawer](#) \* **categoryDrawer** = nullptr
- bool **dragLeftViewport** = false
- bool **drawItemsWhileDragging** = true
- QVector< [ElementInfo](#) > **elementsInfo**
- QHash< int, QRect > **elementsPosition**
- int **forcedSelectionPosition** = 0
- QModelIndex **hovered**
- QString **hoveredCategory**
- QPoint **initialPressPosition**
- QModelIndexList **intersectedIndexes**
- QItemSelection **lastCategorySelection**
- QRect **lastDraggedItemsRect**

- QItemSelection **lastSelection**
- [DCategorizedView](#) \* **listView** = nullptr
  - Basic data.*
- bool **mouseButtonPressed** = false
  - Behavior data.*
- QPoint **mousePosition**
- [DCategorizedSortFilterProxyModel](#) \* **proxyModel** = nullptr
  - Attributes for speed reasons.*
- bool **rightMouseButtonPressed** = false

## 6.325.1 Member Function Documentation

### 6.325.1.1 cacheCategory()

```
const QRect & Digikam::DCategorizedView::Private::cacheCategory (
    const QString & category )
```

Caches and returns the rect that corresponds to `category`

We're sure `categoriesPosition` doesn't contain `category`

### 6.325.1.2 cachedRectCategory()

```
const QRect & Digikam::DCategorizedView::Private::cachedRectCategory (
    const QString & category )
```

Returns the rect that corresponds to `category`

#### Note

If the rect is not cached, it becomes cached

### 6.325.1.3 cachedRectIndex()

```
const QRect & Digikam::DCategorizedView::Private::cachedRectIndex (
    const QModelIndex & index )
```

Returns the rect that corresponds to `index`

#### Note

If the rect is not cached, it becomes cached

### 6.325.1.4 cacheIndex()

```
const QRect & Digikam::DCategorizedView::Private::cacheIndex (
    const QModelIndex & index )
```

Caches and returns the rect that corresponds to `index`

We're sure `elementsPosition` doesn't contain `index`

### 6.325.1.5 categoryUpperBound()

```
int Digikam::DCategorizedView::Private::categoryUpperBound (
    SparseModelIndexVector & modelIndexList,
    int begin,
    int averageSize = 0 )
```

This method will, starting from the index at `begin` in the given (sorted) `modelIndexList`, find the last index having the same category as the index to begin with.

### 6.325.1.6 categoryVisualRect()

```
QRect Digikam::DCategorizedView::Private::categoryVisualRect (
    const QString & category )
```

Returns the visual rect (taking in count x and y offsets) for `category`

#### Note

If the rect is not cached, it becomes cached

### 6.325.1.7 contentsSize()

```
QSize Digikam::DCategorizedView::Private::contentsSize ( )
```

Returns the contents size of this view (topmost category to bottommost index + spacing)

### 6.325.1.8 drawDraggedItems() [1/2]

```
void Digikam::DCategorizedView::Private::drawDraggedItems ( )
```

This method will determine which rect needs to be updated because of a dragging operation

### 6.325.1.9 drawDraggedItems() [2/2]

```
void Digikam::DCategorizedView::Private::drawDraggedItems (
    QPainter * painter )
```

This method will draw dragged items in the painting operation

### 6.325.1.10 drawNewCategory()

```
void Digikam::DCategorizedView::Private::drawNewCategory (
    const QModelIndex & index,
    int sortRole,
    const QStyleOption & option,
    QPainter * painter )
```

This method will draw a new category represented by index `index` on the rect specified by `option.rect`, with painter `painter`



#### 6.325.1.11 intersectionSet()

```
const QModelIndexList & Digikam::DCategorizedView::Private::intersectionSet (
    const QRect & rect )
```

Returns the list of items that intersects with `rect`

#### 6.325.1.12 selectionForRect()

```
QItemSelection Digikam::DCategorizedView::Private::selectionForRect (
    const QRect & rect )
```

Returns a `QItemSelection` for all items intersection `rect`.

#### 6.325.1.13 updateScrollbars()

```
void Digikam::DCategorizedView::Private::updateScrollbars ( )
```

This method will update scrollbars ranges. Called when our model changes or when the view is resized

#### 6.325.1.14 visualCategoryRectInViewport()

```
QRect Digikam::DCategorizedView::Private::visualCategoryRectInViewport (
    const QString & category ) const
```

Returns the category rect in the viewport for `category`

#### 6.325.1.15 visualRect()

```
QRect Digikam::DCategorizedView::Private::visualRect (
    const QModelIndex & index )
```

Returns the visual rect (taking in count x and y offsets) for `index`

#### Note

If the rect is not cached, it becomes cached

#### 6.325.1.16 visualRectInViewport()

```
QRect Digikam::DCategorizedView::Private::visualRectInViewport (
    const QModelIndex & index ) const
```

Gets the item rect in the viewport for `index`

## 6.325.2 Member Data Documentation

### 6.325.2.1 elementsInfo

`QVector<ElementInfo> Digikam::DCategorizedView::Private::elementsInfo`

Cache data We cannot merge some of them into structs because it would affect performance

## 6.326 Digikam::DCategorizedView::Private::ElementInfo Class Reference

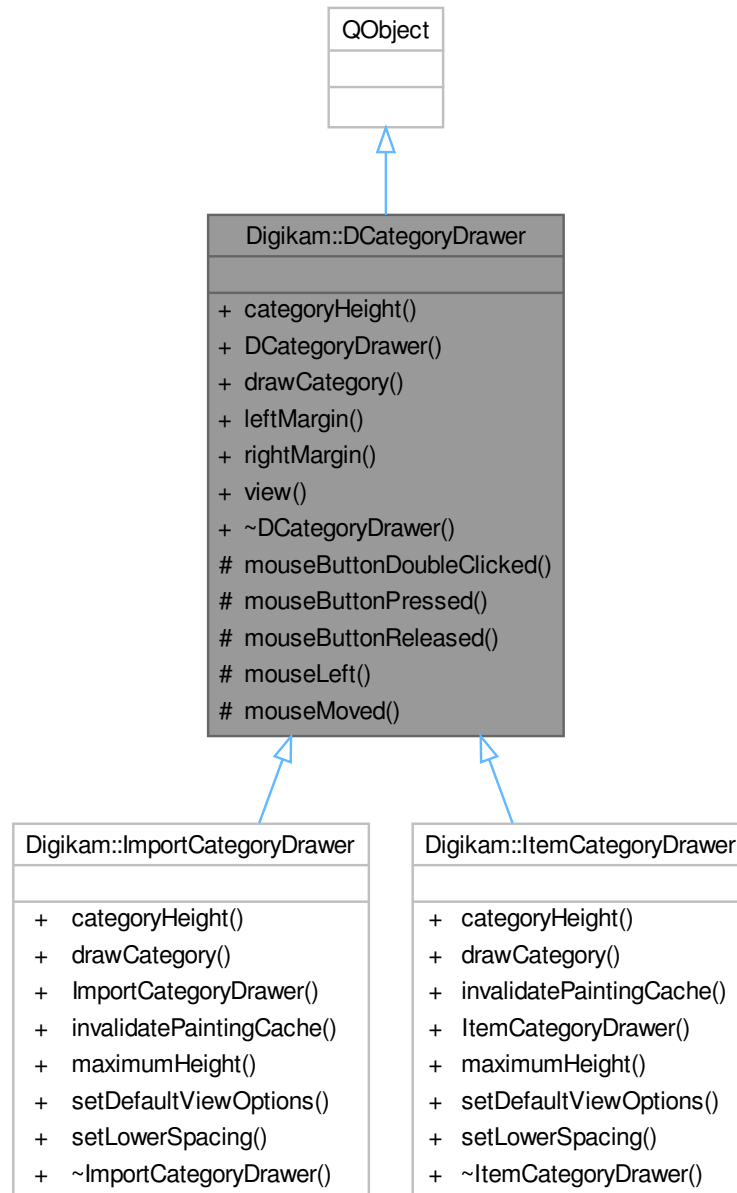
Attributes.

### Public Attributes

- QString **category**
- int **relativeOffsetToCategory** = 0

## 6.327 Digikam::DCategoryDrawer Class Reference

Inheritance diagram for Digikam::DCategoryDrawer:



### Signals

- void `actionRequested` (int action, const QModelIndex &index)
- void `collapseOrExpandClicked` (const QModelIndex &index)

## Public Member Functions

- virtual int [categoryHeight](#) (const QModelIndex &index, const QStyleOption &option) const
- [DCategoryDrawer](#) ([DategorizedView](#) \*const view)
- virtual void [drawCategory](#) (const QModelIndex &index, int sortRole, const QStyleOption &option, QPainter \*painter) const
- virtual int [leftMargin](#) () const
- virtual int [rightMargin](#) () const
- [DategorizedView](#) \* view () const

## Protected Member Functions

- virtual void [mouseButtonDoubleClicked](#) (const QModelIndex &index, const QRect &blockRect, QMouseEvent \*event)
- virtual void [mouseButtonPressed](#) (const QModelIndex &index, const QRect &blockRect, QMouseEvent \*event)
- virtual void [mouseButtonReleased](#) (const QModelIndex &index, const QRect &blockRect, QMouseEvent \*event)
- virtual void [mouseLeft](#) (const QModelIndex &index, const QRect &blockRect)
- virtual void [mouseMoved](#) (const QModelIndex &index, const QRect &blockRect, QMouseEvent \*event)

## Friends

- class [DategorizedView](#)

### 6.327.1 Detailed Description

The category drawing is performed by this class. It also gives information about the category height and margins.

### 6.327.2 Constructor & Destructor Documentation

#### 6.327.2.1 DCategoryDrawer()

```
Digikam::DCategoryDrawer::DCategoryDrawer (
    DategorizedView *const view ) [explicit]
```

Construct a category drawer for a given view

### 6.327.3 Member Function Documentation

#### 6.327.3.1 actionRequested

```
void Digikam::DCategoryDrawer::actionRequested (
    int action,
    const QModelIndex & index ) [signal]
```

Emit this signal on your subclass implementation to notify that something happened. Usually this will be triggered when you have received an event, and its position matched some "hot spot".

You give this action the integer you want, and having connected this signal to your code, the connected slot can perform the needed changes (view, model, selection model, delegate...)

### 6.327.3.2 categoryHeight()

```
int Digikam::DCategoryDrawer::categoryHeight (
    const QModelIndex & index,
    const QStyleOption & option ) const [virtual]
```

#### Returns

The category height for the category represented by index *index* with style options *option*.

Reimplemented in [Digikam::ItemCategoryDrawer](#), and [Digikam::ImportCategoryDrawer](#).

### 6.327.3.3 collapseOrExpandClicked

```
void Digikam::DCategoryDrawer::collapseOrExpandClicked (
    const QModelIndex & index ) [signal]
```

This signal becomes emitted when collapse or expand has been clicked.

### 6.327.3.4 drawCategory()

```
void Digikam::DCategoryDrawer::drawCategory (
    const QModelIndex & index,
    int sortRole,
    const QStyleOption & option,
    QPainter * painter ) const [virtual]
```

This method purpose is to draw a category represented by the given

#### Parameters

<i>index</i>	with the given
<i>sortRole</i>	sorting role
<i>option</i>	painter style options
<i>painter</i>	painter instance

#### Note

This method will be called one time per category, always with the first element in that category

Reimplemented in [Digikam::ItemCategoryDrawer](#), and [Digikam::ImportCategoryDrawer](#).

### 6.327.3.5 leftMargin()

```
int Digikam::DCategoryDrawer::leftMargin ( ) const [virtual]
```

#### Note

0 by default

**6.327.3.6 mouseButtonDoubleClicked()**

```
void Digikam::DCategoryDrawer::mouseButtonDoubleClicked (
    const QModelIndex & index,
    const QRect & blockRect,
    QMouseEvent * event ) [protected], [virtual]
```

Method called when the mouse button has been double clicked.

**Parameters**

<i>index</i>	The representative index of the block of items.
<i>blockRect</i>	The rect occupied by the block of items.
<i>event</i>	The mouse event.

**Warning**

You explicitly have to determine whether the event has been accepted or not. You have to call `event->accept()` or `event->ignore()` at all possible case branches in your code.

**6.327.3.7 mouseButtonPressed()**

```
void Digikam::DCategoryDrawer::mouseButtonPressed (
    const QModelIndex & index,
    const QRect & blockRect,
    QMouseEvent * event ) [protected], [virtual]
```

Method called when the mouse button has been pressed.

**Parameters**

<i>index</i>	The representative index of the block of items.
<i>blockRect</i>	The rect occupied by the block of items.
<i>event</i>	The mouse event.

**Warning**

You explicitly have to determine whether the event has been accepted or not. You have to call `event->accept()` or `event->ignore()` at all possible case branches in your code.

**6.327.3.8 mouseButtonReleased()**

```
void Digikam::DCategoryDrawer::mouseButtonReleased (
    const QModelIndex & index,
    const QRect & blockRect,
    QMouseEvent * event ) [protected], [virtual]
```

Method called when the mouse button has been released.

## Parameters

<i>index</i>	The representative index of the block of items.
<i>blockRect</i>	The rect occupied by the block of items.
<i>event</i>	The mouse event.

## Warning

You explicitly have to determine whether the event has been accepted or not. You have to call `event->accept()` or `event->ignore()` at all possible case branches in your code.

**6.327.3.9 mouseLeft()**

```
void Digikam::DCategoryDrawer::mouseLeft (
    const QModelIndex & index,
    const QRect & blockRect ) [protected], [virtual]
```

Method called when the mouse button has left this block.

## Parameters

<i>index</i>	The representative index of the block of items.
<i>blockRect</i>	The rect occupied by the block of items.

**6.327.3.10 mouseMoved()**

```
void Digikam::DCategoryDrawer::mouseMoved (
    const QModelIndex & index,
    const QRect & blockRect,
    QMouseEvent * event ) [protected], [virtual]
```

Method called when the mouse has been moved.

## Parameters

<i>index</i>	The representative index of the block of items.
<i>blockRect</i>	The rect occupied by the block of items.
<i>event</i>	The mouse event.

**6.327.3.11 rightMargin()**

```
int Digikam::DCategoryDrawer::rightMargin ( ) const [virtual]
```

## Note

0 by default

### 6.327.3.12 view()

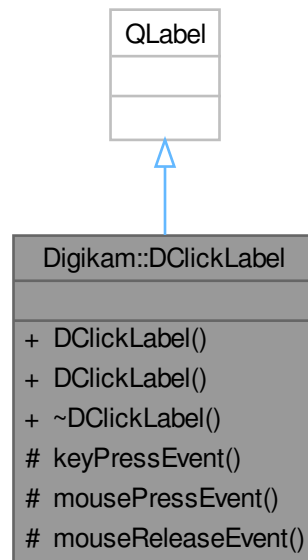
```
DCategorizedView * Digikam::DCategoryDrawer::view ( ) const
```

#### Returns

The view this category drawer is associated with.

## 6.328 Digikam::DClickLabel Class Reference

Inheritance diagram for Digikam::DClickLabel:



#### Signals

- void **activated** ()  
*Emitted when activated, by mouse or key press.*
- void **leftClicked** ()  
*Emitted when activated by left mouse click.*

#### Public Member Functions

- **DClickLabel** (const QString &text, QWidget \*const parent=nullptr)
- **DClickLabel** (QWidget \*const parent=nullptr)



**Protected Member Functions**

- void **keyPressEvent** (QKeyEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override

**6.329 Digikam::DColor Class Reference****Public Member Functions**

- int **alpha** () const
- void **blendAdd** (const DColor &src)
- void **blendAlpha16** (int alpha)
- void **blendAlpha8** (int alpha)
- void **blendClamp16** ()
- void **blendClamp8** ()
- void **blendInvAlpha16** (int alpha)
- void **blendInvAlpha8** (int alpha)
- void **blendZero** ()
- int **blue** () const
- void **convertToEightBit** ()
- void **convertToSixteenBit** ()
- DColor ()=default
- DColor (const QColor &color, bool sixteenBit=false)
- DColor (int red, int green, int blue, int alpha, bool sixteenBit)
- DColor (uchar \*data, bool sixteenBit=false)
- void **demultiply** ()
- void **demultiply16** (int alpha)
- void **demultiply8** (int alpha)
- void **getHSL** (int \*const h, int \*const s, int \*const l) const
- QColor **getQColor** () const
- void **getYCbCr** (double \*const y, double \*const cb, double \*const cr) const
- int **green** () const
- bool **isPureGray** ()
- bool **isPureGrayValue** (int v)
- void **multiply** (float factor)
- void **premultiply** ()
- void **premultiply16** (int alpha)
- void **premultiply8** (int alpha)
- int **red** () const
- void **setAlpha** (int alpha)
- void **setBlue** (int blue)
- void **setColor** (uchar \*const data, bool sixteenBit=false)
- void **setGreen** (int green)
- void **setHSL** (int h, int s, int l, bool sixteenBit)
- void **setPixel** (uchar \*const data) const
- void **setRed** (int red)
- void **setSixteenBit** (bool sixteenBit)
- void **setYCbCr** (double y, double cb, double cr, bool sixteenBit)
- bool **sixteenBit** () const

## 6.329.1 Constructor & Destructor Documentation

### 6.329.1.1 DColor() [1/4]

```
Digikam::DColor::DColor ( ) [default]
```

Initialize with default value, fully transparent eight bit black

### 6.329.1.2 DColor() [2/4]

```
Digikam::DColor::DColor (
    uchar * data,
    bool sixteenBit = false ) [inline], [explicit]
```

Read value from data. Equivalent to [setColor\(\)](#)

### 6.329.1.3 DColor() [3/4]

```
Digikam::DColor::DColor (
    int red,
    int green,
    int blue,
    int alpha,
    bool sixteenBit ) [inline]
```

Initialize with given RGBA values

### 6.329.1.4 DColor() [4/4]

```
Digikam::DColor::DColor (
    const QColor & color,
    bool sixteenBit = false ) [explicit]
```

Read values from QColor, convert to sixteenBit if sixteenBit is true

## 6.329.2 Member Function Documentation

### 6.329.2.1 blendZero()

```
void Digikam::DColor::blendZero ( ) [inline]
```

Inline alpha blending helper functions. These functions are used by [DColorComposer](#). Look at that code to learn how to use them for composition if you want to use them in optimized code.

### 6.329.2.2 convertToSixteenBit()

```
void Digikam::DColor::convertToSixteenBit ( )
```

Convert the color values of this color to and from sixteen bit and set the sixteenBit value accordingly

### 6.329.2.3 getHSL()

```
void Digikam::DColor::getHSL (
    int *const h,
    int *const s,
    int *const l ) const
```

Return the current RGB color values of this color in the HSL color space. Alpha is ignored for the conversion.

### 6.329.2.4 getYCbCr()

```
void Digikam::DColor::getYCbCr (
    double *const y,
    double *const cb,
    double *const cr ) const
```

Return the current RGB color values of this color in the YCrCb color space. Alpha is ignored for the conversion.

### 6.329.2.5 premultiply()

```
void Digikam::DColor::premultiply ( ) [inline]
```

Premultiply and demultiply this color. [DImg](#) stores the color non-premultiplied. Inline methods.

### 6.329.2.6 setColor()

```
void Digikam::DColor::setColor (
    uchar *const data,
    bool sixteenBit = false ) [inline]
```

Read color values as RGBA from the given memory location. If sixteenBit is false, 4 bytes are read. If sixteenBit is true, 8 bytes are read. Inline method.

These methods are used in quite a few image effects, typically in loops iterating the data. Providing them as inline methods allows the compiler to optimize better.

### 6.329.2.7 setHSL()

```
void Digikam::DColor::setHSL (
    int h,
    int s,
    int l,
    bool sixteenBit )
```

Set the RGB color values of this color to the given HSL values converted to RGB. Alpha is set to be fully opaque. sixteenBit determines both how the HSL values are interpreted and the sixteenBit value of this color after this operation.

### 6.329.2.8 setPixel()

```
void Digikam::DColor::setPixel (
    uchar *const data ) const [inline]
```

Write the values of this color to the given memory location. If sixteenBit is false, 4 bytes are written. If sixteenBit is true, 8 bytes are written. Inline method.

### 6.329.2.9 setYCbCr()

```
void Digikam::DColor::setYCbCr (
    double y,
    double cb,
    double cr,
    bool sixteenBit )
```

Set the RGB color values of this color to the given YCrCb values converted to RGB. Alpha is set to be fully opaque. sixteenBit determines both how the YCrCb values are interpreted and the sixteenBit value of this color after this operation.

## 6.330 Digikam::DColorComposer Class Reference

Inheritance diagram for Digikam::DColorComposer:



### Public Types

- enum [CompositingOperation](#) {
  - PorterDuffNone** , **PorterDuffClear** , **PorterDuffSrc** , **PorterDuffSrcOver** ,
  - PorterDuffDstOver** , **PorterDuffSrcIn** , **PorterDuffDstIn** , **PorterDuffSrcOut** ,
  - PorterDuffDstOut** , **PorterDuffSrcAtop** , **PorterDuffDstAtop** , **PorterDuffXor** }
- enum **MultiplicationFlags** {
  - NoMultiplication** = 0x00 , **PremultiplySrc** = 0x01 , **PremultiplyDst** = 0x02 , **DemultiplyDst** = 0x04 ,
  - MultiplicationFlagsDImg** = PremultiplySrc | PremultiplyDst | DemultiplyDst , **MultiplicationFlags** ←
  - PremultipliedColorOnDImg** = PremultiplyDst | DemultiplyDst }

### Public Member Functions

- virtual void [compose](#) (DColor &dest, DColor &src)=0
- virtual void [compose](#) (DColor &dest, DColor &src, MultiplicationFlags multiplicationFlags)

### Static Public Member Functions

- static [DColorComposer](#) \* [getComposer](#) (CompositingOperation rule)

## 6.330.1 Member Enumeration Documentation

### 6.330.1.1 CompositingOperation

```
enum Digikam::DColorComposer::CompositingOperation
```

The available rules to combine src and destination color.

For the Porter-Duff rules, the formula is  $\text{component} = (\text{source} * \text{fs} + \text{destination} * \text{fd})$  where fs, fd according to the following table with sa = source alpha, da = destination alpha:

None fs: sa fd: 1.0-sa Clear fs: 0.0 fd: 0.0 Src fs: 1.0 fd: 0.0 Src Over fs: 1.0 fd: 1.0-sa Dst Over fs: 1.0-da fd: 1.0 Src In fs: da fd: 0.0 Dst In fs: 0.0 fd: sa Src Out fs: 1.0-da fd: 0.0 Dst Out fs: 0.0 fd: 1.0-sa

Src Atop fs: da fd: 1.0-sa Dst Atop fs: 1.0-da fd: sa Xor fs: 1.0-da fd: 1.0-sa

None is the default, classical blending mode, a "Src over" simplification: Blend non-premultiplied RGBA data "src over" a fully opaque background. Src is the painter's algorithm. All other operations require premultiplied colors. The documentation of `java.awt.AlphaComposite` (Java 1.5) provides a good introduction and documentation on Porter Duff.

## 6.330.2 Member Function Documentation

### 6.330.2.1 compose() [1/2]

```
virtual void Digikam::DColorComposer::compose (
    DColor & dest,
    DColor & src ) [pure virtual]
```

Carry out the actual composition process. Src and Dest are composed and the result is written to dest. No pre-/demultiplication is done by this method, use the other overloaded methods, which call this method, if you need pre- or demultiplication (you need it if any of the colors are read from or written to a [DImg](#)).

If you just pass the object to a [DImg](#) method, you do not need to call this. Call this function if you want to compose two colors. Implement this function if you create a custom [DColorComposer](#).

The bit depth of source and destination color must be identical.

### 6.330.2.2 compose() [2/2]

```
void Digikam::DColorComposer::compose (
    DColor & dest,
    DColor & src,
    DColorComposer::MultiplicationFlags multiplicationFlags ) [virtual]
```

Compose the two colors by calling `compose(dest, src)`. Pre- and demultiplication operations are done as specified. For PorterDuff operations except PorterDuffNone, you need

- PremultiplySrc if src is not premultiplied (read from a [DImg](#))
- PremultiplyDst if dst is not premultiplied (read from a [DImg](#))
- DemultiplyDst if dst will be written to non-premultiplied data (a [DImg](#))

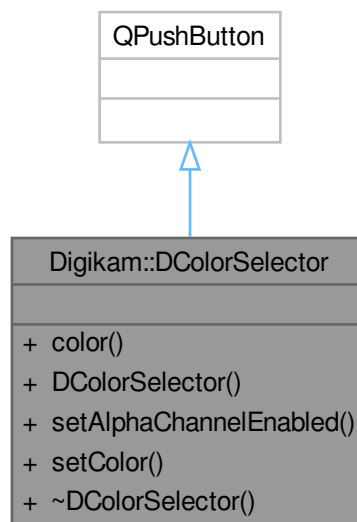
### 6.330.2.3 getComposer()

```
DColorComposer * Digikam::DColorComposer::getComposer (
    DColorComposer::CompositingOperation rule ) [static]
```

Retrieve a [DColorComposer](#) object for one of the predefined rules. The object needs to be deleted by the caller.

## 6.331 Digikam::DColorSelector Class Reference

Inheritance diagram for Digikam::DColorSelector:



### Signals

- void **signalColorSelected** (const QColor &)

### Public Member Functions

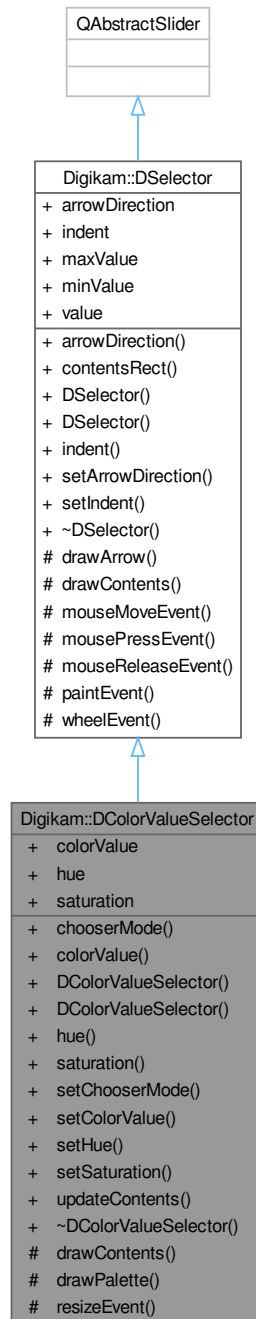
- QColor **color** () const
- **DColorSelector** (QWidget \*const parent=nullptr)
- void **setAlphaChannelEnabled** (bool)
- void **setColor** (const QColor &color)

### 6.331.1 Detailed Description

A widget to choose a color from a palette.

## 6.332 Digikam::DColorValueSelector Class Reference

Inheritance diagram for Digikam::DColorValueSelector:



### Public Member Functions

- DColorChooserMode `chooserMode` () const
- int `colorValue` () const

- **DColorValueSelector** (Qt::Orientation o, QWidget \*const parent=nullptr)
- **DColorValueSelector** (QWidget \*const parent=nullptr)
- int **hue** () const
- int **saturation** () const
- void **setChooserMode** (DColorChooserMode chooserMode)
- void **setColorValue** (int colorValue)
- void **setHue** (int hue)
- void **setSaturation** (int saturation)
- void **updateContents** ()

### Public Member Functions inherited from [Digikam::DSelector](#)

- Qt::ArrowType **arrowDirection** () const
- QRect **contentsRect** () const
- **DSelector** (Qt::Orientation o, QWidget \*const parent=nullptr)
- **DSelector** (QWidget \*const parent=nullptr)
- bool **indent** () const
- void **setArrowDirection** (Qt::ArrowType direction)
- void **setIndent** (bool i)

### Protected Member Functions

- void **drawContents** (QPainter \*) override
- virtual void **drawPalette** (QPixmap \*)
- void **resizeEvent** (QResizeEvent \*) override

### Protected Member Functions inherited from [Digikam::DSelector](#)

- virtual void **drawArrow** (QPainter \*painter, const QPoint &pos)
- void **mouseMoveEvent** (QMouseEvent \*e) override
- void **mousePressEvent** (QMouseEvent \*e) override
- void **mouseReleaseEvent** (QMouseEvent \*e) override
- void **paintEvent** (QPaintEvent \*) override
- void **wheelEvent** (QWheelEvent \*) override

### Properties

- int **colorValue**
- int **hue**
- int **saturation**

### Properties inherited from [Digikam::DSelector](#)

- Qt::ArrowType **arrowDirection**
- bool **indent**
- int **maxValue**
- int **minValue**
- int **value**



## Friends

- class **Private**

## 6.332.1 Member Function Documentation

### 6.332.1.1 chooserMode()

```
DColorChooserMode Digikam::DColorValueSelector::chooserMode ( ) const
```

Returns the current chooser mode.

#### Returns

The chooser mode (one of the DColorChooserMode constants)

### 6.332.1.2 colorValue()

```
int Digikam::DColorValueSelector::colorValue ( ) const
```

Returns the current color value.

#### Returns

The color value (0-255)

### 6.332.1.3 drawContents()

```
void Digikam::DColorValueSelector::drawContents (
    QPainter * painter ) [override], [protected], [virtual]
```

Reimplemented from [DSelector](#). The drawing is buffered in a pixmap here. As real drawing routine, [drawPalette\(\)](#) is used.

Reimplemented from [Digikam::DSelector](#).

### 6.332.1.4 drawPalette()

```
void Digikam::DColorValueSelector::drawPalette (
    QPixmap * pixmap ) [protected], [virtual]
```

Draws the contents of the widget on a pixmap, which is used for buffering.

### 6.332.1.5 hue()

```
int Digikam::DColorValueSelector::hue ( ) const
```

Returns the current hue value.

#### Returns

The hue value (0-359)

### 6.332.1.6 saturation()

```
int Digikam::DColorValueSelector::saturation ( ) const
```

Returns the current saturation value.

#### Returns

The saturation value (0-255)

### 6.332.1.7 setChooserMode()

```
void Digikam::DColorValueSelector::setChooserMode (
    DColorChooserMode chooserMode )
```

Sets the chooser mode. Doesn't automatically update the widget; you have to call `updateContents` manually.

#### Parameters

<i>chooserMode</i>	Sets the chooser mode (one of the <code>DColorChooserMode</code> constants)
--------------------	---

### 6.332.1.8 setColorValue()

```
void Digikam::DColorValueSelector::setColorValue (
    int colorValue )
```

Sets the color value. Doesn't automatically update the widget; you have to call `updateContents` manually.

#### Parameters

<i>colorValue</i>	Sets the color value (0-255)
-------------------	------------------------------

### 6.332.1.9 setHue()

```
void Digikam::DColorValueSelector::setHue (
    int hue )
```

Sets the hue value. Doesn't automatically update the widget; you have to call `updateContents` manually.

#### Parameters

<i>hue</i>	Sets the hue value (0-359)
------------	----------------------------

#### 6.332.1.10 `setSaturation()`

```
void Digikam::DColorValueSelector::setSaturation (
    int saturation )
```

Sets the saturation value. Doesn't automatically update the widget; you have to call `updateContents` manually.

#### Parameters

<i>saturation</i>	Sets the saturation value (0-255)
-------------------	-----------------------------------

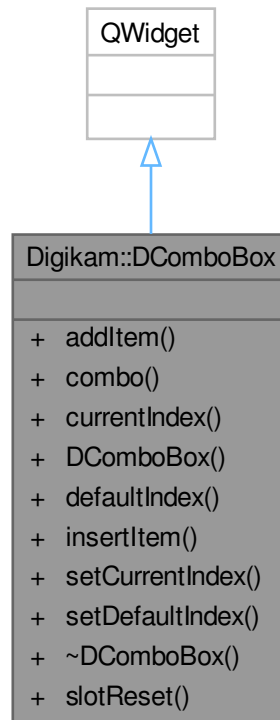
#### 6.332.1.11 `updateContents()`

```
void Digikam::DColorValueSelector::updateContents ( )
```

Updates the widget's contents.

## 6.333 Digikam::DComboBox Class Reference

Inheritance diagram for Digikam::DComboBox:



### Public Slots

- void **slotReset** ()

### Signals

- void **activated** (int)
- void **currentIndexChanged** (int)
- void **reset** ()

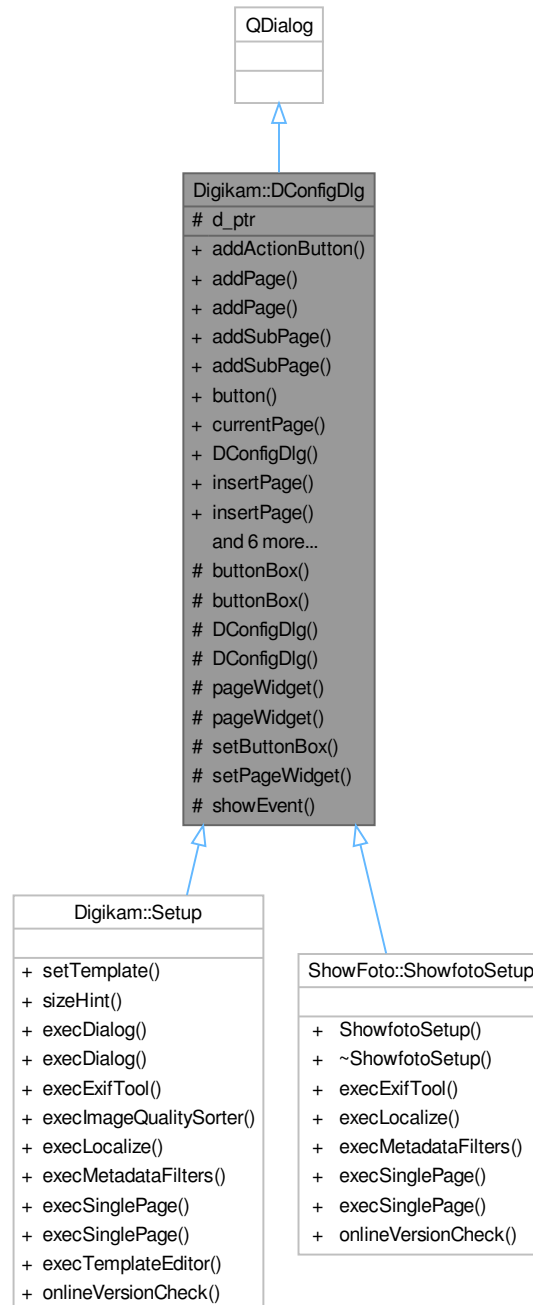
### Public Member Functions

- void **addItem** (const QString &t, const QVariant &data=QVariant())
- QComboBox \* **combo** () const
- int **currentIndex** () const
- **DComboBox** (QWidget \*const parent=nullptr)
- int **defaultIndex** () const
- void **insertItem** (int index, const QString &t, const QVariant &data=QVariant())
- void **setCurrentIndex** (int d)
- void **setDefaultIndex** (int d)

## 6.334 Digikam::DConfigDlg Class Reference

A dialog base class which can handle multiple pages.

Inheritance diagram for Digikam::DConfigDlg:



### Public Types

- enum `FaceType` {
  - Auto** = `DConfigDlgView::Auto` , **Plain** = `DConfigDlgView::Plain` , **List** = `DConfigDlgView::List` , **Tree** =

```
DConfigDlgView::Tree ,
Tabbed = DConfigDlgView::Tabbed }
```

## Signals

- void [currentPageChanged](#) (DConfigDlgWdgItem \*current, DConfigDlgWdgItem \*before)
- void [pageRemoved](#) (DConfigDlgWdgItem \*page)

## Public Member Functions

- void [addActionButton](#) (QAbstractButton \*const button)
- void [addPage](#) (DConfigDlgWdgItem \*const item)
- DConfigDlgWdgItem \* [addPage](#) (QWidget \*const widget, const QString &name)
- void [addSubPage](#) (DConfigDlgWdgItem \*const parent, DConfigDlgWdgItem \*const item)
- DConfigDlgWdgItem \* [addSubPage](#) (DConfigDlgWdgItem \*const parent, QWidget \*const widget, const QString &name)
- QPushButton \* [button](#) (QDialogButtonBox::StandardButton which) const
- DConfigDlgWdgItem \* [currentPage](#) () const
- DConfigDlg (QWidget \*const parent=nullptr, Qt::WindowFlags flags=Qt::WindowFlags())
- void [insertPage](#) (DConfigDlgWdgItem \*const before, DConfigDlgWdgItem \*const item)
- DConfigDlgWdgItem \* [insertPage](#) (DConfigDlgWdgItem \*const before, QWidget \*const widget, const QString &name)
- void [removePage](#) (DConfigDlgWdgItem \*const item)
- void [setConfigGroup](#) (const QString &group)
- void [setCurrentPage](#) (DConfigDlgWdgItem \*const item)
- void [setFaceType](#) (FaceType faceType)
- void [setStandardButtons](#) (QDialogButtonBox::StandardButtons buttons)
- ~DConfigDlg () override

## Protected Member Functions

- QDialogButtonBox \* [buttonBox](#) ()
- const QDialogButtonBox \* [buttonBox](#) () const
- **DConfigDlg** (DConfigDlgPrivate &dd, DConfigDlgWdg \*const widget, QWidget \*const parent, Qt::Window↔Flags flags=Qt::WindowFlags())
- DConfigDlg (DConfigDlgWdg \*const widget, QWidget \*const parent, Qt::WindowFlags flags=Qt::Window↔Flags())
- DConfigDlgWdg \* [pageWidget](#) ()
- const DConfigDlgWdg \* [pageWidget](#) () const
- void [setButtonBox](#) (QDialogButtonBox \*const box)
- void [setPageWidget](#) (DConfigDlgWdg \*const widget)
- void [showEvent](#) (QShowEvent \*) override

## Protected Attributes

- DConfigDlgPrivate \*const **d\_ptr** = nullptr

## 6.334.1 Detailed Description

This class provides a dialog base class which handles multiple pages and allows the user to switch between these pages in different ways.

Currently, `Auto`, `Plain`, `List`, `Tree` and `Tabbed` face types are available (

See also

[DConfigDlgView](#)).

## 6.334.2 Member Enumeration Documentation

### 6.334.2.1 FaceType

enum `Digikam::DConfigDlg::FaceType`

- `Auto` - A dialog with a face based on the structure of the available pages. If only a single page is added, the dialog behaves like in `Plain` mode, with multiple pages without sub pages it behaves like in `List` mode and like in `Tree` mode otherwise.
- `Plain` - A normal dialog.
- `List` - A dialog with an icon list on the left side and a representation of the contents on the right side.
- `Tree` - A dialog with a tree on the left side and a representation of the contents on the right side.
- `Tabbed` - A dialog with a tab bar above the representation of the contents.

## 6.334.3 Constructor & Destructor Documentation

### 6.334.3.1 DConfigDlg() [1/2]

```
Digikam::DConfigDlg::DConfigDlg (
    QWidget *const parent = nullptr,
    Qt::WindowFlags flags = Qt::WindowFlags() ) [explicit]
```

Creates a new page dialog.

### 6.334.3.2 ~DConfigDlg()

```
Digikam::DConfigDlg::~DConfigDlg ( ) [override]
```

Destroys the page dialog.

### 6.334.3.3 DConfigDlg() [2/2]

```
Digikam::DConfigDlg::DConfigDlg (
    DConfigDlgWdg *const widget,
    QWidget *const parent,
    Qt::WindowFlags flags = Qt::WindowFlags() ) [protected]
```

This constructor can be used by subclasses to provide a custom page widget.

## Parameters

<i>widget</i>	The <a href="#">DConfigDlgWdg</a> object will be reparented to this object, so you can create it without parent and you are not allowed to delete it.
<i>parent</i>	The widget parent instance
<i>flags</i>	The window flags

## 6.334.4 Member Function Documentation

### 6.334.4.1 addActionButton()

```
void Digikam::DConfigDlg::addActionButton (
    QAbstractButton *const button )
```

Set an action button.

### 6.334.4.2 addPage() [1/2]

```
void Digikam::DConfigDlg::addPage (
    DConfigDlgWdgItem *const item )
```

Adds a new top level page to the dialog.

## Parameters

<i>item</i>	The
-------------	-----

## See also

[DConfigDlgWdgItem](#) which describes the page.

### 6.334.4.3 addPage() [2/2]

```
DConfigDlgWdgItem * Digikam::DConfigDlg::addPage (
    QWidget *const widget,
    const QString & name )
```

Adds a new top level page to the dialog.

## Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.



**Returns**

The associated

**See also**

[DConfigDlgWdgItem](#).

**6.334.4.4 addSubPage() [1/2]**

```
void Digikam::DConfigDlg::addSubPage (
    DConfigDlgWdgItem *const parent,
    DConfigDlgWdgItem *const item )
```

Inserts a new sub page in the dialog.

**Parameters**

<i>parent</i>	The new page will be insert as child of this
---------------	--

**See also**

[DConfigDlgWdgItem](#).

**Parameters**

<i>item</i>	The
-------------	-----

**See also**

[DConfigDlgWdgItem](#) which describes the page.

**6.334.4.5 addSubPage() [2/2]**

```
DConfigDlgWdgItem * Digikam::DConfigDlg::addSubPage (
    DConfigDlgWdgItem *const parent,
    QWidget *const widget,
    const QString & name )
```

Inserts a new sub page in the dialog.

**Parameters**

<i>parent</i>	The new page will be insert as child of this
---------------	--

**See also**

[DConfigDlgWdgItem](#).

## Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

## Returns

The associated

## See also

[DConfigDlgWdgItem](#).

**6.334.4.6 button()**

```
QPushButton * Digikam::DConfigDlg::button (
    QDialogButtonBox::StandardButton which ) const
```

Returns the QPushButton corresponding to the standard button which, or 0 if the standard button doesn't exist in this dialog.

**6.334.4.7 buttonBox() [1/2]**

```
QDialogButtonBox * Digikam::DConfigDlg::buttonBox ( ) [protected]
```

Returns the button box of the dialog or 0 if no button box is set.

**6.334.4.8 buttonBox() [2/2]**

```
const QDialogButtonBox * Digikam::DConfigDlg::buttonBox ( ) const [protected]
```

Returns the button box of the dialog or 0 if no button box is set.

**6.334.4.9 currentPage()**

```
DConfigDlgWdgItem * Digikam::DConfigDlg::currentPage ( ) const
```

Returns the

## See also

[DConfigDlgWdgItem](#) for the current page or 0 if there is no current page.

**6.334.4.10 currentPageChanged**

```
void Digikam::DConfigDlg::currentPageChanged (
    DConfigDlgWdgItem * current,
    DConfigDlgWdgItem * before ) [signal]
```

This signal is emitted whenever the current page has changed.

## Parameters

<i>current</i>	The new current page or 0 if no current page is available.
----------------	--

**6.334.4.11 insertPage()** [1/2]

```
void Digikam::DConfigDlg::insertPage (
    DConfigDlgWdgItem *const before,
    DConfigDlgWdgItem *const item )
```

Inserts a new page in the dialog.

## Parameters

<i>before</i>	The new page will be insert before this
---------------	---

## See also

[DConfigDlgWdgItem](#) on the same level in hierarchy.

## Parameters

<i>item</i>	The
-------------	-----

## See also

[DConfigDlgWdgItem](#) which describes the page.

**6.334.4.12 insertPage()** [2/2]

```
DConfigDlgWdgItem * Digikam::DConfigDlg::insertPage (
    DConfigDlgWdgItem *const before,
    QWidget *const widget,
    const QString & name )
```

Inserts a new page in the dialog.

## Parameters

<i>before</i>	The new page will be insert before this
---------------	---

## See also

[DConfigDlgWdgItem](#) on the same level in hierarchy.

## Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

## Returns

The associated

## See also

[DConfigDlgWdgItem](#).

**6.334.4.13 pageRemoved**

```
void Digikam::DConfigDlg::pageRemoved (
    DConfigDlgWdgItem * page ) [signal]
```

This signal is emitted whenever a page has been removed.

## Parameters

<i>page</i>	The page which has been removed
-------------	---------------------------------

**6.334.4.14 pageWidget() [1/2]**

```
DConfigDlgWdg * Digikam::DConfigDlg::pageWidget ( ) [protected]
```

Returns the page widget of the dialog or 0 if no page widget is set.

**6.334.4.15 pageWidget() [2/2]**

```
const DConfigDlgWdg * Digikam::DConfigDlg::pageWidget ( ) const [protected]
```

Returns the page widget of the dialog or 0 if no page widget is set.

**6.334.4.16 removePage()**

```
void Digikam::DConfigDlg::removePage (
    DConfigDlgWdgItem *const item )
```

Removes the page associated with the given

## See also

[DConfigDlgWdgItem](#).

#### 6.334.4.17 setButtonBox()

```
void Digikam::DConfigDlg::setButtonBox (
    QDialogButtonBox *const box ) [protected]
```

Set the button box of the dialog

##### Note

the previous buttonBox will be deleted.

##### Parameters

<i>box</i>	The QDialogButtonBox object will be reparented to this object, so you can create it without parent and you are not allowed to delete it.
------------	--

#### 6.334.4.18 setConfigGroup()

```
void Digikam::DConfigDlg::setConfigGroup (
    const QString & group )
```

Sets the config group name for restore or save dialog window size.

#### 6.334.4.19 setCurrentPage()

```
void Digikam::DConfigDlg::setCurrentPage (
    DConfigDlgWdgItem *const item )
```

Sets the page which is associated with the given

##### See also

[DConfigDlgWdgItem](#) to be the current page and emits the [currentPageChanged\(\)](#) signal.

#### 6.334.4.20 setFaceType()

```
void Digikam::DConfigDlg::setFaceType (
    FaceType faceType )
```

Sets the face type of the dialog.

#### 6.334.4.21 setPageWidget()

```
void Digikam::DConfigDlg::setPageWidget (
    DConfigDlgWdg *const widget ) [protected]
```

Set the page widget of the dialog.

##### Note

the previous pageWidget will be deleted.

## Parameters

<i>widget</i>	The <a href="#">DConfigDlgWdg</a> object will be reparented to this object, so you can create it without parent and you are not allowed to delete it.
---------------	---

**6.334.4.22 setStandardButtons()**

```
void Digikam::DConfigDlg::setStandardButtons (
    QDialogButtonBox::StandardButtons buttons )
```

Sets the collection of standard buttons displayed by this dialog.

## 6.335 Digikam::DConfigDlgMngr Class Reference

Inheritance diagram for Digikam::DConfigDlgMngr:



### Public Slots

- void [updateSettings](#) ()
- void [updateWidgets](#) ()
- void [updateWidgetsDefault](#) ()

## Signals

- void [settingsChanged](#) ()
- void [settingsChanged](#) (QWidget \*widget)
- void [widgetModified](#) ()

## Public Member Functions

- void [addWidget](#) (QWidget \*const widget)
- [DConfigDlgMngr](#) (QWidget \*const parent, KConfigSkeleton \*const conf)
- bool [hasChanged](#) () const
- bool [isDefault](#) () const
- [~DConfigDlgMngr](#) () override

## Static Public Member Functions

- static QHash< QString, QByteArray > \* [changedMap](#) ()
- static QHash< QString, QByteArray > \* [propertyMap](#) ()

## Protected Member Functions

- QByteArray [getCustomProperty](#) (const QWidget \*widget) const
- QByteArray [getCustomPropertyChangedSignal](#) (const QWidget \*widget) const
- QByteArray [getUserProperty](#) (const QWidget \*widget) const
- QByteArray [getUserPropertyChangedSignal](#) (const QWidget \*widget) const
- void [init](#) (bool trackChanges)
- bool [parseChildren](#) (const QWidget \*widget, bool trackChanges)
- QVariant [property](#) (QWidget \*w) const
- void [setProperty](#) (QWidget \*w, const QVariant &v)
- void [setupWidget](#) (QWidget \*widget, KConfigSkeletonItem \*item)

## Static Protected Member Functions

- static void [initMaps](#) ()

## 6.335.1 Detailed Description

The [DConfigDlgMngr](#) class provides a means of automatically retrieving, saving and resetting basic settings. It also can emit signals when settings have been changed (settings were saved) or modified (the user changes a checkbox from on to off).

The object names of the widgets to be managed have to correspond to the names of the configuration entries in the KConfigSkeleton object plus an additional "kcfg\_" prefix. For example a widget with the object name "kcfg\_↔ MyOption" would be associated to the configuration entry "MyOption".

The widget classes of Qt are supported out of the box.

Custom widget classes are supported if they have a Q\_PROPERTY defined for the property representing the value edited by the widget. By default the property is used for which "USER true" is set. For using another property, see below.

## 6.335.2 Constructor & Destructor Documentation

### 6.335.2.1 DConfigDlgMngr()

```
Digikam::DConfigDlgMngr::DConfigDlgMngr (
    QWidget *const parent,
    KConfigSkeleton *const conf )
```

Constructor.



## Parameters

<i>parent</i>	Dialog widget to manage
<i>conf</i>	Object that contains settings

**6.335.2.2 ~DConfigDlgMngr()**

```
Digikam::DConfigDlgMngr::~DConfigDlgMngr ( ) [override]
```

Destructor.

**6.335.3 Member Function Documentation****6.335.3.1 addWidget()**

```
void Digikam::DConfigDlgMngr::addWidget (
    QWidget *const widget )
```

Add additional widgets to manage

## Parameters

<i>widget</i>	Additional widget to manage, including all its children
---------------	---

**6.335.3.2 changedMap()**

```
QHash< QString, QByteArray > * Digikam::DConfigDlgMngr::changedMap ( ) [static]
```

Retrieve the map between widgets class names and signals that are listened to detect changes in the configuration values.

**6.335.3.3 getCustomProperty()**

```
QByteArray Digikam::DConfigDlgMngr::getCustomProperty (
    const QWidget * widget ) const [protected]
```

Find the property to use for a widget by querying the "kcfg\_property" property of the widget. Like a widget can use a property other than the USER property.

**6.335.3.4 getCustomPropertyChangedSignal()**

```
QByteArray Digikam::DConfigDlgMngr::getCustomPropertyChangedSignal (
    const QWidget * widget ) const [protected]
```

Find the changed signal of the property to use for a widget by querying the "kcfg\_propertyNotify" property of the widget. Like a widget can use a property change signal other than the one for USER property, if there even is one.

**6.335.3.5 getUserProperty()**

```
QByteArray Digikam::DConfigDlgMngr::getUserProperty (
    const QWidget * widget ) const [protected]
```

Finds the USER property name using Qt's MetaProperty system, and caches it in the property map (the cache could be retrieved by [propertyMap\(\)](#) ).

**6.335.3.6 getUserPropertyChangedSignal()**

```
QByteArray Digikam::DConfigDlgMngr::getUserPropertyChangedSignal (
    const QWidget * widget ) const [protected]
```

Finds the changed signal of the USER property using Qt's MetaProperty system.

**6.335.3.7 hasChanged()**

```
bool Digikam::DConfigDlgMngr::hasChanged ( ) const
```

Returns whether the current state of the known widgets are different from the state in the config object.

**6.335.3.8 init()**

```
void Digikam::DConfigDlgMngr::init (
    bool trackChanges ) [protected]
```

**Parameters**

<i>trackChanges</i>	- If any changes by the widgets should be tracked set true. This causes the emitting the modified() signal when something changes.
---------------------	--

**6.335.3.9 initMaps()**

```
void Digikam::DConfigDlgMngr::initMaps ( ) [static], [protected]
```

Initializes the property maps

**6.335.3.10 isDefault()**

```
bool Digikam::DConfigDlgMngr::isDefault ( ) const
```

Returns whether the current state of the known widgets are the same as the default state in the config object.

**6.335.3.11 parseChildren()**

```
bool Digikam::DConfigDlgMngr::parseChildren (
    const QWidget * widget,
    bool trackChanges ) [protected]
```

Recursive function that finds all known children. Goes through the children of widget and if any are known and not being ignored, stores them in currentGroup. Also checks if the widget should be disabled because it is set immutable.

**Parameters**

<i>widget</i>	- Parent of the children to look at.
<i>trackChanges</i>	- If true then tracks any changes to the children of widget that are known.

**Returns**

bool - If a widget was set to something other than its default.

**6.335.3.12 property()**

```
QVariant Digikam::DConfigDlgMngr::property (
    QWidget * w ) const [protected]
```

Retrieve a property

**6.335.3.13 propertyMap()**

```
QHash< QString, QByteArray > * Digikam::DConfigDlgMngr::propertyMap ( ) [static]
```

Retrieve the map between widgets class names and the USER properties used for the configuration values.

**6.335.3.14 setProperty()**

```
void Digikam::DConfigDlgMngr::setProperty (
    QWidget * w,
    const QVariant & v ) [protected]
```

Set a property

**6.335.3.15 settingsChanged [1/2]**

```
void Digikam::DConfigDlgMngr::settingsChanged ( ) [signal]
```

One or more of the settings have been saved (such as when the user clicks on the Apply button). This is only emitted by [updateSettings\(\)](#) whenever one or more setting were changed and consequently saved.

**6.335.3.16 settingsChanged [2/2]**

```
void Digikam::DConfigDlgMngr::settingsChanged (
    QWidget * widget ) [signal]
```

One or more of the settings have been changed.

## Parameters

<i>widget</i>	- The widget group (pass in via <a href="#">addWidget()</a> ) that contains the one or more modified setting.
---------------	---

## See also

[settingsChanged\(\)](#)**6.335.3.17 setupWidget()**

```
void Digikam::DConfigDlgMngr::setupWidget (
    QWidget * widget,
    KConfigSkeletonItem * item ) [protected]
```

[Setup](#) secondary widget properties

**6.335.3.18 updateSettings**

```
void Digikam::DConfigDlgMngr::updateSettings ( ) [slot]
```

Traverse the specified widgets, saving the settings of all known widgets in the settings object.

Example use: User clicks Ok or Apply button in a configure dialog.

**6.335.3.19 updateWidgets**

```
void Digikam::DConfigDlgMngr::updateWidgets ( ) [slot]
```

Traverse the specified widgets, sets the state of all known widgets according to the state in the settings object.

Example use: Initialisation of dialog. Example use: User clicks Reset button in a configure dialog.

**6.335.3.20 updateWidgetsDefault**

```
void Digikam::DConfigDlgMngr::updateWidgetsDefault ( ) [slot]
```

Traverse the specified widgets, sets the state of all known widgets according to the default state in the settings object.

Example use: User clicks Defaults button in a configure dialog.

**6.335.3.21 widgetModified**

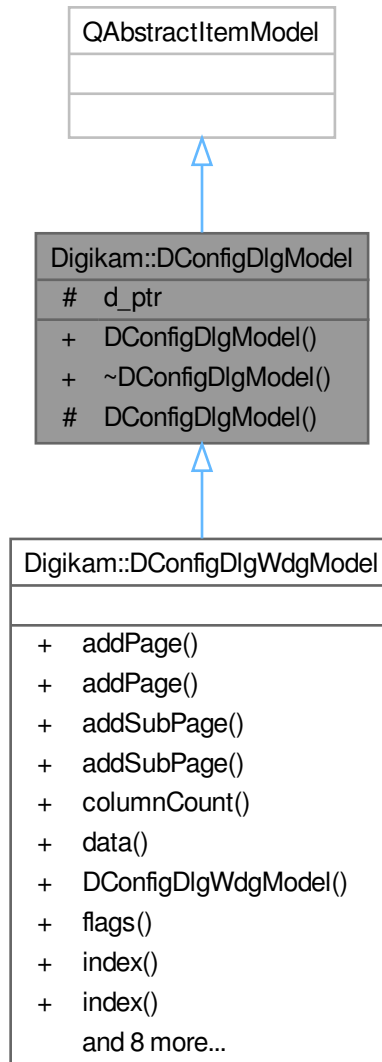
```
void Digikam::DConfigDlgMngr::widgetModified ( ) [signal]
```

If [retrieveSettings\(\)](#) was told to track changes then if any known setting was changed this signal will be emitted. Note that a settings can be modified several times and might go back to the original saved state. [hasChanged\(\)](#) will tell you if anything has actually changed from the saved values.

## 6.336 Digikam::DConfigDlgModel Class Reference

A base class for a model used by [DConfigDlgView](#).

Inheritance diagram for Digikam::DConfigDlgModel:



### Public Types

- enum `Role` { `HeaderRole` = `Qt::UserRole + 1` , `WidgetRole` }

### Public Member Functions

- `DConfigDlgModel` (`QObject *const parent=nullptr`)
- `~DConfigDlgModel` () override

## Protected Member Functions

- `DConfigDlgModel` (`DConfigDlgModelPrivate` &dd, `QObject *const parent`)

## Protected Attributes

- `DConfigDlgModelPrivate` \*const `d_ptr`

### 6.336.1 Detailed Description

This class is an abstract base class which must be used to implement custom models for `DConfigDlgView`. Additional to the standard `Qt::ItemDataRoles` it provides the two roles

- `HeaderRole`
- `WidgetRole`

which are used to return a header string for a page and a `QWidget` pointer to the page itself.

### 6.336.2 Member Enumeration Documentation

#### 6.336.2.1 Role

```
enum Digikam::DConfigDlgModel::Role
```

Additional roles that `DConfigDlgView` uses.

Enumerator

<code>HeaderRole</code>	A string to be rendered as page header.
<code>WidgetRole</code>	A pointer to the page widget. This is the widget that is shown when the item is selected.

### 6.336.3 Constructor & Destructor Documentation

#### 6.336.3.1 `DConfigDlgModel()`

```
Digikam::DConfigDlgModel::DConfigDlgModel (
    QObject *const parent = nullptr ) [explicit]
```

Constructs a page model with the given parent.

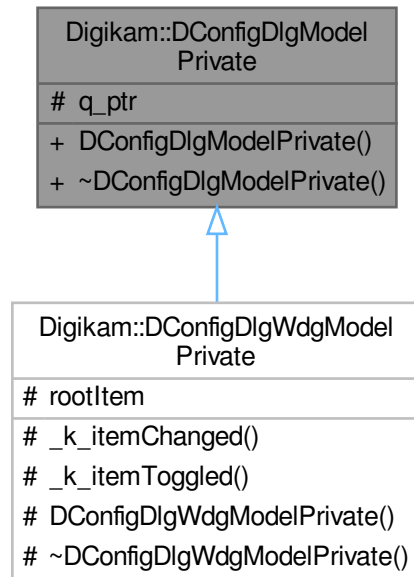
#### 6.336.3.2 `~DConfigDlgModel()`

```
Digikam::DConfigDlgModel::~DConfigDlgModel ( ) [override]
```

Destroys the page model.

## 6.337 Digikam::DConfigDlgModelPrivate Class Reference

Inheritance diagram for Digikam::DConfigDlgModelPrivate:

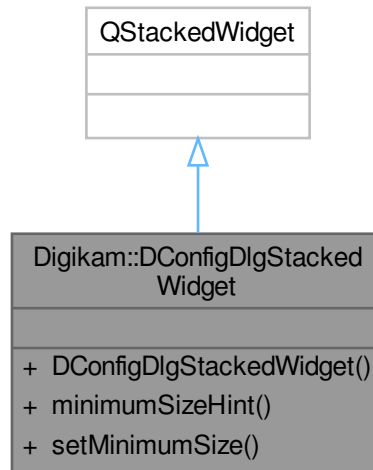


### Protected Attributes

- `DConfigDlgModel` \* `q_ptr` = nullptr

## 6.338 Digikam::DConfigDlgStackedWidget Class Reference

Inheritance diagram for Digikam::DConfigDlgStackedWidget:



### Public Member Functions

- **DConfigDlgStackedWidget** (`QWidget *const parent=nullptr`)
- `QSize` **minimumSizeHint** () const override
- void **setMinimumSize** (const `QSize &size`)



## 6.339 Digikam::DConfigDlgTitle Class Reference

Inheritance diagram for Digikam::DConfigDlgTitle:



### Classes

- class [Private](#)

### Public Types

- enum [ImageAlignment](#) { [ImageLeft](#) , [ImageRight](#) }
- enum [MessageType](#) { [PlainMessage](#) , [InfoMessage](#) , [WarningMessage](#) , [ErrorMessage](#) }

### Public Slots

- void [setAutoHideTimeout](#) (int msec)
- void [setComment](#) (const QString &comment, [MessageType](#) type=[PlainText](#))
- void [setPixmap](#) (const QIcon &icon, [ImageAlignment](#) alignment=[ImageRight](#))
- void [setPixmap](#) (const QPixmap &pixmap, [ImageAlignment](#) alignment=[ImageRight](#))
- void [setPixmap](#) (const QString &icon, [ImageAlignment](#) alignment=[ImageRight](#))
- void [setPixmap](#) ([MessageType](#) type, [ImageAlignment](#) alignment=[ImageRight](#))
- void [setText](#) (const QString &text, [MessageType](#) type)
- void [setText](#) (const QString &text, Qt::Alignment alignment=Qt::AlignLeft|Qt::AlignVCenter)

### Public Member Functions

- int [autoHideTimeout](#) () const
- QString [comment](#) () const
- [DConfigDlgTitle](#) (QWidget \*const parent=nullptr)
- QPixmap [pixmap](#) () const
- void [setBuddy](#) (QWidget \*const buddy)
- void [setWidget](#) (QWidget \*const widget)
- QString [text](#) () const

### Protected Member Functions

- void [changeEvent](#) (QEvent \*) override
- bool [eventFilter](#) (QObject \*, QEvent \*) override
- void [showEvent](#) (QShowEvent \*) override

### Properties

- int [autoHideTimeout](#)
- QString [comment](#)
- QPixmap [pixmap](#)
- QString [text](#)

## 6.339.1 Detailed Description

This class provides a widget often used for [DConfigDlg](#) titles.

[DConfigDlgTitle](#) uses the general application font at 1.4 times its size to style the text.

[DConfigDlgTitle](#) is very simple to use. You can either use its default text (and pixmap) properties or display your own widgets in the title widget.

## 6.339.2 Member Enumeration Documentation

### 6.339.2.1 ImageAlignment

```
enum Digikam::DConfigDlgTitle::ImageAlignment
```

Possible title pixmap alignments.

- ImageLeft: Display the pixmap left
- ImageRight: Display the pixmap right (default)

## Enumerator

ImageLeft	Display the pixmap on the left.
ImageRight	Display the pixmap on the right.

**6.339.2.2 MessageType**

```
enum Digikam::DConfigDlgTitle::MessageType
```

Comment message types

## Enumerator

PlainMessage	Normal comment.
InfoMessage	Information the user should be alerted to.
WarningMessage	A warning the user should be alerted to.
ErrorMessage	An error message.

**6.339.3 Constructor & Destructor Documentation****6.339.3.1 DConfigDlgTitle()**

```
Digikam::DConfigDlgTitle::DConfigDlgTitle (
    QWidget *const parent = nullptr ) [explicit]
```

Constructs a title widget with the given

## Parameters

<i>parent</i>	.
---------------	---

**6.339.4 Member Function Documentation****6.339.4.1 autoHideTimeout()**

```
int Digikam::DConfigDlgTitle::autoHideTimeout ( ) const
```

Get the current timeout value in milliseconds

## Returns

timeout value in msec

#### 6.339.4.2 comment()

```
QString Digikam::DConfigDlgTitle::comment ( ) const
```

##### Returns

the text displayed in the comment below the title, if any

##### See also

[setComment\(\)](#)

#### 6.339.4.3 pixmap()

```
QPixmap Digikam::DConfigDlgTitle::pixmap ( ) const
```

##### Returns

the pixmap displayed in the title

##### See also

[setPixmap\(\)](#)

#### 6.339.4.4 setAutoHideTimeout

```
void Digikam::DConfigDlgTitle::setAutoHideTimeout (
    int msec ) [slot]
```

Set the autohide timeout of the label Set value to 0 to disable autohide, which is the default.

##### Parameters

<i>msec</i>	timeout value in milliseconds
-------------	-------------------------------

#### 6.339.4.5 setBuddy()

```
void Digikam::DConfigDlgTitle::setBuddy (
    QWidget *const buddy )
```

Sets this label's buddy to buddy. When the user presses the shortcut key indicated by the label in this title widget, the keyboard focus is transferred to the label's buddy widget.

##### Parameters

<i>buddy</i>	the widget to activate when the shortcut key is activated
--------------	---

**6.339.4.6 setComment**

```
void Digikam::DConfigDlgTitle::setComment (
    const QString & comment,
    MessageType type = PlainMessage ) [slot]
```

**Parameters**

<i>comment</i>	Text displayed beneath the main title as a comment. It can either be plain text or rich text.
<i>type</i>	The sort of message it is.

**See also**

[MessageType](#)

comment()

**6.339.4.7 setPixmap [1/4]**

```
void Digikam::DConfigDlgTitle::setPixmap (
    const QIcon & icon,
    ImageAlignment alignment = ImageRight ) [slot]
```

**Parameters**

<i>icon</i>	The pixmap to display in the header. The pixmap is by default right, but
<i>alignment</i>	can be used to display it also left.

**See also**

pixmap()

**6.339.4.8 setPixmap [2/4]**

```
void Digikam::DConfigDlgTitle::setPixmap (
    const QPixmap & pixmap,
    ImageAlignment alignment = ImageRight ) [slot]
```

**Parameters**

<i>pixmap</i>	Pixmap displayed in the header. The pixmap is by default right, but
<i>alignment</i>	can be used to display it also left.

**See also**

pixmap()

**6.339.4.9 setPixmap [3/4]**

```
void Digikam::DConfigDlgTitle::setPixmap (
    const QString & icon,
    ImageAlignment alignment = ImageRight ) [slot]
```

**Parameters**

<i>icon</i>	name of the icon to display in the header. The pixmap is by default right, but
<i>alignment</i>	can be used to display it also left.

**See also**

[pixmap\(\)](#)

**6.339.4.10 setPixmap [4/4]**

```
void Digikam::DConfigDlgTitle::setPixmap (
    MessageType type,
    ImageAlignment alignment = ImageRight ) [slot]
```

**Parameters**

<i>type</i>	The message type to display as pixmap in the header. The message is by default right, but
<i>alignment</i>	can be used to display it also left.

**See also**

[pixmap\(\)](#)

**6.339.4.11 setText [1/2]**

```
void Digikam::DConfigDlgTitle::setText (
    const QString & text,
    MessageType type ) [slot]
```

**Parameters**

<i>text</i>	Text displayed on the label. It can either be plain text or rich text. If it is plain text, the text is displayed as a bold title text.
<i>type</i>	The sort of message it is; will also set the icon accordingly

**See also**

[MessageType](#)

[text\(\)](#)

**6.339.4.12 setText** [2/2]

```
void Digikam::DConfigDlgTitle::setText (
    const QString & text,
    Qt::Alignment alignment = Qt::AlignLeft | Qt::AlignVCenter ) [slot]
```

**Parameters**

<i>text</i>	Text displayed on the label. It can either be plain text or rich text. If it is plain text, the text is displayed as a bold title text.
<i>alignment</i>	Alignment of the text. Default is left and vertical centered.

**See also**

[text\(\)](#)

**6.339.4.13 setWidget()**

```
void Digikam::DConfigDlgTitle::setWidget (
    QWidget *const widget )
```

**Parameters**

<i>widget</i>	the widget displayed on the title widget.
---------------	---

**6.339.4.14 text()**

```
QString Digikam::DConfigDlgTitle::text ( ) const
```

**Returns**

the text displayed in the title

**See also**

[setText\(\)](#)

**6.340 Digikam::DConfigDlgTitle::Private Class Reference****Public Member Functions**

- void **\_k\_timeoutFinished** ()
- QString **commentStyleSheet** () const
- QString **iconTypeToIconName** (DConfigDlgTitle::MessageType type)  
*Get the icon name from the icon type.*
- **Private** (DConfigDlgTitle \*const parent)
- QString **textStyleSheet** () const

## Public Attributes

- int **autoHideTimeout** = 0
- QLabel \* **commentLabel** = nullptr
- QGridLayout \* **headerLayout** = nullptr
- QLabel \* **imageLabel** = nullptr
- [MessageType](#) **messageType** = [InfoMessage](#)
- [DConfigDlgTitle](#) \* **q** = nullptr
- QLabel \* **textLabel** = nullptr

## 6.340.1 Member Function Documentation

### 6.340.1.1 iconTypeToIconName()

```
QString Digikam::DConfigDlgTitle::Private::iconTypeToIconName (  
    DConfigDlgTitle::MessageType type ) [inline]
```

#### Parameters

<i>type</i>	icon type from the enum
-------------	-------------------------

#### Returns

named icon as QString

## 6.341 Digikam::DConfigDlgView Class Reference

A base class which can handle multiple pages.



Inheritance diagram for Digikam::DConfigDlgView:



### Public Types

- enum `FaceType` {  
**Auto** , **Plain** , **List** , **Tree** ,  
**Tabbed** }

### Signals

- void `currentPageChanged` (const `QModelIndex` &current, const `QModelIndex` &previous)

## Public Member Functions

- QModelIndex [currentPage](#) () const
- [DConfigDlgView](#) (QWidget \*const parent=nullptr)
- [FaceType](#) [faceType](#) () const
- QAbstractItemDelegate \* [itemDelegate](#) () const
- QAbstractItemModel \* [model](#) () const
- void [setCurrentPage](#) (const QModelIndex &index)
- void [setDefaultWidget](#) (QWidget \*widget)
- void [setFaceType](#) ([FaceType](#) faceType)
- void [setItemDelegate](#) (QAbstractItemDelegate \*delegate)
- void [setModel](#) (QAbstractItemModel \*model)
- [~DConfigDlgView](#) () override

## Protected Member Functions

- virtual QAbstractItemView \* [createView](#) ()
- **DConfigDlgView** ([DConfigDlgViewPrivate](#) &dd, QWidget \*const parent)
- virtual bool [showPageHeader](#) () const
- virtual Qt::Alignment [viewPosition](#) () const

## Protected Attributes

- [DConfigDlgViewPrivate](#) \*const [d\\_ptr](#)

## Properties

- [FaceType](#) [faceType](#)

### 6.341.1 Detailed Description

This class provides a widget base class which handles multiple pages and allows the user to switch between these pages in different ways.

Currently, `Auto`, `Plain`, `List`, `Tree` and `Tabbed` face types are available.

See also

[DConfigDlgWdg](#)

### 6.341.2 Member Enumeration Documentation

#### 6.341.2.1 FaceType

```
enum Digikam::DConfigDlgView::FaceType
```

This enum is used to decide which type of navigation view shall be used in the page view.

- `Auto` - Depending on the number of pages in the model, the `Plain` (one page), the `List` (several pages) or the `Tree` face (nested pages) will be used. This is the default face type.
- `Plain` - No navigation view will be visible and only the first page of the model will be shown.
- `List` - An icon list is used as navigation view.
- `Tree` - A tree list is used as navigation view.
- `Tabbed` - A tab widget is used as navigation view.

## 6.341.3 Constructor & Destructor Documentation

### 6.341.3.1 DConfigDlgView()

```
Digikam::DConfigDlgView::DConfigDlgView (
    QWidget *const parent = nullptr ) [explicit]
```

Creates a page view with given parent.

### 6.341.3.2 ~DConfigDlgView()

```
Digikam::DConfigDlgView::~DConfigDlgView ( ) [override]
```

Destroys the page view.

## 6.341.4 Member Function Documentation

### 6.341.4.1 createView()

```
QAbstractItemView * Digikam::DConfigDlgView::createView ( ) [protected], [virtual]
```

Returns the navigation view, depending on the current face type.

This method can be reimplemented to provide custom navigation views.

### 6.341.4.2 currentPage()

```
QModelIndex Digikam::DConfigDlgView::currentPage ( ) const
```

Returns the index for the current page or an invalid index if no current page exists.

### 6.341.4.3 currentPageChanged

```
void Digikam::DConfigDlgView::currentPageChanged (
    const QModelIndex & current,
    const QModelIndex & previous ) [signal]
```

This signal is emitted whenever the current page changes. The previous page index is replaced by the current index.

### 6.341.4.4 faceType()

```
DConfigDlgView::FaceType Digikam::DConfigDlgView::faceType ( ) const
```

Returns the face type of the page view.

#### 6.341.4.5 itemDelegate()

```
QAbstractItemDelegate * Digikam::DConfigDlgView::itemDelegate ( ) const
```

Returns the item delegate of the page view.

#### 6.341.4.6 model()

```
QAbstractItemModel * Digikam::DConfigDlgView::model ( ) const
```

Returns the model of the page view.

#### 6.341.4.7 setCurrentPage()

```
void Digikam::DConfigDlgView::setCurrentPage (
    const QModelIndex & index )
```

Sets the page with

##### Parameters

<i>index</i>	to be the current page and emits the
--------------	--------------------------------------

##### See also

[currentPageChanged](#) signal.

#### 6.341.4.8 setDefaultWidget()

```
void Digikam::DConfigDlgView::setDefaultWidget (
    QWidget * widget )
```

Sets the widget which will be shown when a page is selected that has no own widget set.

#### 6.341.4.9 setFaceType()

```
void Digikam::DConfigDlgView::setFaceType (
    FaceType faceType )
```

Sets the face type of the page view.

#### 6.341.4.10 setItemDelegate()

```
void Digikam::DConfigDlgView::setItemDelegate (
    QAbstractItemDelegate * delegate )
```

Sets the item

## Parameters

<i>delegate</i>	which can be used customize the page view.
-----------------	--

**6.341.4.11 setModel()**

```
void Digikam::DConfigDlgView::setModel (
    QAbstractItemModel * model )
```

Sets the `model` of the page view.

The model has to provide data for the roles defined in [DConfigDlgModel::Role](#).

**6.341.4.12 showPageHeader()**

```
bool Digikam::DConfigDlgView::showPageHeader ( ) const [protected], [virtual]
```

Returns whether the page header should be visible.

This method can be reimplemented for adapting custom views.

**6.341.4.13 viewPosition()**

```
Qt::Alignment Digikam::DConfigDlgView::viewPosition ( ) const [protected], [virtual]
```

Returns the position where the navigation view should be located according to the page stack.

This method can be reimplemented for adapting custom views.

## 6.342 Digikam::DConfigDlgViewPrivate Class Reference

Inheritance diagram for Digikam::DConfigDlgViewPrivate:



### Protected Member Functions

- void `_k_dataChanged` (const QModelIndex &, const QModelIndex &)
- void `_k_modelChanged` ()
- void `_k_pageSelected` (const QItemSelection &, const QItemSelection &)
- void `_k_rebuildGui` ()
- void `cleanupPages` ()
- QList< QWidget \* > `collectPages` (const QModelIndex &parent=QModelIndex())
- `DConfigDlgViewPrivate` (`DConfigDlgView` \*const)
- `DConfigDlgView::FaceType` `detectAutoFace` () const
- void `updateSelection` ()
- void `updateTitleWidget` (const QModelIndex &index)

### Protected Attributes

- QWidget \* **defaultWidget** = nullptr
- DConfigDlgView::FaceType **faceType** = DConfigDlgView::Auto
- QGridLayout \* **layout** = nullptr
- QAbstractItemModel \* **model** = nullptr
- DConfigDlgView \* **q\_ptr** = nullptr
- DConfigDlgStackedWidget \* **stack** = nullptr
- DConfigDlgTitle \* **titleLabel** = nullptr
- QAbstractItemView \* **view** = nullptr

## 6.342.1 Member Function Documentation

### 6.342.1.1 `_k_dataChanged()`

```
void Digikam::DConfigDlgViewPrivate::_k_dataChanged (
    const QModelIndex & ,
    const QModelIndex & ) [protected]
```

When data has changed we update the header and icon for the currently selected page.

### 6.342.1.2 `_k_modelChanged()`

```
void Digikam::DConfigDlgViewPrivate::_k_modelChanged ( ) [protected]
```

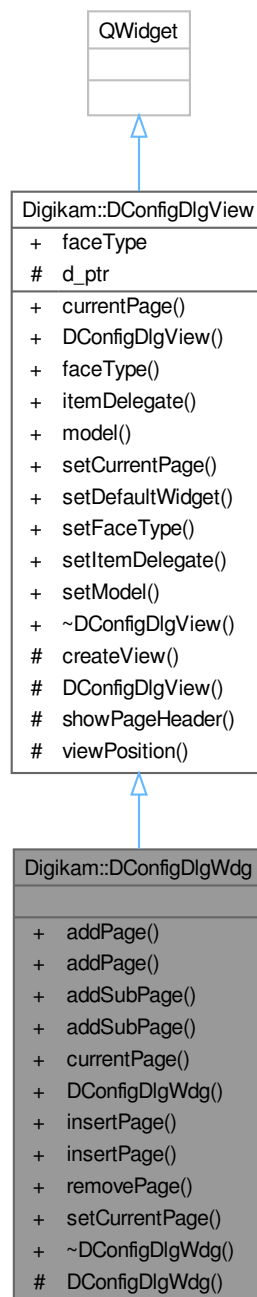
If the face type is Auto, we rebuild the GUI whenever the layout of the model changes.

Set the stack to the minimum size of the largest widget.

## 6.343 Digikam::DConfigDlgWdg Class Reference

Page widget with many layouts (faces).

Inheritance diagram for Digikam::DConfigDlgWdg:



## Signals

- void [currentPageChanged](#) ([DConfigDlgWdgItem](#) \*current, [DConfigDlgWdgItem](#) \*before)
- void [pageRemoved](#) ([DConfigDlgWdgItem](#) \*page)
- void [pageToggled](#) ([DConfigDlgWdgItem](#) \*page, bool checked)



## Signals inherited from [Digikam::DConfigDlgView](#)

- void [currentPageChanged](#) (const QModelIndex &current, const QModelIndex &previous)

## Public Member Functions

- void [addPage](#) ([DConfigDlgWdgItem](#) \*item)
- [DConfigDlgWdgItem](#) \* [addPage](#) (QWidget \*widget, const QString &name)
- void [addSubPage](#) ([DConfigDlgWdgItem](#) \*parent, [DConfigDlgWdgItem](#) \*item)
- [DConfigDlgWdgItem](#) \* [addSubPage](#) ([DConfigDlgWdgItem](#) \*parent, QWidget \*widget, const QString &name)
- [DConfigDlgWdgItem](#) \* [currentPage](#) () const
- [DConfigDlgWdg](#) (QWidget \*const parent=nullptr)
- void [insertPage](#) ([DConfigDlgWdgItem](#) \*before, [DConfigDlgWdgItem](#) \*item)
- [DConfigDlgWdgItem](#) \* [insertPage](#) ([DConfigDlgWdgItem](#) \*before, QWidget \*widget, const QString &name)
- void [removePage](#) ([DConfigDlgWdgItem](#) \*item)
- void [setCurrentPage](#) ([DConfigDlgWdgItem](#) \*item)
- [~DConfigDlgWdg](#) () override=default

## Public Member Functions inherited from [Digikam::DConfigDlgView](#)

- QModelIndex [currentPage](#) () const
- [DConfigDlgView](#) (QWidget \*const parent=nullptr)
- [FaceType](#) [faceType](#) () const
- QAbstractItemDelegate \* [itemDelegate](#) () const
- QAbstractItemModel \* [model](#) () const
- void [setCurrentPage](#) (const QModelIndex &index)
- void [setDefaultWidget](#) (QWidget \*widget)
- void [setFaceType](#) ([FaceType](#) faceType)
- void [setItemDelegate](#) (QAbstractItemDelegate \*delegate)
- void [setModel](#) (QAbstractItemModel \*model)
- [~DConfigDlgView](#) () override

## Protected Member Functions

- [DConfigDlgWdg](#) ([DConfigDlgWdgPrivate](#) &dd, QWidget \*const parent)

## Protected Member Functions inherited from [Digikam::DConfigDlgView](#)

- virtual QAbstractItemView \* [createView](#) ()
- [DConfigDlgView](#) ([DConfigDlgViewPrivate](#) &dd, QWidget \*const parent)
- virtual bool [showPageHeader](#) () const
- virtual Qt::Alignment [viewPosition](#) () const

## Additional Inherited Members

## Public Types inherited from [Digikam::DConfigDlgView](#)

- enum [FaceType](#) {  
**Auto** , **Plain** , **List** , **Tree** ,  
**Tabbed** }

## Protected Attributes inherited from [Digikam::DConfigDlgView](#)

- [DConfigDlgViewPrivate](#) \*const `d_ptr`

## Properties inherited from [Digikam::DConfigDlgView](#)

- [FaceType](#) `faceType`

### 6.343.1 Detailed Description

See also

[DConfigDlgView](#) with hierarchical page [model](#).

### 6.343.2 Constructor & Destructor Documentation

#### 6.343.2.1 DConfigDlgWdg()

```
Digikam::DConfigDlgWdg::DConfigDlgWdg (
    QWidget *const parent = nullptr ) [explicit]
```

Creates a new page widget.

Parameters

<i>parent</i>	The parent widget.
---------------	--------------------

#### 6.343.2.2 ~DConfigDlgWdg()

```
Digikam::DConfigDlgWdg::~DConfigDlgWdg ( ) [override], [default]
```

Destroys the page widget.

### 6.343.3 Member Function Documentation

#### 6.343.3.1 addPage() [1/2]

```
void Digikam::DConfigDlgWdg::addPage (
    DConfigDlgWdgItem * item )
```

Adds a new top level page to the widget.

Parameters

<i>item</i>	The
-------------	-----

See also

[DConfigDlgWdgItem](#) which describes the page.

### 6.343.3.2 addPage() [2/2]

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdg::addPage (
    QWidget * widget,
    const QString & name )
```

Adds a new top level page to the widget.

Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

Returns

The associated

See also

[DConfigDlgWdgItem](#).

### 6.343.3.3 addSubPage() [1/2]

```
void Digikam::DConfigDlgWdg::addSubPage (
    DConfigDlgWdgItem * parent,
    DConfigDlgWdgItem * item )
```

Inserts a new sub page in the widget.

Parameters

<i>parent</i>	The new page will be insert as child of this
---------------	--

See also

[DConfigDlgWdgItem](#).

Parameters

<i>item</i>	The
-------------	-----

See also

[DConfigDlgWdgItem](#) which describes the page.

#### 6.343.3.4 addSubPage() [2/2]

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdg::addSubPage (
    DConfigDlgWdgItem * parent,
    QWidget * widget,
    const QString & name )
```

Inserts a new sub page in the widget.

Parameters

<i>parent</i>	The new page will be insert as child of this
---------------	--

See also

[DConfigDlgWdgItem](#).

Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

Returns

The associated

See also

[DConfigDlgWdgItem](#).

#### 6.343.3.5 currentPage()

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdg::currentPage ( ) const
```

Returns the

See also

[DConfigDlgWdgItem](#) for the current page or 0 if there is no current page.

#### 6.343.3.6 currentPageChanged

```
void Digikam::DConfigDlgWdg::currentPageChanged (
    DConfigDlgWdgItem * current,
    DConfigDlgWdgItem * before ) [signal]
```

This signal is emitted whenever the current page has changed.

## Parameters

<i>current</i>	The new current page or 0 if no current page is available.
----------------	--

**6.343.3.7 insertPage()** [1/2]

```
void Digikam::DConfigDlgWdg::insertPage (  
    DConfigDlgWdgItem * before,  
    DConfigDlgWdgItem * item )
```

Inserts a new page in the widget.

## Parameters

<i>before</i>	The new page will be insert before this
---------------	---

## See also

[DConfigDlgWdgItem](#) on the same level in hierarchy.

## Parameters

<i>item</i>	The
-------------	-----

## See also

[DConfigDlgWdgItem](#) which describes the page.

**6.343.3.8 insertPage()** [2/2]

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdg::insertPage (  
    DConfigDlgWdgItem * before,  
    QWidget * widget,  
    const QString & name )
```

Inserts a new page in the widget.

## Parameters

<i>before</i>	The new page will be insert before this
---------------	---

## See also

[DConfigDlgWdgItem](#) on the same level in hierarchy.

## Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

## Returns

The associated

## See also

[DConfigDlgWdgItem](#).

**6.343.3.9 pageRemoved**

```
void Digikam::DConfigDlgWdg::pageRemoved (
    DConfigDlgWdgItem * page ) [signal]
```

This signal is emitted when a page is removed.

## Parameters

<i>page</i>	The page which is removed
-------------	---------------------------

**6.343.3.10 pageToggled**

```
void Digikam::DConfigDlgWdg::pageToggled (
    DConfigDlgWdgItem * page,
    bool checked ) [signal]
```

This signal is emitted whenever a checkable page changes its state.

## Parameters

<i>checked</i>	is true when the
<i>page</i>	is checked, or false if the
<i>page</i>	is unchecked.

**6.343.3.11 removePage()**

```
void Digikam::DConfigDlgWdg::removePage (
    DConfigDlgWdgItem * item )
```

Removes the page associated with the given

## See also

[DConfigDlgWdgItem](#).

### 6.343.3.12 setCurrentPage()

```
void Digikam::DConfigDlgWdg::setCurrentPage (
    DConfigDlgWdgItem * item )
```

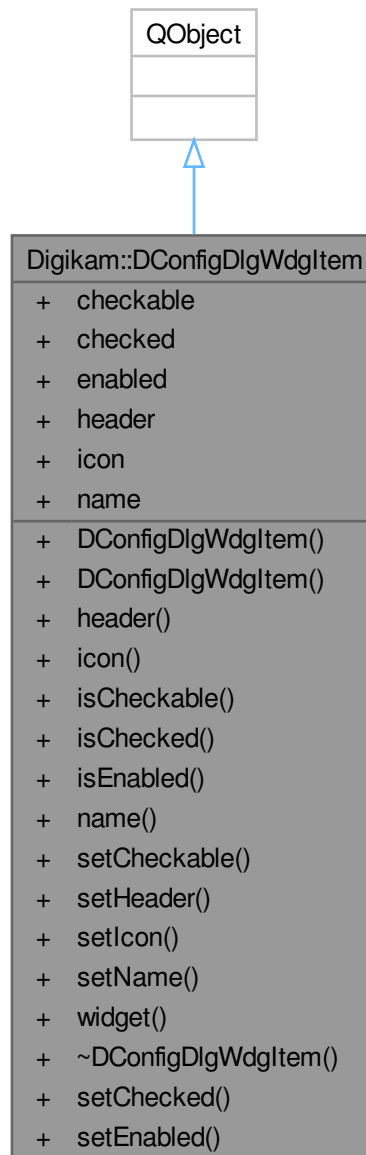
Sets the page which is associated with the given

See also

[DConfigDlgWdgItem](#) to be the current page and emits the [currentPageChanged\(\)](#) signal.

## 6.344 Digikam::DConfigDlgWdgItem Class Reference

Inheritance diagram for Digikam::DConfigDlgWdgItem:



## Public Slots

- void [setChecked](#) (bool checked)
- void [setEnabled](#) (bool)

## Signals

- void [changed](#) ()
- void [toggled](#) (bool checked)

## Public Member Functions

- [DConfigDlgWdgItem](#) (QWidget \*const [widget](#))
- [DConfigDlgWdgItem](#) (QWidget \*const [widget](#), const QString &name)
- QString [header](#) () const
- QIcon [icon](#) () const
- bool [isCheckable](#) () const
- bool [isChecked](#) () const
- bool [isEnabled](#) () const
- QString [name](#) () const
- void [setCheckable](#) (bool checkable)
- void [setHeader](#) (const QString &header)
- void [setIcon](#) (const QIcon &icon)
- void [setName](#) (const QString &name)
- QWidget \* [widget](#) () const
- [~DConfigDlgWdgItem](#) () override

## Properties

- bool **checkable**
- bool **checked**
- bool [enabled](#)
- QString **header**
- QIcon **icon**
- QString **name**

### 6.344.1 Detailed Description

[DConfigDlgWdgItem](#) is used by [DConfigDlgWdg](#) and represents a page.

### 6.344.2 Constructor & Destructor Documentation

#### 6.344.2.1 DConfigDlgWdgItem() [1/2]

```
Digikam::DConfigDlgWdgItem::DConfigDlgWdgItem (
    QWidget *const widget ) [explicit]
```

Creates a new page widget item.



## Parameters

<i>widget</i>	The widget that is shown as page in the <a href="#">DConfigDlgWdg</a> .
---------------	---

Hide the widget, otherwise when the widget has this [DConfigDlgView](#) as parent the widget is shown outside the `QStackedWidget` if the page was not selected ( and reparented ) yet.

**6.344.2.2 DConfigDlgWdgItem()** [2/2]

```
Digikam::DConfigDlgWdgItem::DConfigDlgWdgItem (
    QWidget *const widget,
    const QString & name )
```

Creates a new page widget item.

## Parameters

<i>widget</i>	The widget that is shown as page in the <a href="#">DConfigDlgWdg</a> .
<i>name</i>	The localized string that is show in the navigation view of the <a href="#">DConfigDlgWdg</a> .

Hide the widget, otherwise when the widget has this [DConfigDlgView](#) as parent the widget is shown outside the `QStackedWidget` if the page was not selected ( and reparented ) yet.

**6.344.2.3 ~DConfigDlgWdgItem()**

```
Digikam::DConfigDlgWdgItem::~DConfigDlgWdgItem ( ) [override]
```

Destroys the page widget item.

**6.344.3 Member Function Documentation****6.344.3.1 changed**

```
void Digikam::DConfigDlgWdgItem::changed ( ) [signal]
```

This signal is emitted whenever the icon or header is changed.

**6.344.3.2 header()**

```
QString Digikam::DConfigDlgWdgItem::header ( ) const
```

Returns the header of the page widget item.

**6.344.3.3 icon()**

```
QIcon Digikam::DConfigDlgWdgItem::icon ( ) const
```

Returns the icon of the page widget item.

#### 6.344.3.4 isCheckable()

```
bool Digikam::DConfigDlgWdgItem::isCheckable ( ) const
```

Returns whether the page widget item is checkable.

#### 6.344.3.5 isChecked()

```
bool Digikam::DConfigDlgWdgItem::isChecked ( ) const
```

Returns whether the page widget item is checked.

#### 6.344.3.6 isEnabled()

```
bool Digikam::DConfigDlgWdgItem::isEnabled ( ) const
```

Returns whether the page widget item is enabled.

#### 6.344.3.7 name()

```
QString Digikam::DConfigDlgWdgItem::name ( ) const
```

Returns the name of the page widget item.

#### 6.344.3.8 setCheckable()

```
void Digikam::DConfigDlgWdgItem::setCheckable (
    bool checkable )
```

Sets whether the page widget item is checkable in the view.

##### Parameters

<i>checkable</i>	True if the page widget is checkable, otherwise false.
------------------	--

#### 6.344.3.9 setChecked

```
void Digikam::DConfigDlgWdgItem::setChecked (
    bool checked ) [slot]
```

Sets whether the page widget item is checked.

#### 6.344.3.10 setEnabled

```
void Digikam::DConfigDlgWdgItem::setEnabled (
    bool enabled ) [slot]
```

Sets whether the page widget item is enabled.

### 6.344.3.11 setHeader()

```
void Digikam::DConfigDlgWdgItem::setHeader (
    const QString & header )
```

Sets the header of the page widget item.

If `setHeader(QString())` is used, what is the default if the header does not got set explicit, then the defined `name()` will also be used for the header. If `setHeader("")` is used, the header will be hidden even if the [DConfigDlgView::FaceType](#) is something else then `Tabbed`.

#### Parameters

<i>header</i>	Header of the page widget item.
---------------	---------------------------------

### 6.344.3.12 setIcon()

```
void Digikam::DConfigDlgWdgItem::setIcon (
    const QIcon & icon )
```

Sets the icon of the page widget item.

#### Parameters

<i>icon</i>	Icon of the page widget item.
-------------	-------------------------------

### 6.344.3.13 setName()

```
void Digikam::DConfigDlgWdgItem::setName (
    const QString & name )
```

Sets the name of the item as shown in the navigation view of the page widget.

### 6.344.3.14 toggled

```
void Digikam::DConfigDlgWdgItem::toggled (
    bool checked ) [signal]
```

This signal is emitted whenever the user checks or unchecks the item of

#### See also

[setChecked\(\)](#) is called.

### 6.344.3.15 widget()

```
QWidget * Digikam::DConfigDlgWdgItem::widget ( ) const
```

Returns the widget of the page widget item.

## 6.344.4 Property Documentation

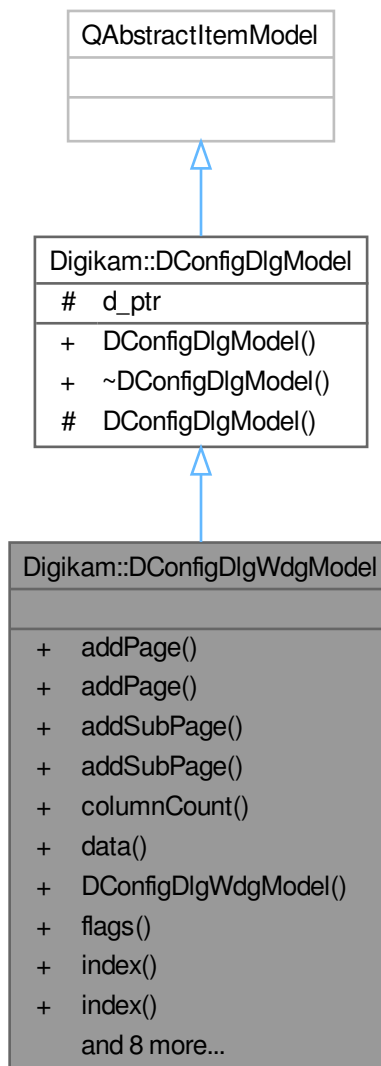
### 6.344.4.1 enabled

```
bool Digikam::DConfigDlgWdgItem::enabled [read], [write]
```

This property holds whether the item is enabled. It dis-/enables both the widget and the item in the list-/treeview.

## 6.345 Digikam::DConfigDlgWdgModel Class Reference

Inheritance diagram for Digikam::DConfigDlgWdgModel:



## Signals

- void [toggled](#) ([DConfigDlgWdgItem](#) \*page, bool checked)

## Public Member Functions

- void [addPage](#) ([DConfigDlgWdgItem](#) \*item)
- [DConfigDlgWdgItem](#) \* [addPage](#) (QWidget \*widget, const QString &name)
- void [addSubPage](#) ([DConfigDlgWdgItem](#) \*parent, [DConfigDlgWdgItem](#) \*item)
- [DConfigDlgWdgItem](#) \* [addSubPage](#) ([DConfigDlgWdgItem](#) \*parent, QWidget \*widget, const QString &name)
- int [columnCount](#) (const QModelIndex &parent=QModelIndex()) const override
- QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const override
- [DConfigDlgWdgModel](#) (QObject \*const parent=nullptr)
- Qt::ItemFlags [flags](#) (const QModelIndex &index) const override
- QModelIndex [index](#) (const [DConfigDlgWdgItem](#) \*item) const
- QModelIndex [index](#) (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- void [insertPage](#) ([DConfigDlgWdgItem](#) \*before, [DConfigDlgWdgItem](#) \*item)
- [DConfigDlgWdgItem](#) \* [insertPage](#) ([DConfigDlgWdgItem](#) \*before, QWidget \*widget, const QString &name)
- [DConfigDlgWdgItem](#) \* [item](#) (const QModelIndex &index) const
- QModelIndex [parent](#) (const QModelIndex &index) const override
- void [removePage](#) ([DConfigDlgWdgItem](#) \*item)
- int [rowCount](#) (const QModelIndex &parent=QModelIndex()) const override
- bool [setData](#) (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override
- [~DConfigDlgWdgModel](#) () override=default

## Public Member Functions inherited from [Digikam::DConfigDlgModel](#)

- [DConfigDlgModel](#) (QObject \*const parent=nullptr)
- [~DConfigDlgModel](#) () override

## Additional Inherited Members

## Public Types inherited from [Digikam::DConfigDlgModel](#)

- enum [Role](#) { [HeaderRole](#) = Qt::UserRole + 1 , [WidgetRole](#) }

## Protected Member Functions inherited from [Digikam::DConfigDlgModel](#)

- [DConfigDlgModel](#) ([DConfigDlgModelPrivate](#) &dd, QObject \*const parent)

## Protected Attributes inherited from [Digikam::DConfigDlgModel](#)

- [DConfigDlgModelPrivate](#) \*const [d\\_ptr](#)

## 6.345.1 Detailed Description

This page model is used by

See also

[DConfigDlgWdg](#) to provide a hierarchical layout of pages.

## 6.345.2 Constructor & Destructor Documentation

### 6.345.2.1 DConfigDlgWdgModel()

```
Digikam::DConfigDlgWdgModel::DConfigDlgWdgModel (
    QObject *const parent = nullptr ) [explicit]
```

Creates a new page widget model.

#### Parameters

<i>parent</i>	The parent object.
---------------	--------------------

### 6.345.2.2 ~DConfigDlgWdgModel()

```
Digikam::DConfigDlgWdgModel::~DConfigDlgWdgModel ( ) [override], [default]
```

Destroys the page widget model.

## 6.345.3 Member Function Documentation

### 6.345.3.1 addPage() [1/2]

```
void Digikam::DConfigDlgWdgModel::addPage (
    DConfigDlgWdgItem * item )
```

Adds a new top level page to the model.

#### Parameters

<i>item</i>	The
-------------	-----

#### See also

[DConfigDlgWdgItem](#) which describes the page.

### 6.345.3.2 addPage() [2/2]

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdgModel::addPage (
    QWidget * widget,
    const QString & name )
```

Adds a new top level page to the model.

#### Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

**Returns**

The associated

**See also**

[DConfigDlgWdgItem](#).

**6.345.3.3 addSubPage() [1/2]**

```
void Digikam::DConfigDlgWdgModel::addSubPage (
    DConfigDlgWdgItem * parent,
    DConfigDlgWdgItem * item )
```

Inserts a new sub page in the model.

**Parameters**

<i>parent</i>	The new page will be insert as child of this
---------------	--

**See also**

[DConfigDlgWdgItem](#).

**Parameters**

<i>item</i>	The
-------------	-----

**See also**

[DConfigDlgWdgItem](#) which describes the page.

**6.345.3.4 addSubPage() [2/2]**

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdgModel::addSubPage (
    DConfigDlgWdgItem * parent,
    QWidget * widget,
    const QString & name )
```

Inserts a new sub page in the model.

**Parameters**

<i>parent</i>	The new page will be insert as child of this
---------------	--

**See also**

[DConfigDlgWdgItem](#).

## Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

## Returns

The associated

## See also

[DConfigDlgWdgItem](#).

**6.345.3.5 columnCount()**

```
int Digikam::DConfigDlgWdgModel::columnCount (
    const QModelIndex & parent = QModelIndex() ) const [override]
```

These methods are reimplemented from QAbstractItemModel.

**6.345.3.6 index()**

```
QModelIndex Digikam::DConfigDlgWdgModel::index (
    const DConfigDlgWdgItem * item ) const
```

Returns the index for a given

## See also

[DConfigDlgWdgItem](#). The index is invalid if the [item](#) can't be found in the model.

**6.345.3.7 insertPage() [1/2]**

```
void Digikam::DConfigDlgWdgModel::insertPage (
    DConfigDlgWdgItem * before,
    DConfigDlgWdgItem * item )
```

Inserts a new page in the model.

## Parameters

<i>before</i>	The new page will be insert before this
---------------	---

## See also

[DConfigDlgWdgItem](#) on the same level in hierarchy.



## Parameters

<i>item</i>	The
-------------	-----

## See also

[DConfigDlgWdgItem](#) which describes the page.

**6.345.3.8 insertPage() [2/2]**

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdgModel::insertPage (
    DConfigDlgWdgItem * before,
    QWidget * widget,
    const QString & name )
```

Inserts a new page in the model.

## Parameters

<i>before</i>	The new page will be insert before this
---------------	---

## See also

[DConfigDlgWdgItem](#) on the same level in hierarchy.

## Parameters

<i>widget</i>	The widget of the page.
<i>name</i>	The name which is displayed in the navigation view.

## Returns

The associated

## See also

[DConfigDlgWdgItem](#).

**6.345.3.9 item()**

```
DConfigDlgWdgItem * Digikam::DConfigDlgWdgModel::item (
    const QModelIndex & index ) const
```

Returns the

## See also

[DConfigDlgWdgItem](#) for a given index or 0 if the index is invalid.

### 6.345.3.10 removePage()

```
void Digikam::DConfigDlgWdgModel::removePage (
    DConfigDlgWdgItem * item )
```

Removes the page associated with the given

See also

[DConfigDlgWdgItem](#).

### 6.345.3.11 toggled

```
void Digikam::DConfigDlgWdgModel::toggled (
    DConfigDlgWdgItem * page,
    bool checked ) [signal]
```

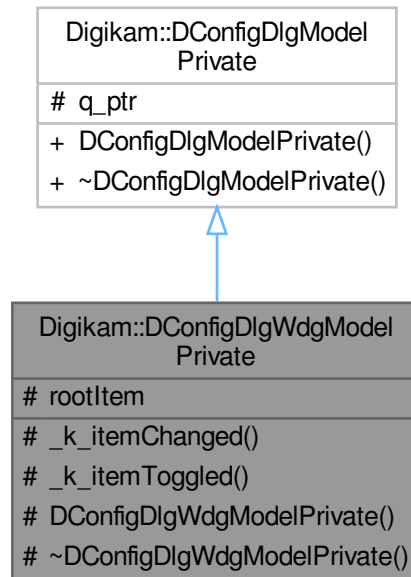
This signal is emitted whenever a checkable page changes its state.

Parameters

<i>checked</i>	is true when the
<i>page</i>	is checked, or false if the
<i>page</i>	is unchecked.

## 6.346 Digikam::DConfigDlgWdgModelPrivate Class Reference

Inheritance diagram for Digikam::DConfigDlgWdgModelPrivate:



### Protected Member Functions

- `void _k_itemChanged ()`
- `void _k_itemToggled (bool checked)`

### Protected Attributes

- `PageItem * rootItem = nullptr`

### Protected Attributes inherited from [Digikam::DConfigDlgModelPrivate](#)

- `DConfigDlgModel * q_ptr = nullptr`

## 6.347 Digikam::DConfigDlgWdgPrivate Class Reference

Inheritance diagram for Digikam::DConfigDlgWdgPrivate:



### Protected Member Functions

- `void _k_slotCurrentPageChanged` (const QModelIndex &, const QModelIndex &)
- `DConfigDlgWdgPrivate` (`DConfigDlgWdg` \*const q)
- `DConfigDlgWdgModel` \* `model` () const

### Protected Member Functions inherited from [Digikam::DConfigDlgViewPrivate](#)

- `void _k_dataChanged` (const QModelIndex &, const QModelIndex &)

- void `_k_modelChanged` ()
- void `_k_pageSelected` (const QItemSelection &, const QItemSelection &)
- void `_k_rebuildGui` ()
- void `cleanupPages` ()
- QList< QWidget \* > `collectPages` (const QModelIndex &parent=QModelIndex())
- `DConfigDlgViewPrivate` (`DConfigDlgView` \*const)
- `DConfigDlgView::FaceType` `detectAutoFace` () const
- void `updateSelection` ()
- void `updateTitleWidget` (const QModelIndex &index)

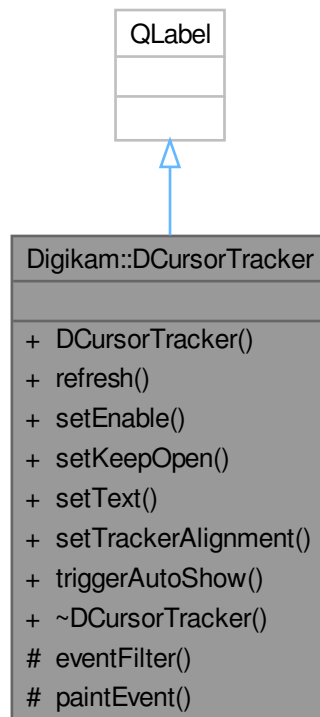
#### Additional Inherited Members

#### Protected Attributes inherited from `Digikam::DConfigDlgViewPrivate`

- QWidget \* `defaultWidget` = nullptr
- `DConfigDlgView::FaceType` `faceType` = `DConfigDlgView::Auto`
- QGridLayout \* `layout` = nullptr
- QAbstractItemModel \* `model` = nullptr
- `DConfigDlgView` \* `q_ptr` = nullptr
- `DConfigDlgStackedWidget` \* `stack` = nullptr
- `DConfigDlgTitle` \* `titleWidget` = nullptr
- QAbstractItemView \* `view` = nullptr

## 6.348 Digikam::DCursorTracker Class Reference

Inheritance diagram for `Digikam::DCursorTracker`:



## Public Member Functions

- **DCursorTracker** (const QString &txt, QWidget \*const parent, Qt::Alignment align=Qt::AlignCenter)
- void **refresh** ()
- void **setEnabled** (bool b)
- void **setKeepOpen** (bool b)
- void **setText** (const QString &txt)
- void **setTrackerAlignment** (Qt::Alignment alignment)
- void **triggerAutoShow** (int timeout=2000)

## Protected Member Functions

- bool **eventFilter** (QObject \*, QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override

### 6.348.1 Detailed Description

This class implements a window which looks like a tool tip. It will follow the cursor when it's over a specified widget.

### 6.348.2 Member Function Documentation

#### 6.348.2.1 setText()

```
void Digikam::DCursorTracker::setText (
    const QString & txt )
```

Overload to make sure the widget size is correct

## 6.349 Digikam::DDateEdit Class Reference

Inheritance diagram for Digikam::DDateEdit:



### Public Slots

- void `setDate` (const `QDate` &`date`)

### Signals

- void `dateChanged` (const `QDate` &`date`)

### Public Member Functions

- `QDate` `date` () const
- `DDateEdit` (`QWidget` \*const `parent`=nullptr, const `QString` &`name`=`QString`())
- bool `isReadOnly` () const
- void `setReadOnly` (bool `readOnly`)
- void `showPopup` () override

### Protected Slots

- void **dateEntered** (const QDate &)
- void **dateSelected** (const QDate &)
- void **lineEnterPressed** ()
- void **slotTextChanged** (const QString &)

### Protected Member Functions

- virtual bool [assignDate](#) (const QDate &date)
- bool **eventFilter** (QObject \*, QEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void [setupKeywords](#) ()

## 6.349.1 Detailed Description

A date editing widget that consists of an editable combo box. The combo box contains the date in text form, and clicking the combo box arrow will display a 'popup' style date picker.

This widget also supports advanced features like allowing the user to type in the day name to get the date. The following keywords are supported (in the native language): tomorrow, yesterday, today, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

## 6.349.2 Member Function Documentation

### 6.349.2.1 assignDate()

```
bool Digikam::DDateEdit::assignDate (
    const QDate & date ) [protected], [virtual]
```

Sets the date, without altering the display. This method is used internally to set the widget's date value. As a virtual method, it allows derived classes to perform additional validation on the date value before it is set. Derived classes should return true if `QDate::isValid(date)` returns false.

#### Parameters

<i>date</i>	The new date to set.
-------------	----------------------

#### Returns

True if the date was set, false if it was considered invalid and remains unchanged.

### 6.349.2.2 date()

```
QDate Digikam::DDateEdit::date ( ) const
```

#### Returns

The date entered. This date could be invalid, you have to check validity yourself.



### 6.349.2.3 dateChanged

```
void Digikam::DDateEdit::dateChanged (
    const QDate & date ) [signal]
```

This signal is emitted whenever the user modifies the date. The passed date can be invalid.

### 6.349.2.4 isReadOnly()

```
bool Digikam::DDateEdit::isReadOnly ( ) const
```

#### Returns

True if the widget is read-only, false if read-write.

### 6.349.2.5 setDate

```
void Digikam::DDateEdit::setDate (
    const QDate & date ) [slot]
```

Sets the date.

#### Parameters

<i>date</i>	The new date to display. This date must be valid or it will not be set
-------------	--

### 6.349.2.6 setReadOnly()

```
void Digikam::DDateEdit::setReadOnly (
    bool readOnly )
```

Sets whether the widget is read-only for the user. If read-only, the date picker pop-up is inactive, and the displayed date cannot be edited.

#### Parameters

<i>readOnly</i>	True to set the widget read-only, false to set it read-write.
-----------------	---

### 6.349.2.7 setupKeywords()

```
void Digikam::DDateEdit::setupKeywords ( ) [protected]
```

Fills the keyword map. Re-implement it if you want additional keywords.

## 6.350 Digikam::DDatePicker Class Reference

Inheritance diagram for Digikam::DDatePicker:



### Classes

- class [Private](#)

## Signals

- void [dateChanged](#) (const QDate &date)
- void [dateEntered](#) (const QDate &date)
- void [dateSelected](#) (const QDate &date)
- void [tableClicked](#) ()

## Public Member Functions

- const QDate & [date](#) () const
- [DDateTable](#) \* [dateTable](#) () const
- [DDatePicker](#) (const QDate &dt, QWidget \*const parent=nullptr)
- [DDatePicker](#) (QWidget \*const parent=nullptr)
- int [fontSize](#) () const
- bool [hasCloseButton](#) () const
- void [setCloseButton](#) (bool enable)
- bool [setDate](#) (const QDate &date)
- void [setFontSize](#) (int)
- QSize [sizeHint](#) () const override
- [~DDatePicker](#) () override

## Protected Slots

- void [dateChangedSlot](#) (const QDate &date)
- void [lineEnterPressed](#) ()
- void [monthBackwardClicked](#) ()
- void [monthForwardClicked](#) ()
- void [selectMonthClicked](#) ()
- void [selectYearClicked](#) ()
- void [tableClickedSlot](#) ()
- void [todayButtonClicked](#) ()
- void [uncheckYearSelector](#) ()
- void [weekSelected](#) (int)
- void [yearBackwardClicked](#) ()
- void [yearForwardClicked](#) ()

## Protected Member Functions

- void [changeEvent](#) (QEvent \*) override
- bool [eventFilter](#) (QObject \*, QEvent \*) override  
*to catch move keyEvents when QLineEdit has keyFocus*
- void [resizeEvent](#) (QResizeEvent \*) override  
*the resize event*

## Properties

- bool [closeButton](#)
- QDate [date](#)
- int [fontSize](#)

## Friends

- class **Private**

### 6.350.1 Detailed Description

Provides a widget for calendar date input.

### 6.350.2 Constructor & Destructor Documentation

#### 6.350.2.1 DDatePicker() [1/2]

```
Digikam::DDatePicker::DDatePicker (
    QWidget *const parent = nullptr ) [explicit]
```

The constructor. The current date will be displayed initially.

#### 6.350.2.2 DDatePicker() [2/2]

```
Digikam::DDatePicker::DDatePicker (
    const QDate & dt,
    QWidget *const parent = nullptr ) [explicit]
```

The constructor. The given date will be displayed initially.

#### 6.350.2.3 ~DDatePicker()

```
Digikam::DDatePicker::~DDatePicker ( ) [override]
```

The destructor.

### 6.350.3 Member Function Documentation

#### 6.350.3.1 date()

```
const QDate & Digikam::DDatePicker::date ( ) const
```

#### Returns

the selected date.

### 6.350.3.2 dateChanged

```
void Digikam::DDatePicker::dateChanged (
    const QDate & date ) [signal]
```

This signal is emitted each time the selected date is changed. Usually, this does not mean that the date has been entered, since the date also changes, for example, when another month is selected.

See also

[dateSelected](#)

### 6.350.3.3 dateEntered

```
void Digikam::DDatePicker::dateEntered (
    const QDate & date ) [signal]
```

This signal is emitted when enter is pressed and a VALID date has been entered before into the line edit. Connect to both [dateEntered\(\)](#) and [dateSelected\(\)](#) to receive all events where the user really enters a date.

### 6.350.3.4 dateSelected

```
void Digikam::DDatePicker::dateSelected (
    const QDate & date ) [signal]
```

This signal is emitted each time a day has been selected by clicking on the table (hitting a day in the current month). It has the same meaning as [dateSelected\(\)](#) in older versions of [DDatePicker](#).

### 6.350.3.5 dateTable()

```
DDateTable * Digikam::DDatePicker::dateTable ( ) const
```

Returns

the [DDateTable](#) widget child of this [DDatePicker](#) widget.

### 6.350.3.6 fontSize()

```
int Digikam::DDatePicker::fontSize ( ) const
```

Returns the font size of the widget elements.

### 6.350.3.7 hasCloseButton()

```
bool Digikam::DDatePicker::hasCloseButton ( ) const
```

#### Returns

true if a [DDatePicker](#) shows a close-button.

#### See also

[setCloseButton](#)

### 6.350.3.8 setCloseButton()

```
void Digikam::DDatePicker::setCloseButton (
    bool enable )
```

By calling this method with `enable = true`, [DDatePicker](#) will show a little close-button in the upper button-row. Clicking the close-button will cause the [DDatePicker](#)'s `topLevelWidget()`'s `close()` method being called. This is mostly useful for toplevel datepickers without a window manager decoration.

#### See also

[hasCloseButton](#)

### 6.350.3.9 setDate()

```
bool Digikam::DDatePicker::setDate (
    const QDate & date )
```

Sets the date.

#### Returns

`false` and does not change anything if the date given is invalid.

### 6.350.3.10 setFontSize()

```
void Digikam::DDatePicker::setFontSize (
    int s )
```

Sets the font size of the widgets elements.

### 6.350.3.11 sizeHint()

```
QSize Digikam::DDatePicker::sizeHint ( ) const [override]
```

The size hint for date pickers. The size hint recommends the minimum size of the widget so that all elements may be placed without clipping. This sometimes looks ugly, so when using the size hint, try adding 28 to each of the reported numbers of pixels.

### 6.350.3.12 tableClicked

```
void Digikam::DDatePicker::tableClicked ( ) [signal]
```

This signal is emitted when the day has been selected by clicking on it in the table.

## 6.351 Digikam::DDatePicker::Private Class Reference

### Public Member Functions

- void [fillWeeksCombo](#) ()
- **Private** ([DDatePicker](#) \*const qq)
- [QDate](#) **validDateInYearMonth** (int year, int month)

### Public Attributes

- [QToolButton](#) \* **closeButton** = nullptr
- int **fontSize** = 0  
*the font size for the widget*
- [QLineEdit](#) \* **line** = nullptr  
*the line edit to enter the date directly*
- [QSize](#) **maxMonthRect**  
*the widest month string in pixels:*
- [QToolButton](#) \* **monthBackward** = nullptr  
*the month backward button*
- [QToolButton](#) \* **monthForward** = nullptr  
*the month forward button*
- [QBoxLayout](#) \* **navigationLayout** = nullptr
- [DDatePicker](#) \* **q** = nullptr  
*the date table*
- [QToolButton](#) \* **selectMonth** = nullptr  
*the button for selecting the month directly*
- [QComboBox](#) \* **selectWeek** = nullptr
- [QToolButton](#) \* **selectYear** = nullptr  
*the button for selecting the year directly*
- [DDateTable](#) \* **table** = nullptr  
*the date table*
- [QToolButton](#) \* **todayButton** = nullptr
- [DatePickerValidator](#) \* **val** = nullptr  
*the validator for the line edit:*
- [QToolButton](#) \* **yearBackward** = nullptr  
*the year backward button*
- [QToolButton](#) \* **yearForward** = nullptr  
*the year forward button*

## 6.351.1 Member Function Documentation

### 6.351.1.1 fillWeeksCombo()

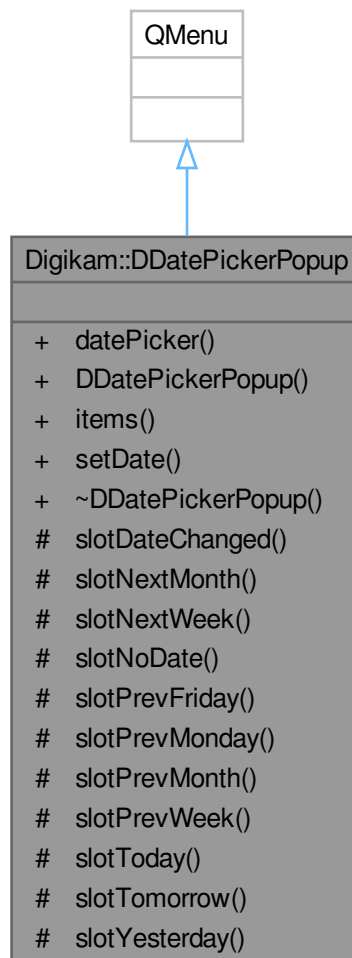
```
void Digikam::DDatePicker::Private::fillWeeksCombo ( )
```

NOTE: every year can have a different number of weeks it could be that we had 53,1..52 and now 1..53 which is the same number but different so always fill with new values We show all week numbers for all weeks between first day of year to last day of year This of course can be a list like 53,1,2..52

## 6.352 Digikam::DDatePickerPopup Class Reference

This menu helps the user to select a date quickly.

Inheritance diagram for Digikam::DDatePickerPopup:





## Public Types

- enum **ItemFlag** { **NoDate** = 1 , **DatePicker** = 2 , **Words** = 4 }

## Signals

- void **dateChanged** (const QDate &)

## Public Member Functions

- **DDatePicker** \* **datePicker** () const
- **DDatePickerPopup** (Items **items**, const QDate &**date**=QDate::currentDate(), QWidget \*const **parent**=nullptr)
- int **items** () const
- void **setDate** (const QDate &**date**)

## Protected Slots

- void **slotDateChanged** (const QDate &)
- void **slotNextMonth** ()
- void **slotNextWeek** ()
- void **slotNoDate** ()
- void **slotPrevFriday** ()
- void **slotPrevMonday** ()
- void **slotPrevMonth** ()
- void **slotPrevWeek** ()
- void **slotToday** ()
- void **slotTomorrow** ()
- void **slotYesterday** ()

### 6.352.1 Detailed Description

This menu helps the user to select a date quickly. It offers various ways of selecting, e.g. with a [DDatePicker](#) or with words like "Tomorrow".

The available items are:

- NoDate: A menu-item with "No Date". If chosen, the datepicker will emit a null QDate.
- DatePicker: Show a DDatePicker-widget.
- Words: Show items like "Today", "Tomorrow" or "Next Week".

When supplying multiple items, separate each item with a bitwise OR.

### 6.352.2 Constructor & Destructor Documentation

#### 6.352.2.1 DDatePickerPopup()

```
Digikam::DDatePickerPopup::DDatePickerPopup (
    Items items,
    const QDate & date = QDate::currentDate(),
    QWidget *const parent = nullptr ) [explicit]
```

A constructor for the [DDatePickerPopup](#).

**Parameters**

<i>items</i>	List of all desirable items, separated with a bitwise OR.
<i>date</i>	Initial date of datepicker-widget.
<i>parent</i>	The object's parent.

**6.352.3 Member Function Documentation****6.352.3.1 dateChanged**

```
void Digikam::DDatePickerPopup::dateChanged (
    const QDate & ) [signal]
```

This signal emits the new date (selected with datepicker or other menu-items).

**6.352.3.2 datePicker()**

```
DDatePicker * Digikam::DDatePickerPopup::datePicker ( ) const
```

**Returns**

A pointer to the private variable `mDatePicker`, an instance of [DDatePicker](#).

**6.352.3.3 items()**

```
int Digikam::DDatePickerPopup::items ( ) const
```

**Returns**

Returns the bitwise result of the active items in the popup.

## 6.353 Digikam::DDateTable Class Reference

Inheritance diagram for Digikam::DDateTable:



### Classes

- class [Private](#)

### Public Types

- enum `BackgroundMode` { `NoBgMode` = 0 , `RectangleMode` , `CircleMode` }

## Signals

- void [aboutToShowContextMenu](#) (QMenu \*menu, const QDate &dt)
- void [dateChanged](#) (const QDate &cur, const QDate &old)
- void [dateChanged](#) (const QDate &date)
- void [tableClicked](#) ()

## Public Member Functions

- const QDate & [date](#) () const
- **DDateTable** (const QDate &dt, QWidget \*const parent=nullptr)
- **DDateTable** (QWidget \*const parent=nullptr)
- bool [popupMenuEnabled](#) () const
- void [setCustomDatePainting](#) (const QDate &date, const QColor &fgColor, BackgroundMode bgMode=No↔BgMode, const QColor &bgColor=QColor())
- bool [setDate](#) (const QDate &date)
- void [setFontSize](#) (int size)
- void [setPopupMenuEnabled](#) (bool enable)
- QSize [sizeHint](#) () const override
- void [unsetCustomDatePainting](#) (const QDate &dt)

## Protected Member Functions

- virtual QDate [dateFromPos](#) (int pos)
- bool [event](#) (QEvent \*e) override
- void **focusInEvent** (QFocusEvent \*e) override
- void **focusOutEvent** (QFocusEvent \*e) override
- void **keyPressEvent** (QKeyEvent \*e) override
- void [mousePressEvent](#) (QMouseEvent \*e) override
- void **paintEvent** (QPaintEvent \*e) override
- virtual int [posFromDate](#) (const QDate &dt)
- void **wheelEvent** (QWheelEvent \*e) override

## Properties

- QDate **date**
- bool **popupMenu**

## Friends

- class **Private**

### 6.353.1 Detailed Description

This is a support class for the [DDatePicker](#) class. It just draws the calendar table without titles, but could theoretically be used as a standalone.

When a date is selected by the user, it emits a signal: `dateSelected(QDate)`

## 6.353.2 Member Function Documentation

### 6.353.2.1 aboutToShowContextMenu

```
void Digikam::DDateTable::aboutToShowContextMenu (
    QMenu * menu,
    const QDate & dt ) [signal]
```

A popup menu for a given date is about to be shown (as when the user right clicks on that date and the popup menu is enabled). Connect the slot where you fill the menu to this signal.

### 6.353.2.2 date()

```
const QDate & Digikam::DDateTable::date ( ) const
```

#### Returns

the selected date.

### 6.353.2.3 dateChanged [1/2]

```
void Digikam::DDateTable::dateChanged (
    const QDate & cur,
    const QDate & old ) [signal]
```

This function behaves essentially like the one above. The selected date changed.

#### Parameters

<i>cur</i>	The current date
<i>old</i>	The date before the date was changed

### 6.353.2.4 dateChanged [2/2]

```
void Digikam::DDateTable::dateChanged (
    const QDate & date ) [signal]
```

The selected date changed.

### 6.353.2.5 dateFromPos()

```
QDate Digikam::DDateTable::dateFromPos (
    int pos ) [protected], [virtual]
```

calculate the date that is displayed at a given cell in the matrix. *pos* is the 0-based index in the matrix. Inverse function to `posFromDate()`.

### 6.353.2.6 event()

```
bool Digikam::DDateTable::event (
    QEvent * e ) [override], [protected]
```

Cell highlight on mouse hovering

### 6.353.2.7 mousePressEvent()

```
void Digikam::DDateTable::mousePressEvent (
    QMouseEvent * e ) [override], [protected]
```

React on mouse clicks that select a date.

### 6.353.2.8 popupMenuEnabled()

```
bool Digikam::DDateTable::popupMenuEnabled ( ) const
```

Returns if the popup menu is enabled or not

### 6.353.2.9 posFromDate()

```
int Digikam::DDateTable::posFromDate (
    const QDate & dt ) [protected], [virtual]
```

calculate the position of the cell in the matrix for the given date. The result is the 0-based index.

### 6.353.2.10 setCustomDatePainting()

```
void Digikam::DDateTable::setCustomDatePainting (
    const QDate & date,
    const QColor & fgColor,
    BackgroundMode bgMode = NoBgMode,
    const QColor & bgColor = QColor() )
```

Makes a given date be painted with a given foregroundColor, and background (a rectangle, or a circle/ellipse) in a given color.

### 6.353.2.11 setDate()

```
bool Digikam::DDateTable::setDate (
    const QDate & date )
```

Select and display this date.

**6.353.2.12 setFontSize()**

```
void Digikam::DDateTable::setFontSize (
    int size )
```

Set the font size of the date table.

**6.353.2.13 setPopupMenuEnabled()**

```
void Digikam::DDateTable::setPopupMenuEnabled (
    bool enable )
```

Enables a popup menu when right clicking on a date.

When it's enabled, this object emits a `aboutToShowContextMenu` signal where you can fill in the menu items.

**6.353.2.14 sizeHint()**

```
QSize Digikam::DDateTable::sizeHint ( ) const [override]
```

Returns a recommended size for the widget. To save some time, the size of the largest used cell content is calculated in each `paintCell()` call, since all calculations have to be done there anyway. The size is stored in `maxCell`. The `sizeHint()` simply returns a multiple of `maxCell`.

**6.353.2.15 tableClicked**

```
void Digikam::DDateTable::tableClicked ( ) [signal]
```

A date has been selected by clicking on the table.

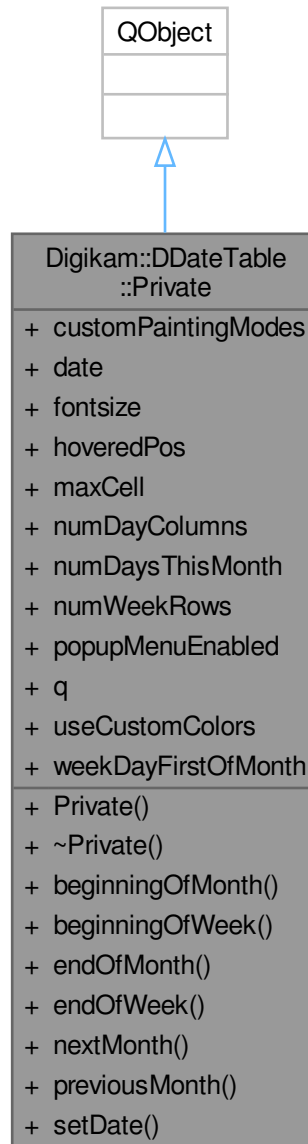
**6.353.2.16 unsetCustomDatePainting()**

```
void Digikam::DDateTable::unsetCustomDatePainting (
    const QDate & dt )
```

Unsets the custom painting of a date so that the date is painted as usual.

## 6.354 Digikam::DDateTable::Private Class Reference

Inheritance diagram for Digikam::DDateTable::Private:



### Classes

- class [DatePaintingMode](#)

### Public Slots

- void **beginningOfMonth** ()



- void **beginningOfWeek** ()
- void **endOfMonth** ()
- void **endOfWeek** ()
- void **nextMonth** ()
- void **previousMonth** ()
- void **setDate** (const QDate &)

### Public Member Functions

- **Private** (DDateTable \*const qq)

### Public Attributes

- QHash< int, DatePaintingMode > **customPaintingModes**
- QDate **date**
- int **fontsize** = 0
- int **hoveredPos** = -1
- QRectF **maxCell**
- int **numDayColumns** = 0
- int **numDaysThisMonth** = 0
- int **numWeekRows** = 0
- bool **popupMenuEnabled** = false
- DDateTable \* **q** = nullptr
- bool **useCustomColors** = false
- int **weekDayFirstOfMonth** = 0

## 6.354.1 Member Data Documentation

### 6.354.1.1 date

```
QDate Digikam::DDateTable::Private::date
```

The currently selected date.

### 6.354.1.2 fontsize

```
int Digikam::DDateTable::Private::fontsize = 0
```

The font size of the displayed text.

### 6.354.1.3 maxCell

```
QRectF Digikam::DDateTable::Private::maxCell
```

Save the size of the largest used cell content.

#### 6.354.1.4 numDayColumns

```
int Digikam::DDateTable::Private::numDayColumns = 0
```

How many day columns we are to draw, i.e. days in a week.

#### 6.354.1.5 numDaysThisMonth

```
int Digikam::DDateTable::Private::numDaysThisMonth = 0
```

The number of days in the current month.

#### 6.354.1.6 numWeekRows

```
int Digikam::DDateTable::Private::numWeekRows = 0
```

How many week rows we are to draw.

#### 6.354.1.7 weekDayFirstOfMonth

```
int Digikam::DDateTable::Private::weekDayFirstOfMonth = 0
```

The weekday number of the first day in the month [1..daysInWeek()].

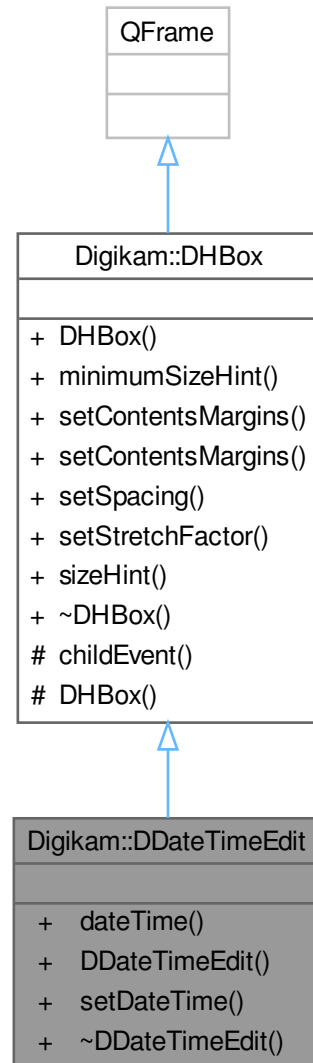
## 6.355 Digikam::DDateTable::Private::DatePaintingMode Class Reference

### Public Attributes

- QColor **bgColor**
- BackgroundMode **bgMode**
- QColor **fgColor**

## 6.356 Digikam::DDateTimeEdit Class Reference

Inheritance diagram for Digikam::DDateTimeEdit:



### Signals

- void `dateTimeChanged` (const QDateTime &`dateTime`)

### Public Member Functions

- QDateTime `dateTime` () const
- `DDateTimeEdit` (QWidget \*const parent, const QString &name)
- void `setDateTime` (const QDateTime &`dateTime`)
- `~DDateTimeEdit` () override

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &argins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

### 6.356.1 Detailed Description

This class is basically the same as the KDE Date Time widget with the exception that a QTimeEdit is placed directly besides it.

### 6.356.2 Constructor & Destructor Documentation

#### 6.356.2.1 DDateTimeEdit()

```
Digikam::DDateTimeEdit::DDateTimeEdit (
    QWidget *const parent,
    const QString & name ) [explicit]
```

constructor

#### Parameters

<i>parent</i>	the parent widget
<i>name</i>	the name of the widget

#### 6.356.2.2 ~DDateTimeEdit()

```
Digikam::DDateTimeEdit::~DDateTimeEdit ( ) [override]
```

destructor

### 6.356.3 Member Function Documentation

#### 6.356.3.1 dateTime()

```
QDateTime Digikam::DDateTimeEdit::dateTime ( ) const
```

returns the date and time

#### Returns

a QDateTime with the currently chosen date and time

#### 6.356.3.2 dateTimeChanged

```
void Digikam::DDateTimeEdit::dateTimeChanged (
    const QDateTime & dateTime ) [signal]
```

This signal is emitted whenever the user modifies the date or time. The passed date and time can be invalid.

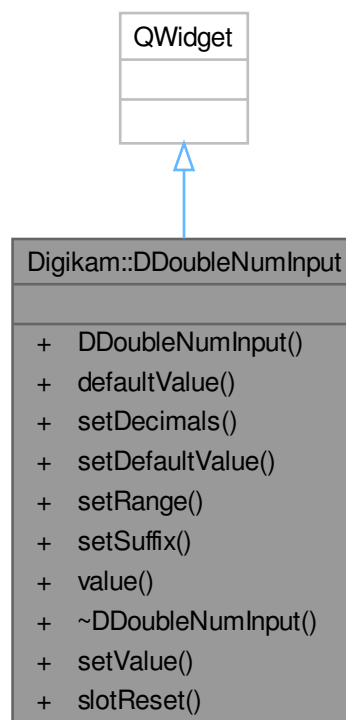
#### 6.356.3.3 setDateTime()

```
void Digikam::DDateTimeEdit::setDateTime (
    const QDateTime & dateTime )
```

Sets the date and the time of this widget.

## 6.357 Digikam::DDoubleNumInput Class Reference

Inheritance diagram for Digikam::DDoubleNumInput:



**Public Slots**

- void **setValue** (double d)
- void **slotReset** ()

**Signals**

- void **reset** ()
- void **valueChanged** (double)

**Public Member Functions**

- **DDoubleNumInput** (QWidget \*const parent=nullptr)
- double **defaultValue** () const
- void **setDecimals** (int p)
- void **setDefaultValue** (double d)
- void **setRange** (double min, double max, double step)
- void **setSuffix** (const QString &suffix)
- double **value** () const

## 6.358 Digikam::DDoubleSliderSpinBox Class Reference

Inheritance diagram for Digikam::DDoubleSliderSpinBox:



### Public Slots

- void **setValue** (double value)

## Signals

- void **valueChanged** (double value)

## Public Member Functions

- **DDoubleSliderSpinBox** (QWidget \*const parent=nullptr)
- double **fastSliderStep** () const
- double **maximum** () const
- double **minimum** () const
- void **setFastSliderStep** (double step)
- void **setMaximum** (double maximum)
- void **setMinimum** (double minimum)
- void **setRange** (double minimum, double maximum, int decimals=0)
- void **setSingleStep** (double value)
- double **value** ()

## Public Member Functions inherited from [Digikam::DAbstractSliderSpinBox](#)

- void **hideEdit** ()
- bool **isDragging** () const
- virtual QSize **minimumSize** () const
- QSize **minimumSizeHint** () const override
- void **setBlockUpdateSignalOnDrag** (bool block)
- void **setExponentRatio** (double dbl)
- void **setPrefix** (const QString &prefix)
- void **setSuffix** (const QString &suffix)
- void **showEdit** ()
- QSize **sizeHint** () const override

## Protected Member Functions

- void **setInternalValue** (int value, bool blockUpdateSignal) override
- QString **valueString** () const override

## Protected Member Functions inherited from [Digikam::DAbstractSliderSpinBox](#)

- void **changeEvent** (QEvent \*e) override
- **DAbstractSliderSpinBox** (QWidget \*const parent, DAbstractSliderSpinBoxPrivate \*const q)
- QRect **downButtonRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- bool **eventFilter** (QObject \*recv, QEvent \*e) override
- void **focusInEvent** (QFocusEvent \*e) override
- void **keyPressEvent** (QKeyEvent \*e) override
- void **mouseMoveEvent** (QMouseEvent \*e) override
- void **mousePressEvent** (QMouseEvent \*e) override
- void **mouseReleaseEvent** (QMouseEvent \*e) override
- void **paint** (QPainter &painter)
- void **paintBreeze** (QPainter &painter)
- void **paintEvent** (QPaintEvent \*e) override
- void **paintFusion** (QPainter &painter)
- void **paintPlastique** (QPainter &painter)
- QStyleOptionProgressBar **progressBarOptions** () const
- QRect **progressRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- QStyleOptionSpinBox **spinBoxOptions** () const
- QRect **upButtonRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- int **valueForX** (int x, Qt::KeyboardModifiers modifiers=Qt::NoModifier) const
- void **wheelEvent** (QWheelEvent \*e) override



## Additional Inherited Members

### Protected Slots inherited from [Digikam::DAbstractSliderSpinBox](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override
- void **editLostFocus** ()

### Protected Attributes inherited from [Digikam::DAbstractSliderSpinBox](#)

- DAbstractSliderSpinBoxPrivate \*const **d\_ptr**

## 6.358.1 Member Function Documentation

### 6.358.1.1 setInternalValue()

```
void Digikam::DDoubleSliderSpinBox::setInternalValue (
    int value,
    bool blockUpdateSignal ) [override], [protected], [virtual]
```

Sets the slider internal value. Inheriting classes should respect blockUpdateSignal so that, in specific cases, we have a performance improvement. See setIgnoreMouseMoveEvents.

Implements [Digikam::DAbstractSliderSpinBox](#).

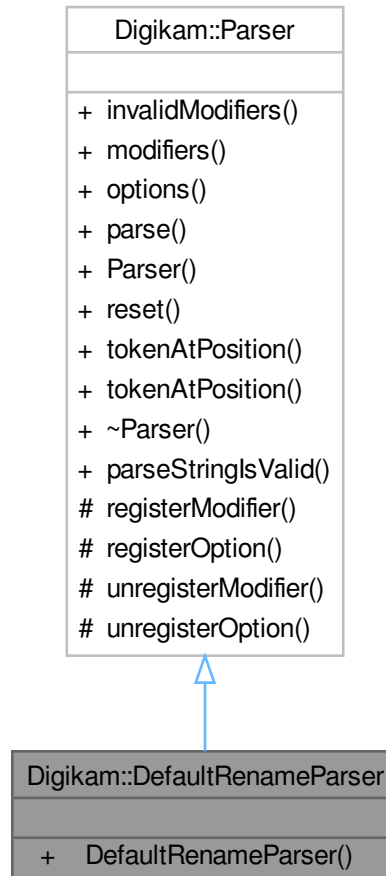
### 6.358.1.2 valueString()

```
QString Digikam::DDoubleSliderSpinBox::valueString ( ) const [override], [protected], [virtual]
```

Implements [Digikam::DAbstractSliderSpinBox](#).

## 6.359 Digikam::DefaultRenameParser Class Reference

Inheritance diagram for Digikam::DefaultRenameParser:



### Additional Inherited Members

### Public Member Functions inherited from [Digikam::Parser](#)

- [ParseResults](#) `invalidModifiers` ([ParseSettings](#) &settings)
- RulesList `modifiers` () const
- RulesList `options` () const
- QString `parse` ([ParseSettings](#) &settings)
- void `reset` ()
- bool `tokenAtPosition` ([ParseSettings](#) &settings, int pos)
- bool `tokenAtPosition` ([ParseSettings](#) &settings, int pos, int &start, int &length)

### Static Public Member Functions inherited from [Digikam::Parser](#)

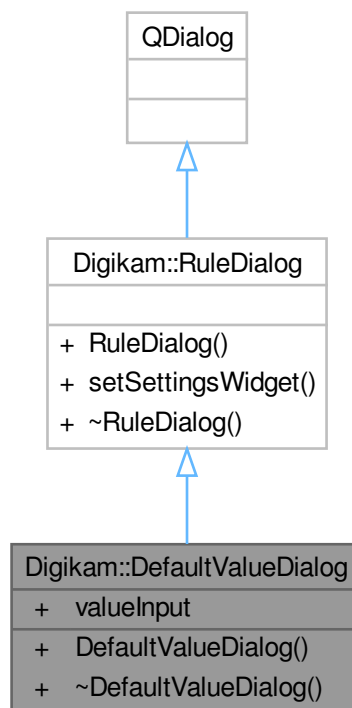
- static bool `parseStringIsValid` (const QString &str)

### Protected Member Functions inherited from [Digikam::Parser](#)

- void **registerModifier** ([Rule](#) \*modifier)
- void **registerOption** ([Rule](#) \*option)
- void **unregisterModifier** (const [Rule](#) \*modifier)
- void **unregisterOption** (const [Rule](#) \*option)

## 6.360 Digikam::DefaultValueDialog Class Reference

Inheritance diagram for Digikam::DefaultValueDialog:



### Public Member Functions

- **DefaultValueDialog** ([Rule](#) \*parent)

### Public Member Functions inherited from [Digikam::RuleDialog](#)

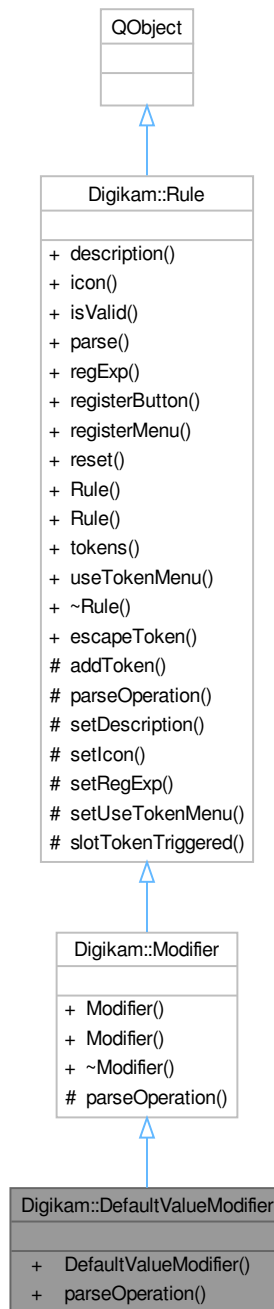
- **RuleDialog** ([Rule](#) \*const parent)
- void **setSettingsWidget** ([QWidget](#) \*const settingsWidget)

## Public Attributes

- QLineEdit \* **valueInput** = nullptr

## 6.361 Digikam::DefaultValueModifier Class Reference

Inheritance diagram for Digikam::DefaultValueModifier:



## Public Member Functions

- QString [parseOperation](#) ([ParseSettings](#) &settings, const QRegularExpressionMatch &match) override

## Public Member Functions inherited from [Digikam::Modifier](#)

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- [ParseResults](#) **parse** ([ParseSettings](#) &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from [Digikam::Rule](#)

- void **signalTokenTriggered** (const QString &)

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool **addToken** (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void **setUseTokenMenu** (bool value)

## 6.361.1 Member Function Documentation

### 6.361.1.1 parseOperation()

```
QString Digikam::DefaultValueModifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [virtual]
```

TODO: describe me

#### Parameters

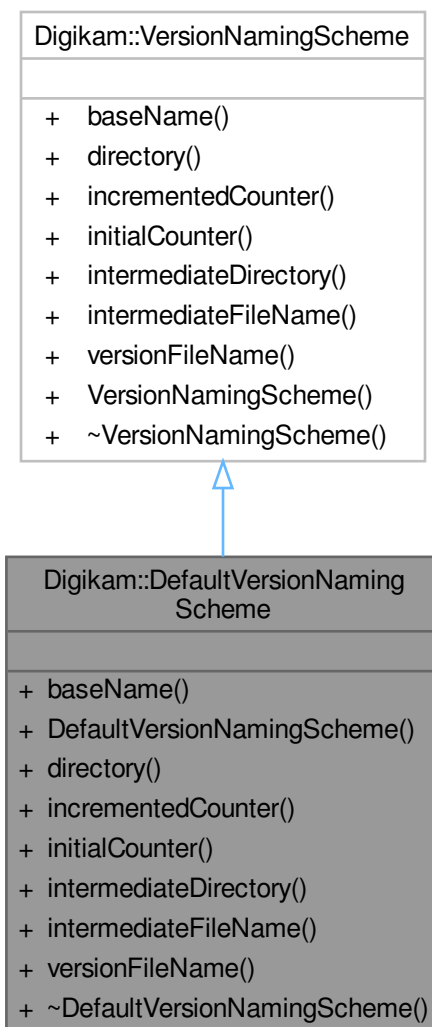
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in Option::parse()

#### Returns

Implements [Digikam::Modifier](#).

## 6.362 Digikam::DefaultVersionNamingScheme Class Reference

Inheritance diagram for Digikam::DefaultVersionNamingScheme:



### Public Member Functions

- virtual QString [baseName](#) (const QString &currentPath, const QString &filename, QVariant \*counter, QVariant \*intermediateCounter) override
- virtual QString [directory](#) (const QString &currentPath, const QString &filename) override
- virtual QVariant [incrementedCounter](#) (const QVariant &counter) override
- virtual QVariant [initialCounter](#) () override
- virtual QString [intermediateDirectory](#) (const QString &currentPath, const QString &fileName) override
- virtual QString [intermediateFileName](#) (const QString &currentPath, const QString &filename, const QVariant &version, const QVariant &counter) override
- virtual QString [versionFileName](#) (const QString &currentPath, const QString &filename, const QVariant &counter) override

## Public Member Functions inherited from [Digikam::VersionNamingScheme](#)

- [VersionNamingScheme](#) ()=default

### 6.362.1 Member Function Documentation

#### 6.362.1.1 `baseName()`

```
QString Digikam::DefaultVersionNamingScheme::baseName (
    const QString & path,
    const QString & filename,
    QVariant * counter,
    QVariant * intermediateCounter ) [override], [virtual]
```

Analyzes the given file name. Returns the basename in the sense of stripping the file name of all added version information: A scheme that appends a number, like "MyFile-1.jpg", shall return "MyFile". Path is the directory, filename the file name, so path + filename is the file path. If counter is given, and the given file name has a version number, write it to counter. If intermediateCounter is given, and the given file name has an intermediate counter number, write it to counter. If not available, do not touch the given counters. See [initialCounter\(\)](#) for the valid counter formats.

Implements [Digikam::VersionNamingScheme](#).

#### 6.362.1.2 `directory()`

```
QString Digikam::DefaultVersionNamingScheme::directory (
    const QString & path,
    const QString & filename ) [override], [virtual]
```

For a loaded file in directory path and with file name filename, returns the directory in which a new version (a new intermediate version, resp.) shall be stored.

Implements [Digikam::VersionNamingScheme](#).

#### 6.362.1.3 `incrementedCounter()`

```
QVariant Digikam::DefaultVersionNamingScheme::incrementedCounter (
    const QVariant & counter ) [override], [virtual]
```

Returns the given counter "incremented", that is, changed in a steady, repeatable fashion. You shall never return the given counter.

Implements [Digikam::VersionNamingScheme](#).

#### 6.362.1.4 `initialCounter()`

```
QVariant Digikam::DefaultVersionNamingScheme::initialCounter ( ) [override], [virtual]
```

Returns an initial counter value for version and intermediate number counters. There are two places where you shall generate counters You will receive the given QVariant in [incrementedCounter\(\)](#), [versionFileName\(\)](#) and [baseName\(\)](#), and you shall read a counter value from a generated file name in [baseName\(\)](#).

Implements [Digikam::VersionNamingScheme](#).



### 6.362.1.5 intermediateDirectory()

```
QString Digikam::DefaultVersionNamingScheme::intermediateDirectory (
    const QString & currentPath,
    const QString & fileName ) [override], [virtual]
```

Implements [Digikam::VersionNamingScheme](#).

### 6.362.1.6 intermediateFileName()

```
QString Digikam::DefaultVersionNamingScheme::intermediateFileName (
    const QString & path,
    const QString & filename,
    const QVariant & version,
    const QVariant & counter ) [override], [virtual]
```

Creates a version file name for an intermediate file in given directory, as previously returned by [directory\(\)](#), given *baseName*, as previously returned by *baseName*, *version* and intermediate number counter. Do not append a file suffix. You do not need to check if the file exists.

Implements [Digikam::VersionNamingScheme](#).

### 6.362.1.7 versionFileName()

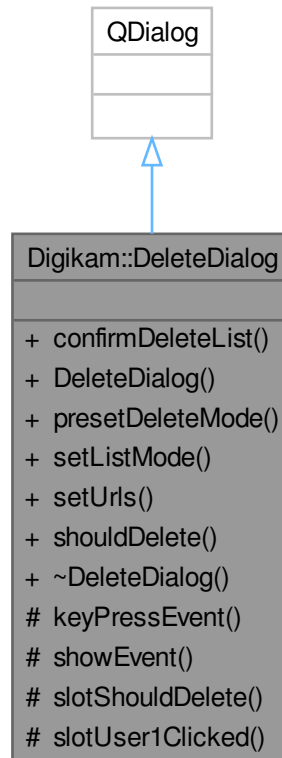
```
QString Digikam::DefaultVersionNamingScheme::versionFileName (
    const QString & path,
    const QString & baseName,
    const QVariant & counter ) [override], [virtual]
```

Creates a version file name for a file in given directory, as previously returned by [directory\(\)](#), given *baseName*, as previously returned by *baseName*, and *version* counter. Do not append a file suffix. You do not need to check if the file exists.

Implements [Digikam::VersionNamingScheme](#).

## 6.363 Digikam::DeleteDialog Class Reference

Inheritance diagram for Digikam::DeleteDialog:



### Public Types

- enum **Mode** { **ModeFiles** , **ModeAlbums** , **ModeSubalbums** }

### Public Member Functions

- bool **confirmDeleteList** (const QList< QUrl > &condemnedURLs, DeleteDialogMode::ListMode listMode, DeleteDialogMode::DeleteMode deleteMode)
- **DeleteDialog** (QWidget \*const parent)
- void **presetDeleteMode** (DeleteDialogMode::DeleteMode mode)
- void **setListMode** (DeleteDialogMode::ListMode mode)
- void **setUrls** (const QList< QUrl > &urls)
- bool **shouldDelete** () const

### Protected Slots

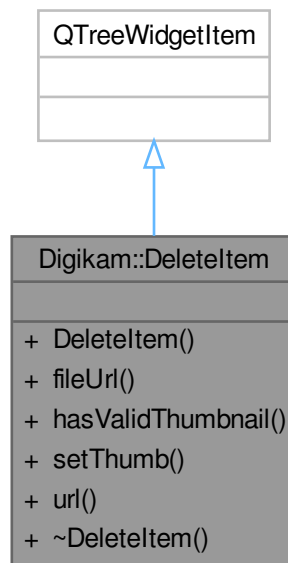
- void **slotShouldDelete** (bool)
- void **slotUser1Clicked** ()

### Protected Member Functions

- void **keyPressEvent** (QKeyEvent \*) override
- void **showEvent** (QShowEvent \*) override

## 6.364 Digikam::Deleteltem Class Reference

Inheritance diagram for Digikam::Deleteltem:

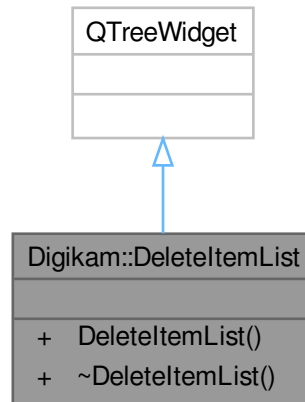


### Public Member Functions

- **Deleteltem** (QTreeWidgetItem \*const parent, const QUrl &url)
- QString **fileUrl** () const
- bool **hasValidThumbnail** () const
- void **setThumb** (const QPixmap &pix, bool hasThumb=true)
- QUrl **url** () const

## 6.365 Digikam::DeleteltemList Class Reference

Inheritance diagram for Digikam::DeleteltemList:

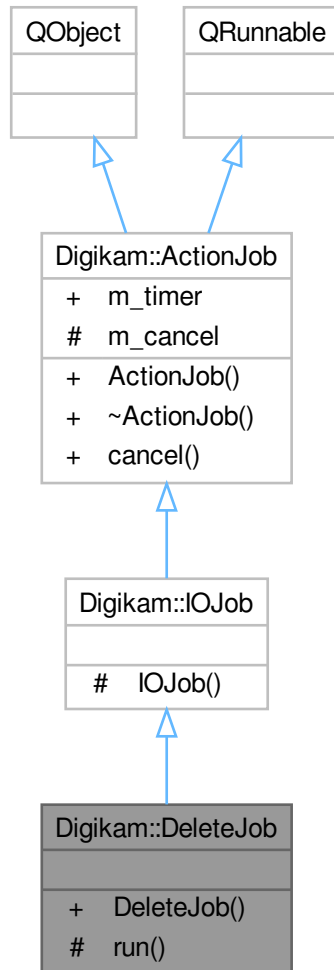


### Public Member Functions

- **DeleteltemList** (QWidget \*const parent=nullptr)

## 6.366 Digikam::DeleteJob Class Reference

Inheritance diagram for Digikam::DeleteJob:



### Public Member Functions

- `DeleteJob` ([IOJobData](#) \*const data)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- `ActionJob` ([QObject](#) \*const parent=nullptr)
- `~ActionJob` () override

### Protected Member Functions

- void `run` () override

**Additional Inherited Members****Public Slots inherited from [Digikam::ActionJob](#)**

- void [cancel](#) ()

**Signals inherited from [Digikam::IOJob](#)**

- void **signalError** (const QString &errMsg)
- void **signalOneProcessed** (const QUrl &url)

**Signals inherited from [Digikam::ActionJob](#)**

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

**Public Attributes inherited from [Digikam::ActionJob](#)**

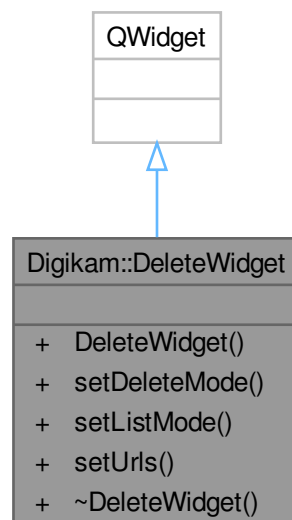
- QElapsedTimer [m\\_timer](#)

**Protected Attributes inherited from [Digikam::ActionJob](#)**

- bool [m\\_cancel](#) = false

## 6.367 [Digikam::DeleteWidget](#) Class Reference

Inheritance diagram for [Digikam::DeleteWidget](#):



### Public Member Functions

- **DeleteWidget** (QWidget \*const parent=nullptr)
- void **setDeleteMode** (DeleteDialogMode::DeleteMode deleteMode)
- void **setListMode** (DeleteDialogMode::ListMode mode)
- void **setUrls** (const QList< QUrl > &urls)

### Friends

- class **DeleteDialog**

## 6.368 Digikam::DeltaTime Class Reference

### Public Member Functions

- bool **isNull** () const

### Public Attributes

- int **deltaDays** = 0
- int **deltaHours** = 0
- int **deltaMinutes** = 0
- bool **deltaNegative** = false
- int **deltaSeconds** = 0

### 6.368.1 Detailed Description

Container that hold the time difference for clock photo dialog.

### 6.368.2 Member Function Documentation

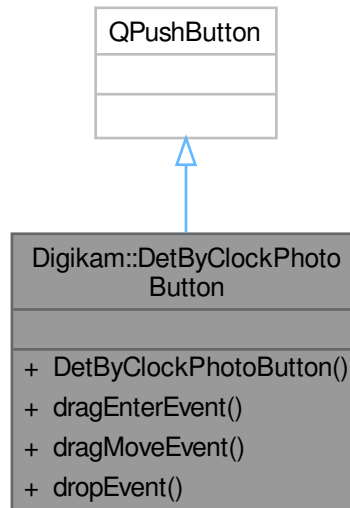
#### 6.368.2.1 isNull()

```
bool Digikam::DeltaTime::isNull ( ) const [inline]
```

Check if at least one option is selected.

## 6.369 Digikam::DetByClockPhotoButton Class Reference

Inheritance diagram for Digikam::DetByClockPhotoButton:



### Signals

- void **signalClockPhotoDropped** (const QUrl &)

### Public Member Functions

- **DetByClockPhotoButton** (const QString &text)
- void **dragEnterEvent** (QDragEnterEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override



## 6.370 Digikam::DetectionBenchmarker Class Reference

Inheritance diagram for Digikam::DetectionBenchmarker:



### Public Slots

- void **process** (const FacePipelineExtendedPackage::Ptr &package)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

## Signals

- void **processed** (const [FacePipelineExtendedPackage::Ptr](#) &package)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Public Member Functions

- **DetectionBenchmark** ([FacePipeline::Private](#) \*const d)
- [QString](#) **result** () const

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const char \*method, [Qt::](#)←  
ConnectionType type=[Qt::AutoConnection](#)) const
- [QThread::Priority](#) **priority** () const
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Protected Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- double **facePixels** = 0.0
- int **faces** = 0
- int **falseNegativeFaces** = 0
- int **falsePositiveFaces** = 0
- int **falsePositiveImages** = 0
- int **totalImages** = 0
- double **totalPixels** = 0.0
- int **trueNegativeImages** = 0
- int **truePositiveFaces** = 0

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

### 6.370.1 Member Function Documentation

#### 6.370.1.1 result()

```
QString Digikam::DetectionBenchmarker::result ( ) const
```

NOTE: Bench performance code. No need i18n here

## 6.371 Digikam::DetectionWorker Class Reference

Inheritance diagram for Digikam::DetectionWorker:



### Public Slots

- void **process** (const FacePipelineExtendedPackage::Ptr &package)
- void **setAccuracyAndModel** (int detectAccuracy, [FaceScanSettings::FaceDetectionModel](#) detectModel, [FaceScanSettings::FaceDetectionSize](#) detectSize, int recognizeAccuracy, [FaceScanSettings::FaceRecognitionModel](#) recognizeModel)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

## Signals

- void **processed** (const [FacePipelineExtendedPackage::Ptr](#) &package)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Public Member Functions

- [DetectionWorker](#) ([FacePipeline::Private](#) \*const dd)
- QImage **scaleForDetection** (const [DImg](#) &image) const

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const QObject \*sender, const char \*signal, const char \*method, Qt::ConnectionType type=Qt::AutoConnection) const
- QThread::Priority **priority** () const
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Protected Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- [FaceDetector](#) **detector**

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

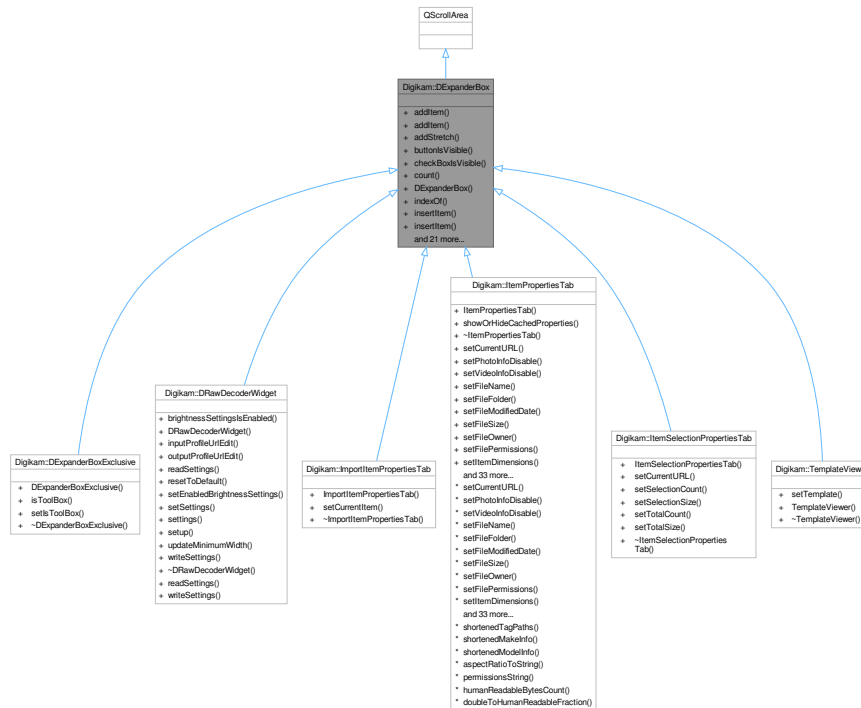
- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## 6.372 Digikam::DExpanderBox Class Reference

Inheritance diagram for Digikam::DExpanderBox:



## Signals

- void **signalItemButtonPressed** (int index)
- void **signalItemExpanded** (int index, bool b)
- void **signalItemToggled** (int index, bool b)

## Public Member Functions

- void [addItem](#) (QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **addItem** (QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **addStretch** ()
- bool **buttonIsVisible** (int index) const
- bool **checkboxIsVisible** (int index) const
- int **count** () const
- **DExpanderBox** (QWidget \*const parent=nullptr)
- int **indexOf** ([DLabelExpander](#) \*const widget) const
- void **insertItem** (int index, QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertItem** (int index, QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertStretch** (int index)
- bool **isChecked** (int index) const
- bool **isItemEnabled** (int index) const
- bool **isItemExpanded** (int index) const
- QIcon **itemIcon** (int index) const
- QString **itemText** (int index) const
- QString **itemToolTip** (int index) const
- virtual void **readSettings** (KConfigGroup &group)
- void **removeItem** (int index)
- void **setButtonIcon** (int index, const QIcon &icon)
- void **setButtonVisible** (int index, bool b)
- void **setCheckBoxVisible** (int index, bool b)
- void **setChecked** (int index, bool b)
- void **setItemEnabled** (int index, bool enabled)
- void **setItemExpanded** (int index, bool b)
- void **setItemIcon** (int index, const QIcon &icon)
- void **setItemText** (int index, const QString &txt)
- void **setItemToolTip** (int index, const QString &tip)
- [DLabelExpander](#) \* **widget** (int index) const
- virtual void **writeSettings** (KConfigGroup &group)

## 6.372.1 Member Function Documentation

### 6.372.1.1 addItem()

```
void Digikam::DExpanderBox::addItem (
    QWidget *const w,
    const QIcon & icon,
    const QString & txt,
    const QString & objName,
    bool expandBydefault )
```

Add [DLabelExpander](#) item at end of box layout with these settings : 'w' : the widget hosted by [DLabelExpander](#). 'pix' : pixmap used as icon to item title. 'txt' : text used as item title. 'objName' : item object name used to read/save expanded settings to rc file. 'expandBydefault' : item state by default (expanded or not).

### 6.372.1.2 insertItem()

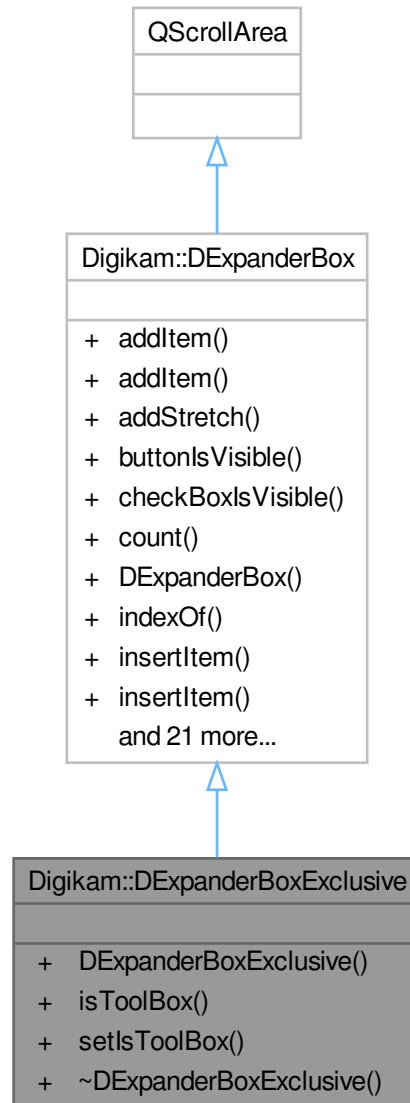
```
void Digikam::DExpanderBox::insertItem (
    int index,
    QWidget *const w,
    const QIcon & icon,
    const QString & txt,
    const QString & objName,
    bool expandBydefault )
```

Insert [DLabelExpander](#) item at box layout index with these settings : 'w' : the widget hosted by [DLabelExpander](#). 'pix' : pixmap used as icon to item title. 'txt' : text used as item title. 'objName' : item object name used to read/save expanded settings to rc file. 'expandBydefault' : item state by default (expanded or not).



## 6.373 Digikam::DExpanderBoxExclusive Class Reference

Inheritance diagram for Digikam::DExpanderBoxExclusive:



### Public Member Functions

- **DExpanderBoxExclusive** (`QWidget *const parent=nullptr`)
- `bool isToolBox () const`
- `void setIsToolBox (bool b)`

## Public Member Functions inherited from [Digikam::DExpanderBox](#)

- void [addItem](#) (QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void [addItem](#) (QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void [addStretch](#) ()
- bool [buttonIsVisible](#) (int index) const
- bool [checkboxIsVisible](#) (int index) const
- int [count](#) () const
- [DExpanderBox](#) (QWidget \*const parent=nullptr)
- int [indexOf](#) ([DLabelExpander](#) \*const widget) const
- void [insertItem](#) (int index, QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void [insertItem](#) (int index, QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void [insertStretch](#) (int index)
- bool [isChecked](#) (int index) const
- bool [isItemEnabled](#) (int index) const
- bool [isItemExpanded](#) (int index) const
- QIcon [itemIcon](#) (int index) const
- QString [itemText](#) (int index) const
- QString [itemToolTip](#) (int index) const
- virtual void [readSettings](#) (KConfigGroup &group)
- void [removeItem](#) (int index)
- void [setButtonIcon](#) (int index, const QIcon &icon)
- void [setButtonVisible](#) (int index, bool b)
- void [setCheckBoxVisible](#) (int index, bool b)
- void [setChecked](#) (int index, bool b)
- void [setItemEnabled](#) (int index, bool enabled)
- void [setItemExpanded](#) (int index, bool b)
- void [setItemIcon](#) (int index, const QIcon &icon)
- void [setItemText](#) (int index, const QString &txt)
- void [setItemToolTip](#) (int index, const QString &tip)
- [DLabelExpander](#) \* [widget](#) (int index) const
- virtual void [writeSettings](#) (KConfigGroup &group)

### Additional Inherited Members

## Signals inherited from [Digikam::DExpanderBox](#)

- void [signalItemButtonPressed](#) (int index)
- void [signalItemExpanded](#) (int index, bool b)
- void [signalItemToggled](#) (int index, bool b)

## 6.373.1 Member Function Documentation

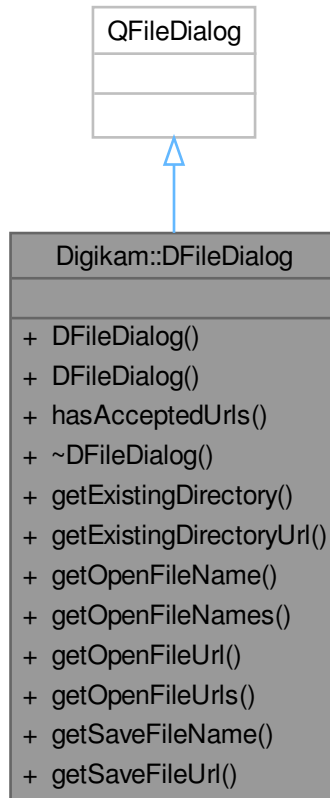
### 6.373.1.1 [setIsToolBox\(\)](#)

```
void Digikam::DExpanderBoxExclusive::setIsToolBox (
    bool b )
```

Show one expander open at most

## 6.374 Digikam::DFileDialog Class Reference

Inheritance diagram for Digikam::DFileDialog:



### Public Member Functions

- **DFileDialog** (`QWidget *const parent`, `Qt::WindowFlags flags`)
- **DFileDialog** (`QWidget *const parent=nullptr`, `const QString &caption=QString()`, `const QString &directory=QString()`, `const QString &filter=QString()`)
- `bool hasAcceptedUrls () const`

### Static Public Member Functions

- `static QString getExistingDirectory (QWidget *const parent=nullptr, const QString &caption=QString(), const QString &dir=QString(), Options options=ShowDirsOnly)`
- `static QUrl getExistingDirectoryUrl (QWidget *const parent=nullptr, const QString &caption=QString(), const QUrl &dir=QUrl(), Options options=ShowDirsOnly, const QStringList &supportedSchemes=QStringList())`
- `static QString getOpenFileName (QWidget *const parent=nullptr, const QString &caption=QString(), const QString &dir=QString(), const QString &filter=QString(), QString *selectedFilter=nullptr, Options options=Options())`

- static QStringList **getOpenFileNames** (QWidget \*const parent=nullptr, const QString &caption=QString(), const QString &dir=QString(), const QString &filter=QString(), QString \*selectedFilter=nullptr, Options options=Options())
- static QUrl **getOpenFileUrl** (QWidget \*const parent=nullptr, const QString &caption=QString(), const QUrl &dir=QUrl(), const QString &filter=QString(), QString \*selectedFilter=nullptr, Options options=Options(), const QStringList &supportedSchemes=QStringList())
- static QList< QUrl > **getOpenFileUrls** (QWidget \*const parent=nullptr, const QString &caption=QString(), const QUrl &dir=QUrl(), const QString &filter=QString(), QString \*selectedFilter=nullptr, Options options=Options(), const QStringList &supportedSchemes=QStringList())
- static QString **getSaveFileName** (QWidget \*const parent=nullptr, const QString &caption=QString(), const QString &dir=QString(), const QString &filter=QString(), QString \*selectedFilter=nullptr, Options options=Options())
- static QUrl **getSaveFileUrl** (QWidget \*const parent=nullptr, const QString &caption=QString(), const QUrl &dir=QUrl(), const QString &filter=QString(), QString \*selectedFilter=nullptr, Options options=Options(), const QStringList &supportedSchemes=QStringList())

## 6.375 Digikam::DFileOperations Class Reference

### Public Types

- enum **SidecarAction** { **Rename** = 0 , **Copy** }

### Static Public Member Functions

- static bool [copyFile](#) (const QString &srcFile, const QString &dstFile, const bool \*const cancel=nullptr)
- static bool [copyFiles](#) (const QStringList &srcPaths, const QString &dstPath)
- static bool [copyFolderRecursively](#) (const QString &srcPath, const QString &dstPath, const QString &item←Id=QString(), bool \*const cancel=nullptr, bool useDstPath=false)
- static bool [copyModificationTime](#) (const QString &srcFile, const QString &dstFile)
- static QString [findExecutable](#) (const QString &name, const QStringList &hints=QStringList())
- static QUrl [getUniqueFileUrl](#) (const QUrl &orgUrl, bool \*const newurl=nullptr)
- static QUrl [getUniqueFolderUrl](#) (const QUrl &orgUrl)
- static bool [localFileRename](#) (const QString &source, const QString &orgPath, const QString &destPath, bool ignoreSettings=false)
- static void [openFilesWithDefaultApplication](#) (const QList< QUrl > &urls)
- static void [openInFileManager](#) (const QList< QUrl > &urls)
- static bool [removeAndCopyFile](#) (const QString &srcFile, const QString &dstFile)
- static bool [renameFile](#) (const QString &srcFile, const QString &dstFile)
- static bool [setModificationTime](#) (const QString &srcFile, const QDateTime &dateTime)
- static bool [sidecarFiles](#) (const QString &srcFile, const QString &dstFile, SidecarAction action)

### 6.375.1 Member Function Documentation

#### 6.375.1.1 copyFile()

```
bool Digikam::DFileOperations::copyFile (
    const QString & srcFile,
    const QString & dstFile,
    const bool *const cancel = nullptr ) [static]
```

Copy file and keep the source file modification time.

### 6.375.1.2 copyFiles()

```
bool Digikam::DFileOperations::copyFiles (
    const QStringList & srcPaths,
    const QString & dstPath ) [static]
```

Copy a list of files to another place.

### 6.375.1.3 copyFolderRecursively()

```
bool Digikam::DFileOperations::copyFolderRecursively (
    const QString & srcPath,
    const QString & dstPath,
    const QString & itemId = QString(),
    bool *const cancel = nullptr,
    bool useDstPath = false ) [static]
```

Copy recursively a directory contents to another one.

### 6.375.1.4 copyModificationTime()

```
bool Digikam::DFileOperations::copyModificationTime (
    const QString & srcFile,
    const QString & dstFile ) [static]
```

Copy file modification time from source to destination file.

### 6.375.1.5 findExecutable()

```
QString Digikam::DFileOperations::findExecutable (
    const QString & name,
    const QStringList & hints = QStringList() ) [static]
```

Returns the path to a program under Windows by searching in the Windows registry. If the path is empty, `QStandardPaths::findExecutable()` is used as under Linux and macOS.

### 6.375.1.6 getUniqueFileUrl()

```
QUrl Digikam::DFileOperations::getUniqueFileUrl (
    const QUrl & orgUrl,
    bool *const newurl = nullptr ) [static]
```

Get unique file url if file exist by appending a counter suffix or return original url.

### 6.375.1.7 getUniqueFolderUrl()

```
QUrl Digikam::DFileOperations::getUniqueFolderUrl (
    const QUrl & orgUrl ) [static]
```

Get unique folder url if folder exist by appending a counter suffix or return original url.

### 6.375.1.8 localFileRename()

```
bool Digikam::DFileOperations::localFileRename (
    const QString & source,
    const QString & orgPath,
    const QString & destPath,
    bool ignoreSettings = false ) [static]
```

This method rename a local file 'orgPath' to 'destPath' with all ACL properties restoration taken from 'source' file. Return true if operation is completed.

### 6.375.1.9 openFilesWithDefaultApplication()

```
void Digikam::DFileOperations::openFilesWithDefaultApplication (
    const QList< QUrl > & urls ) [static]
```

Open file urls to default application relevant of file type-mimes desktop configuration.

### 6.375.1.10 openInFileManager()

```
void Digikam::DFileOperations::openInFileManager (
    const QList< QUrl > & urls ) [static]
```

Open system file manager and select the item.

### 6.375.1.11 removeAndCopyFile()

```
bool Digikam::DFileOperations::removeAndCopyFile (
    const QString & srcFile,
    const QString & dstFile ) [static]
```

If the destination file already exists, it will be removed. Copy file and keep the source file modification time.

### 6.375.1.12 renameFile()

```
bool Digikam::DFileOperations::renameFile (
    const QString & srcFile,
    const QString & dstFile ) [static]
```

Rename or move file and keep the source file modification time.

### 6.375.1.13 setModificationTime()

```
bool Digikam::DFileOperations::setModificationTime (
    const QString & srcFile,
    const QDateTime & dateTime ) [static]
```

Set file modification time from QDateTime. Keep access time from source file.

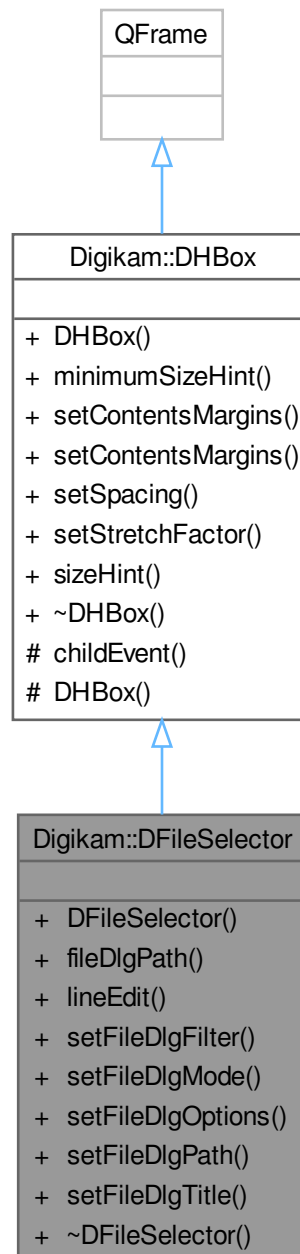
**6.375.1.14 sidecarFiles()**

```
bool Digikam::DFileOperations::sidecarFiles (
    const QString & srcFile,
    const QString & dstFile,
    SidecarAction action ) [static]
```

Rename/move or copy all possible sidecar files and keep the source file modification time.

## 6.376 Digikam::DFileSelector Class Reference

Inheritance diagram for Digikam::DFileSelector:



### Signals

- void **signalOpenFileDialog** ()
- void **signalUrlSelected** (const `QUrl` &)



## Public Member Functions

- **DFileSelector** (QWidget \*const parent=nullptr)
- QString **fileDlgPath** () const
- QLineEdit \* **lineEdit** () const
- void **setFileDlgFilter** (const QString &filter)
- void **setFileDlgMode** (QFileDialog::FileMode mode)
- void **setFileDlgOptions** (QFileDialog::Options opts)
- void **setFileDlgPath** (const QString &path)
- void **setFileDlgTitle** (const QString &title)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DHBox](#)

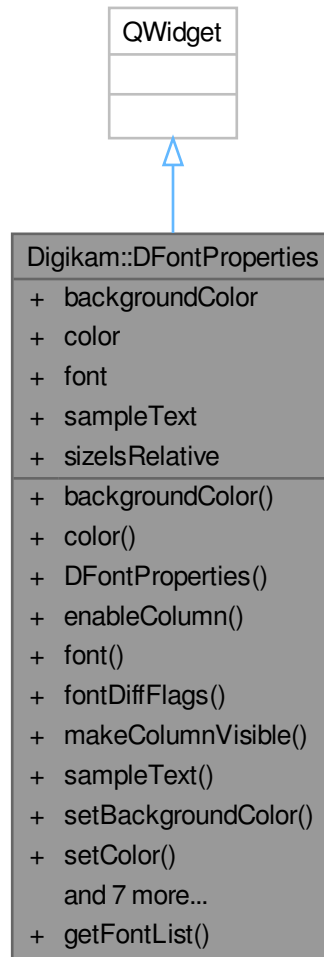
- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

### 6.376.1 Detailed Description

A widget to choose a single local file or path. Use line edit and file dialog properties to customize operation modes.

## 6.377 Digikam::DFontProperties Class Reference

Inheritance diagram for Digikam::DFontProperties:



### Public Types

- enum `DisplayFlag` { `NoDisplayFlags` = 0 , `FixedFontsOnly` = 1 , `DisplayFrame` = 2 , `ShowDifferences` = 4 }
- enum `FontColumn` { `FamilyList` = 0x01 , `StyleList` = 0x02 , `SizeList` = 0x04 }
- enum `FontDiff` { `NoFontDiffFlags` = 0 , `FontDiffFamily` = 1 , `FontDiffStyle` = 2 , `FontDiffSize` = 4 , `AllFontDiffs` = `FontDiffFamily` | `FontDiffStyle` | `FontDiffSize` }
- enum `FontListCriteria` { `FixedWidthFonts` = 0x01 , `ScalableFonts` = 0x02 , `SmoothScalableFonts` = 0x04 }

### Signals

- void `fontSelected` (const QFont &font)

## Public Member Functions

- QColor [backgroundColor](#) () const
- QColor [color](#) () const
- [DFontProperties](#) (QWidget \*const parent=nullptr, const DisplayFlags &flags=DisplayFrame, const QStringList &fontList=QStringList(), int visibleListSize=8, Qt::CheckState \*const sizelsRelativeState=nullptr)
- void [enableColumn](#) (int column, bool state)
- QFont [font](#) () const
- FontDiffFlags [fontDiffFlags](#) () const
- void [makeColumnVisible](#) (int column, bool state)
- QString [sampleText](#) () const
- void [setBackgroundColor](#) (const QColor &col)
- void [setColor](#) (const QColor &col)
- void [setFont](#) (const QFont &font, bool onlyFixed=false)
- void [setSampleBoxVisible](#) (bool visible)
- void [setSampleText](#) (const QString &text)
  - Sets the sample text.*
- void [setSizeRelative](#) (Qt::CheckState relative)
- QSize [sizeHint](#) (void) const override
- Qt::CheckState [sizelsRelative](#) () const
- [~DFontProperties](#) () override

## Static Public Member Functions

- static void [getFontList](#) (QStringList &list, uint fontListCriteria)

## Properties

- QColor **backgroundColor**
- QColor **color**
- QFont **font**
- QString **sampleText**
- Qt::CheckState **sizelsRelative**

## 6.377.1 Member Enumeration Documentation

### 6.377.1.1 DisplayFlag

enum [Digikam::DFontProperties::DisplayFlag](#)

- [FixedFontsOnly](#) only show fixed fonts, excluding proportional fonts
- [DisplayFrame](#) show a visual frame around the chooser
- [ShowDifferences](#) display the font differences interfaces

### 6.377.1.2 FontColumn

enum `Digikam::DFontProperties::FontColumn`

- `FamilyList` - Identifies the family (leftmost) list.
- `StyleList` - Identifies the style (center) list.
- `SizeList` - Identifies the size (rightmost) list.

### 6.377.1.3 FontDiff

enum `Digikam::DFontProperties::FontDiff`

- `FontDiffFamily` - Identifies a requested change in the font family.
- `FontDiffStyle` - Identifies a requested change in the font style.
- `FontDiffSize` - Identifies a requested change in the font size.

### 6.377.1.4 FontListCriteria

enum `Digikam::DFontProperties::FontListCriteria`

The selection criteria for the font families shown in the dialog.

- `FixedWidthFont` when included only fixed-width fonts are returned. The fonts where the width of every character is equal.
- `ScalableFont` when included only scalable fonts are returned; certain configurations allow bitmap fonts to remain unscaled and thus these fonts have limited number of sizes.
- `SmoothScalableFont` when included only return smooth scalable fonts. this will return only non-bitmap fonts which are scalable to any size requested. Setting this option to true will mean the "scalable" flag is irrelevant.

## 6.377.2 Constructor & Destructor Documentation

### 6.377.2.1 DFontProperties()

```
Digikam::DFontProperties::DFontProperties (
    QWidget *const parent = nullptr,
    const DisplayFlags & flags = DisplayFrame,
    const QStringList & fontList = QStringList(),
    int visibleListSize = 8,
    Qt::CheckState *const sizeIsRelativeState = nullptr ) [explicit]
```

Constructs a font picker widget. It normally comes up with all font families present on the system; the `getFont` method below does allow some more fine-tuning of the selection of fonts that will be displayed in the dialog.

## Parameters

<i>parent</i>	The parent widget.
<i>flags</i>	Defines how the font chooser is displayed.

## See also

DisplayFlags

## Parameters

<i>fontList</i>	A list of fonts to display, in XLFD format.
<i>visibleListSize</i>	The minimum number of visible entries in the fontlists.
<i>sizelsRelativeState</i>	If not zero the widget will show a checkbox where the user may choose whether the font size is to be interpreted as relative size. Initial state of this checkbox will be set according to *sizelsRelativeState, user choice may be retrieved by calling sizelsRelative().

**6.377.2.2 ~DFontProperties()**

```
Digikam::DFontProperties::~~DFontProperties ( ) [override]
```

Destructs the font chooser.

**6.377.3 Member Function Documentation****6.377.3.1 backgroundColor()**

```
QColor Digikam::DFontProperties::backgroundColor ( ) const
```

## Returns

The background color currently used in the preview (default: the base color of the active colorgroup)

**6.377.3.2 color()**

```
QColor Digikam::DFontProperties::color ( ) const
```

## Returns

The color currently used in the preview (default: the text color of the active color group)

**6.377.3.3 enableColumn()**

```
void Digikam::DFontProperties::enableColumn (
    int column,
    bool state )
```

Enables or disable a font column in the chooser.

Use this function if your application does not need or supports all font properties.

## Parameters

<i>column</i>	Specify the columns. An or'ed combination of <code>FamilyList</code> , <code>StyleList</code> and <code>SizeList</code> is possible.
<i>state</i>	If <code>false</code> the columns are disabled.

**6.377.3.4 font()**

```
QFont Digikam::DFontProperties::font ( ) const
```

## Returns

The currently selected font in the chooser.

**6.377.3.5 fontDiffFlags()**

```
DFontProperties::FontDiffFlags Digikam::DFontProperties::fontDiffFlags ( ) const
```

## Returns

The bitmask corresponding to the attributes the user wishes to change.

**6.377.3.6 fontSelected**

```
void Digikam::DFontProperties::fontSelected (
    const QFont & font ) [signal]
```

Emitted whenever the selected font changes.

**6.377.3.7 getFontList()**

```
void Digikam::DFontProperties::getFontList (
    QStringList & list,
    uint fontListCriteria ) [static]
```

Creates a list of font strings.

## Parameters

<i>list</i>	The list is returned here.
<i>fontListCriteria</i>	should contain all the restrictions for font selection as OR-ed values

## See also

[DFontProperties::FontListCriteria](#) for the individual values

### 6.377.3.8 makeColumnVisible()

```
void Digikam::DFontProperties::makeColumnVisible (
    int column,
    bool state )
```

Makes a font column in the chooser visible or invisible.

Use this function if your application does not need to show all font properties.

#### Parameters

<i>column</i>	Specify the columns. An or'ed combination of <code>FamilyList</code> , <code>StyleList</code> and <code>SizeList</code> is possible.
<i>state</i>	If <code>false</code> the columns are made invisible.

### 6.377.3.9 sampleText()

```
QString Digikam::DFontProperties::sampleText ( ) const
```

#### Returns

The current text in the sample text input area.

### 6.377.3.10 setBackgroundColor()

```
void Digikam::DFontProperties::setBackgroundColor (
    const QColor & col )
```

Sets the background color to use in the preview.

### 6.377.3.11 setColor()

```
void Digikam::DFontProperties::setColor (
    const QColor & col )
```

Sets the color to use in the preview.

### 6.377.3.12 setFont()

```
void Digikam::DFontProperties::setFont (
    const QFont & font,
    bool onlyFixed = false )
```

Sets the currently selected font in the chooser.

## Parameters

<i>font</i>	The font to select.
<i>onlyFixed</i>	Readjust the font list to display only fixed width fonts if <code>true</code> , or vice-versa.

**6.377.3.13 setSampleBoxVisible()**

```
void Digikam::DFontProperties::setSampleBoxVisible (
    bool visible )
```

Shows or hides the sample text box.

## Parameters

<i>visible</i>	Set it to true to show the box, to false to hide it.
----------------	--

**6.377.3.14 setSampleText()**

```
void Digikam::DFontProperties::setSampleText (
    const QString & text )
```

Normally you should not change this text, but it can be better to do this if the default text is too large for the edit area when using the default font of your application.

## Parameters

<i>text</i>	The new sample text. The current will be removed.
-------------	---

**6.377.3.15 setSizelsRelative()**

```
void Digikam::DFontProperties::setSizeIsRelative (
    Qt::CheckState relative )
```

Sets the state of the checkbox indicating whether the font size is to be interpreted as relative size.

## Note

If parameter `sizelsRelative` was not set in the constructor of the widget this setting will be ignored.

**6.377.3.16 sizeHint()**

```
QSize Digikam::DFontProperties::sizeHint (
    void ) const [override]
```

Reimplemented for internal reasons.



### 6.377.3.17 `sizeIsRelative()`

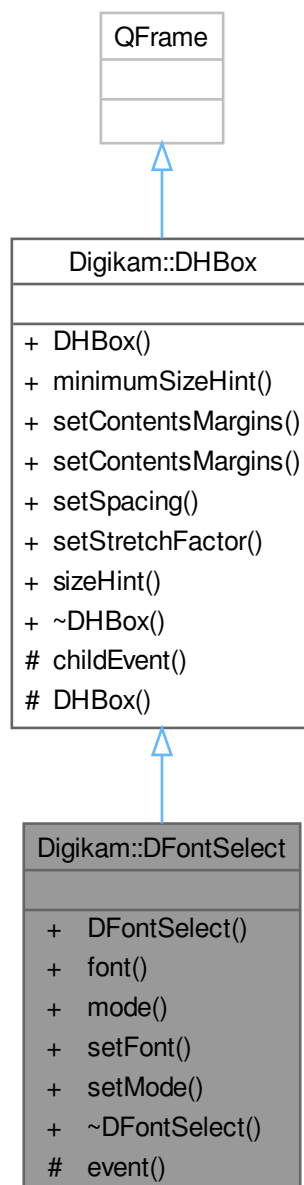
```
Qt::CheckState Digikam::DFontProperties::sizeIsRelative ( ) const
```

#### Returns

Whether the font size is to be interpreted as relative size (default: `QPushButton::Off`)

## 6.378 Digikam::DFontSelect Class Reference

Inheritance diagram for Digikam::DFontSelect:



## Public Types

- enum **FontMode** { **SystemFont** = 0 , **CustomFont** }

## Signals

- void **signalFontChanged** ()

## Public Member Functions

- **DFontSelect** (const QString &text, QWidget \*const parent=nullptr)
- QFont **font** () const
- FontMode **mode** () const
- void **setFont** (const QFont &font)
- void **setMode** (FontMode mode)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Protected Member Functions

- bool **event** (QEvent \*e) override

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.379 Digikam::DGradientSlider Class Reference

Inheritance diagram for Digikam::DGradientSlider:



### Public Slots

- void **setLeftValue** (double)
- void **setMiddleValue** (double)
- void **setRightValue** (double)

### Signals

- void **leftValueChanged** (double)
- void **middleValueChanged** (double)
- void **rightValueChanged** (double)

## Public Member Functions

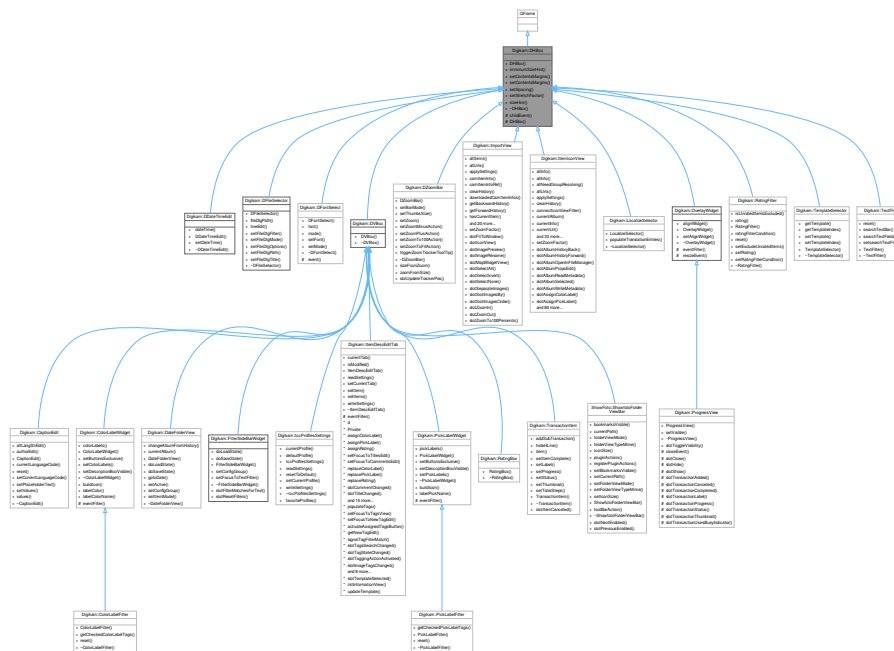
- **DGradientSlider** (QWidget \*const parent=nullptr)
- int **gradientOffset** () const
- double **leftValue** () const
- double **middleValue** () const
- double **rightValue** () const
- void **setColors** (const QColor &lcolor, const QColor &rcolor)
- void **showMiddleCursor** (bool b)

## Protected Member Functions

- void **leaveEvent** (QEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override

## 6.380 Digikam::DHBox Class Reference

Inheritance diagram for Digikam::DHBox:



## Public Member Functions

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

**Protected Member Functions**

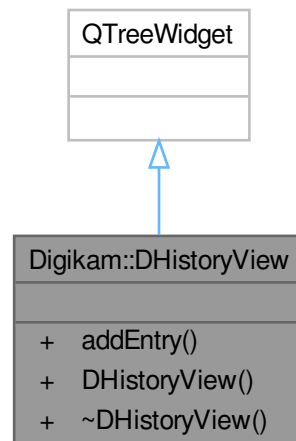
- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

**6.380.1 Detailed Description**

An Horizontal widget to host children widgets

**6.381 Digikam::DHistoryView Class Reference**

Inheritance diagram for Digikam::DHistoryView:

**Public Types**

- enum **EntryType** {  
**StartingEntry** = 0, **SuccessEntry**, **WarningEntry**, **ErrorEntry**,  
**ProgressEntry**, **CancelEntry** }

**Signals**

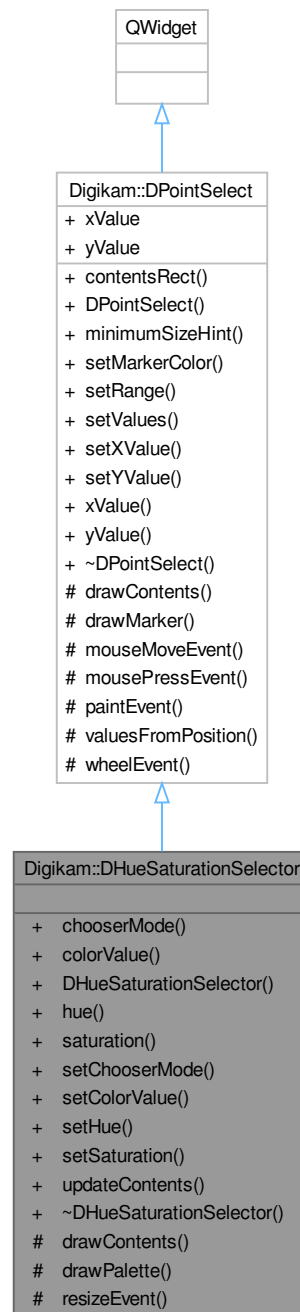
- void **signalEntryClicked** (const QVariant &metadata)

**Public Member Functions**

- void **addEntry** (const QString &msg, EntryType type, const QVariant &metadata=QVariant())
- **DHistoryView** (QWidget \*const parent)

## 6.382 Digikam::DHueSaturationSelector Class Reference

Inheritance diagram for Digikam::DHueSaturationSelector:



### Public Member Functions

- DColorChooserMode [chooserMode](#) () const
- int [colorValue](#) () const

- [DHueSaturationSelector](#) (QWidget \*const parent=nullptr)
- int [hue](#) () const
- int [saturation](#) () const
- void [setChooserMode](#) (DColorChooserMode [chooserMode](#))
- void [setColorValue](#) (int color)
- void [setHue](#) (int [hue](#))
- void [setSaturation](#) (int [saturation](#))
- void [updateContents](#) ()
- [~DHueSaturationSelector](#) () override

### Public Member Functions inherited from [Digikam::DPointSelect](#)

- QRect [contentsRect](#) () const
- [DPointSelect](#) (QWidget \*const parent)
- QSize [minimumSizeHint](#) () const override
- void [setMarkerColor](#) (const QColor &col)
- void [setRange](#) (int minX, int minY, int maxX, int maxY)
- void [setValues](#) (int xPos, int yPos)
- void [setXValue](#) (int xPos)
- void [setYValue](#) (int yPos)
- int [xValue](#) () const
- int [yValue](#) () const

### Protected Member Functions

- void [drawContents](#) (QPainter \*painter) override
- virtual void [drawPalette](#) (QPixmap \*pixmap)
- void [resizeEvent](#) (QResizeEvent \*) override

### Protected Member Functions inherited from [Digikam::DPointSelect](#)

- virtual void [drawMarker](#) (QPainter \*p, int xp, int yp)
- void [mouseMoveEvent](#) (QMouseEvent \*e) override
- void [mousePressEvent](#) (QMouseEvent \*e) override
- void [paintEvent](#) (QPaintEvent \*e) override
- void [valuesFromPosition](#) (int x, int y, int &xVal, int &yVal) const
- void [wheelEvent](#) (QWheelEvent \*) override

### Friends

- class [Private](#)

### Additional Inherited Members

### Signals inherited from [Digikam::DPointSelect](#)

- void [valueChanged](#) (int x, int y)

## Properties inherited from [Digikam::DPointSelect](#)

- int **xValue**
- int **yValue**

## 6.382.1 Constructor & Destructor Documentation

### 6.382.1.1 DHueSaturationSelector()

```
Digikam::DHueSaturationSelector::DHueSaturationSelector (
    QWidget *const parent = nullptr ) [explicit]
```

Constructs a hue/saturation selection widget.

### 6.382.1.2 ~DHueSaturationSelector()

```
Digikam::DHueSaturationSelector::~~DHueSaturationSelector ( ) [override]
```

Destructor.

## 6.382.2 Member Function Documentation

### 6.382.2.1 chooserMode()

```
DColorChooserMode Digikam::DHueSaturationSelector::chooserMode ( ) const
```

Returns the chooser mode.

#### Returns

The chooser mode (defined in DColorChooserMode)

### 6.382.2.2 colorValue()

```
int Digikam::DHueSaturationSelector::colorValue ( ) const
```

Returns the color value (also known as luminosity, 0-255)

#### Returns

The color value (0-255)



### 6.382.2.3 drawContents()

```
void Digikam::DHueSaturationSelector::drawContents (
    QPainter * painter ) [override], [protected], [virtual]
```

Reimplemented from [DPointSelect](#). This drawing is buffered in a pixmap here. As real drawing routine, [drawPalette\(\)](#) is used.

Reimplemented from [Digikam::DPointSelect](#).

### 6.382.2.4 drawPalette()

```
void Digikam::DHueSaturationSelector::drawPalette (
    QPixmap * pixmap ) [protected], [virtual]
```

Draws the contents of the widget on a pixmap, which is used for buffering.

### 6.382.2.5 hue()

```
int Digikam::DHueSaturationSelector::hue ( ) const
```

Returns the hue value

#### Returns

The hue value (0-360)

### 6.382.2.6 saturation()

```
int Digikam::DHueSaturationSelector::saturation ( ) const
```

Returns the saturation (0-255)

#### Returns

The saturation (0-255)

### 6.382.2.7 setChooserMode()

```
void Digikam::DHueSaturationSelector::setChooserMode (
    DColorChooserMode chooserMode )
```

Sets the chooser mode. The allowed modes are defined in [DColorChooserMode](#).

#### Parameters

<i>chooserMode</i>	The chooser mode as defined in <a href="#">DColorChooserMode</a>
--------------------	--

### 6.382.2.8 setColorValue()

```
void Digikam::DHueSaturationSelector::setColorValue (
    int color )
```

Sets the color value (0-255)

#### Parameters

<i>color</i>	The color value (0-255)
--------------	-------------------------

### 6.382.2.9 setHue()

```
void Digikam::DHueSaturationSelector::setHue (
    int hue )
```

Sets the hue value (0-360)

#### Parameters

<i>hue</i>	The hue value (0-360)
------------	-----------------------

### 6.382.2.10 setSaturation()

```
void Digikam::DHueSaturationSelector::setSaturation (
    int saturation )
```

Sets the saturation (0-255)

#### Parameters

<i>saturation</i>	The saturation (0-255)
-------------------	------------------------

### 6.382.2.11 updateContents()

```
void Digikam::DHueSaturationSelector::updateContents ( )
```

Updates the contents

## 6.383 Digikam::DigikamApp Class Reference

Inheritance diagram for Digikam::DigikamApp:



### Classes

- class [Private](#)

## Signals

- void **queuedOpenCameraUiFromPath** (const QString &path)
- void **queuedOpenSolidDevice** (const QString &udi)
- void **signalCopyAlbumItemsSelection** ()
- void **signalCutAlbumItemsSelection** ()
- void **signalEscapePressed** ()
- void **signalFirstItem** ()
- void **signalLastItem** ()
- void **signalNextItem** ()
- void **signalNotificationError** (const QString &message, int type)
- void **signalPasteAlbumItemsSelection** ()
- void **signalPrevItem** ()
- void **signalWindowHasMoved** ()

## Public Member Functions

- void **autoDetect** ()
- void **downloadFrom** (const QString &cameraGuiPath)
- void **downloadFromUdi** (const QString &udi)
- void **enableAlbumBackwardHistory** (bool enable)
- void **enableAlbumForwardHistory** (bool enable)
- void **enableZoomMinusAction** (bool val)
- void **enableZoomPlusAction** (bool val)
- [DInfoInterface](#) \* **infoface** ([DPluginAction](#) \*const ac) override
- void **restoreSession** ()
- virtual void **show** ()
- [ItemIconView](#) \* **view** () const

## Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- [QList](#)< [QAction](#) \* > **allActions** () const
- void **cleanupActions** ()
- [QString](#) **configGroupName** () const
- void **createFullscreenAction** (const [QString](#) &name)
- void **createHelpActions** (const [QString](#) &handbookSection, bool coreOptions=true)
- void **createSettingsActions** ()
- void **createSidebarActions** ()
- [DXmlGuiWindow](#) ([QWidget](#) \*const parent=nullptr, [Qt::WindowFlags](#) f=[Qt::WindowFlags](#)())
- bool **fullScreensIsActive** () const
- void **readFullscreenSettings** (const [KConfigGroup](#) &group)
- virtual void **registerExtraPluginsActions** ([QString](#) &)
- void **registerPluginsActions** ()
- void **setConfigGroupName** (const [QString](#) &name)
- void **setFullscreenOptions** (int options)
- void **unminimizeAndActivateWindow** ()

## Static Public Member Functions

- static [DigikamApp](#) \* **instance** ()

## Static Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- static QAction \* **buildStdAction** (StdActionType type, const QObject \*const recvr, const char \*const slot, QObject \*const parent)
- static QString **configFullScreenHideSideBarsEntry** ()
- static QString **configFullScreenHideStatusBarEntry** ()
- static QString **configFullScreenHideThumbBarEntry** ()
- static QString **configFullScreenHideToolBarsEntry** ()
- static void **restoreWindowSize** (QWindow \*const win, const KConfigGroup &group)
- static void **saveWindowSize** (QWindow \*const win, KConfigGroup &group)
- static void **setGoodDefaultWindowSize** (QWindow \*const win)
- static void **setupIconTheme** ()

## Protected Member Functions

- void **closeEvent** (QCloseEvent \*e) override
- void **moveEvent** (QMoveEvent \*e) override
- bool **queryClose** () override

## Protected Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- void **closeEvent** (QCloseEvent \*e) override
- void **editKeyboardShortcuts** (KActionCollection \*const extraac=nullptr, const QString &actitle=QString())
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override
- void **keyPressEvent** (QKeyEvent \*e) override
- QAction \* **showMenuBarAction** () const
- QAction \* **showStatusBarAction** () const

## Additional Inherited Members

## Protected Slots inherited from [Digikam::DXmlGuiWindow](#)

- bool **slotClose** ()

## Protected Attributes inherited from [Digikam::DXmlGuiWindow](#)

- [DLogoAction](#) \* **m\_animLogo** = nullptr

## 6.383.1 Member Function Documentation

### 6.383.1.1 infoiface()

```
DInfoInterface * Digikam::DigikamApp::infoIface (
    DPluginAction *const ac ) [override], [virtual]
```

Return the interface instance to access to items information.

Implements [Digikam::DXmlGuiWindow](#).

## 6.384 Digikam::DigikamApp::Private Class Reference

### Public Attributes

- QAction \* **addCameraSeparatorAction** = nullptr  
*Application Actions.*
- QAction \* **addFoldersAction** = nullptr
- QAction \* **addImagesAction** = nullptr
- QAction \* **advSearchAction** = nullptr
- KSelectAction \* **albumSortAction** = nullptr
- QAction \* **allGroupsOpenAction** = nullptr
- QAction \* **assignTagAction** = nullptr
- bool **autoShowZoomToolTip** = false
- KToolBarPopupAction \* **backwardActionMenu** = nullptr
- QAction \* **bqmAction** = nullptr  
*Tools Actions.*
- QAction \* **browseTagsAction** = nullptr  
*Tag Actions.*
- QHash< QString, QDateTime > **cameraAppearanceTimes**
- [CameraList](#) \* **cameraList** = nullptr
- QMenu \* **cameraMenu** = nullptr
- QMap< QString, QPointer< [ImportUI](#) > > **cameraUIMap**
- QMenu \* **cardReaderMenu** = nullptr
- KSharedConfig::Ptr **config**
- QAction \* **copyItemsAction** = nullptr
- QAction \* **copySelectionToAction** = nullptr
- QAction \* **cutItemsAction** = nullptr  
*Edit Actions.*
- QAction \* **deleteAction** = nullptr
- QAction \* **deleteTagAction** = nullptr
- QAction \* **editTagAction** = nullptr
- QEventLoop \* **eventLoop** = nullptr
- QActionGroup \* **exifOrientationActionGroup** = nullptr
- [FilterStatusBar](#) \* **filterStatusBar** = nullptr
- KToolBarPopupAction \* **forwardActionMenu** = nullptr
- QAction \* **ieAction** = nullptr
- QAction \* **imageAddCurrentQueueAction** = nullptr
- QAction \* **imageAddLightTableAction** = nullptr
- QAction \* **imageAddNewQueueAction** = nullptr
- QAction \* **imageAutoExifActionMenu** = nullptr
- QAction \* **imageDeleteAction** = nullptr
- QAction \* **imageDeletePermanentlyAction** = nullptr
- QAction \* **imageDeletePermanentlyDirectlyAction** = nullptr
- QMenu \* **imageExifOrientationActionMenu** = nullptr
- QAction \* **imageFindSimilarAction** = nullptr
- QMenu \* **imageFlipActionMenu** = nullptr
- QAction \* **imageIconViewAction** = nullptr
- QAction \* **imageLightTableAction** = nullptr
- QAction \* **imagePreviewAction** = nullptr
- QAction \* **imageReadMetadataAction** = nullptr
- QAction \* **imageRecognizeFacesAction** = nullptr
- QAction \* **imageRemoveAllFacesAction** = nullptr
- QAction \* **imageRenameAction** = nullptr

- QMenu \* **imageRotateActionMenu** = nullptr
- QAction \* **imageScanForFacesAction** = nullptr
- KSelectAction \* **imageSeparationAction** = nullptr
- KSelectAction \* **imageSeparationSortOrderAction** = nullptr
- QAction \* **imageSetExifOrientation1Action** = nullptr
- QAction \* **imageSetExifOrientation2Action** = nullptr
- QAction \* **imageSetExifOrientation3Action** = nullptr
- QAction \* **imageSetExifOrientation4Action** = nullptr
- QAction \* **imageSetExifOrientation5Action** = nullptr
- QAction \* **imageSetExifOrientation6Action** = nullptr
- QAction \* **imageSetExifOrientation7Action** = nullptr
- QAction \* **imageSetExifOrientation8Action** = nullptr
- KSelectAction \* **imageSortAction** = nullptr
- KSelectAction \* **imageSortOrderAction** = nullptr
- QAction \* **imageTableViewAction** = nullptr
- QAction \* **imageTrashDirectlyAction** = nullptr
- QAction \* **imageViewAction** = nullptr
- KSelectAction \* **imageViewSelectionAction** = nullptr

*Image Actions.*

- QAction \* **imageWriteMetadataAction** = nullptr
- QAction \* **ltAction** = nullptr
- QAction \* **maintenanceAction** = nullptr
- QActionGroup \* **manualCameraActionGroup** = nullptr
- [MetadataStatusBar](#) \* **metadataStatusBar** = nullptr
- [DModelFactory](#) \* **modelCollection** = nullptr
- QAction \* **moveSelectionToAlbumAction** = nullptr
- QAction \* **newAction** = nullptr

*Album Actions.*

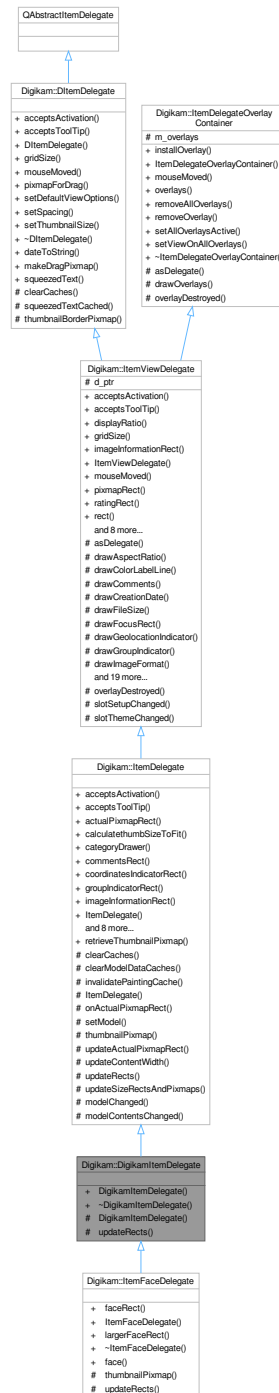
- QAction \* **newTagAction** = nullptr
- QAction \* **openInFileManagerAction** = nullptr
- QAction \* **openTagMngrAction** = nullptr
- QAction \* **openWithAction** = nullptr
- QAction \* **pasteItemsAction** = nullptr
- QAction \* **propsEditAction** = nullptr
- QAction \* **qualityAction** = nullptr
- QMenu \* **quickImportMenu** = nullptr
- QAction \* **quitAction** = nullptr
- QAction \* **readAlbumMetadataAction** = nullptr
- QAction \* **recurseAlbumsAction** = nullptr
- QAction \* **recurseTagsAction** = nullptr
- QAction \* **refreshAction** = nullptr
- QAction \* **renameAction** = nullptr
- QAction \* **scanNewItemAction** = nullptr
- QAction \* **selectAllAction** = nullptr
- QAction \* **selectInvertAction** = nullptr
- QAction \* **selectNoneAction** = nullptr
- QAction \* **showBarAction** = nullptr
- QActionGroup \* **solidCameraActionGroup** = nullptr
- QString **solidErrorMessage**
- QActionGroup \* **solidUsmActionGroup** = nullptr
- [DSplashScreen](#) \* **splashScreen** = nullptr
- [DAdjustableLabel](#) \* **statusLabel** = nullptr
- [TagsActionMngr](#) \* **tagsActionManager** = nullptr
- QAction \* **tipAction** = nullptr

- QMenu \* **usbMediaMenu** = nullptr
  - bool **validIccPath** = true
  - QIconView \* **view** = nullptr
  - QAction \* **viewCMViewAction** = nullptr
  - QAction \* **writeAlbumMetadataAction** = nullptr
  - DZoomBar \* **zoomBar** = nullptr
  - QAction \* **zoomFitToWindowAction** = nullptr
  - QAction \* **zoomMinusAction** = nullptr
  - QAction \* **zoomPlusAction** = nullptr
- View Actions.*
- QAction \* **zoomTo100percents** = nullptr



## 6.385 Digikam::DigikamItemDelegate Class Reference

Inheritance diagram for Digikam::DigikamItemDelegate:



### Public Member Functions

- `DigikamItemDelegate` (`ItemCategorizedView` \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect [actualPixmapRect](#) (const QModelIndex &index) const
- int [calculatethumbSizeToFit](#) (int ws)
- [ItemCategoryDrawer](#) \* [categoryDrawer](#) () const
- QRect [commentsRect](#) () const
- QRect [coordinatesIndicatorRect](#) () const
- QRect [groupIndicatorRect](#) () const
- QRect [imageInformationRect](#) () const override
- [ItemDelegate](#) (QWidget \*const parent)
- void [paint](#) (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setSpacing](#) (int spacing) override
- void [setView](#) ([ItemCategorizedView](#) \*view)
- QRect [tagsRect](#) () const

## Public Member Functions inherited from [Digikam::ItemViewDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double [displayRatio](#) () const
- QSize [gridSize](#) () const override
- [ItemViewDelegate](#) (QWidget \*const parent)
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect [ratingRect](#) () const
- QRect [rect](#) () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setRatingEdited](#) (const QModelIndex &index)
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int [spacing](#) () const
- [ThumbnailSize](#) [thumbnailSize](#) () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- [DItemDelegate](#) (QObject \*const parent=nullptr)

## Public Member Functions inherited from Digikam::ItemDelegateOverlayContainer

- void **installOverlay** (ItemDelegateOverlay \*overlay)
- ItemDelegateOverlayContainer ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< ItemDelegateOverlay \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** (ItemDelegateOverlay \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Protected Member Functions

- **DigikamItemDelegate** (DigikamItemDelegatePrivate &dd, ItemCategorizedView \*parent)
- void **updateRects** () override

## Protected Member Functions inherited from Digikam::ItemDelegate

- void **clearCaches** () override
- virtual void **clearModelDataCaches** ()
- void **invalidatePaintingCache** () override
- **ItemDelegate** (ItemDelegate::ItemDelegatePrivate &dd, QWidget \*const parent)
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- virtual void **updateContentWidth** ()
- void **updateSizeRectsAndPxmmaps** () override

## Protected Member Functions inherited from Digikam::ItemViewDelegate

- QAbstractItemDelegate \* **asDelegate** () override  
*Returns the delegate, typically, the derived class.*
- void **drawAspectRatio** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawColorLabelLine** (QPainter \*p, const QRect &pixRect, int colorId) const
- void **drawComments** (QPainter \*p, const QRect &commentsRect, const QString &comments) const
- void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
- void **drawGroupIndicator** (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
- void **drawImageFormat** (QPainter \*p, const QRect &r, const QString &f, bool drawTop) const
- void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawModificationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
- void **drawPanelSidelcon** (QPainter \*p, bool left, bool right) const
- void **drawPickLabelIcon** (QPainter \*p, const QRect &r, int pickLabel) const
- void **drawRating** (QPainter \*p, const QModelIndex &index, const QRect &ratingRect, int rating, bool is← Selected) const
- void **drawSpecialInfo** (QPainter \*p, const QRect &r, const QString &text) const

- void **drawTags** (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const
- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail, bool isGrouped) const
- void **drawTitle** (QPainter \*p, const QRect &titleRect, const QString &title) const
- **ItemViewDelegate** ([ItemViewDelegatePrivate](#) &dd, QWidget \*const parent)
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- void **prepareRatingPixmap** (bool composeOverBackground=true)
- QPixmap **ratingPixmap** (int rating, bool selected) const

### Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)

*Declare as slot in the derived class calling this method.*

### Additional Inherited Members

### Signals inherited from [Digikam::ItemViewDelegate](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

### Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

### Static Public Member Functions inherited from [Digikam::ItemDelegate](#)

- static QPixmap **retrieveThumbnailPixmap** (const QModelIndex &index, int thumbnailSize)

### Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

### Protected Slots inherited from [Digikam::ItemDelegate](#)

- void `modelChanged` ()
- void `modelContentsChanged` ()

### Protected Slots inherited from [Digikam::ItemViewDelegate](#)

- void `overlayDestroyed` (QObject \*o) override
- void `slotSetupChanged` ()
- void `slotThemeChanged` ()

### Protected Attributes inherited from [Digikam::ItemViewDelegate](#)

- [ItemViewDelegatePrivate](#) \*const `d_ptr` = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > `m_overlays`

## 6.385.1 Member Function Documentation

### 6.385.1.1 `updateRects()`

```
void Digikam::DigikamItemDelegate::updateRects ( ) [override], [protected], [virtual]
```

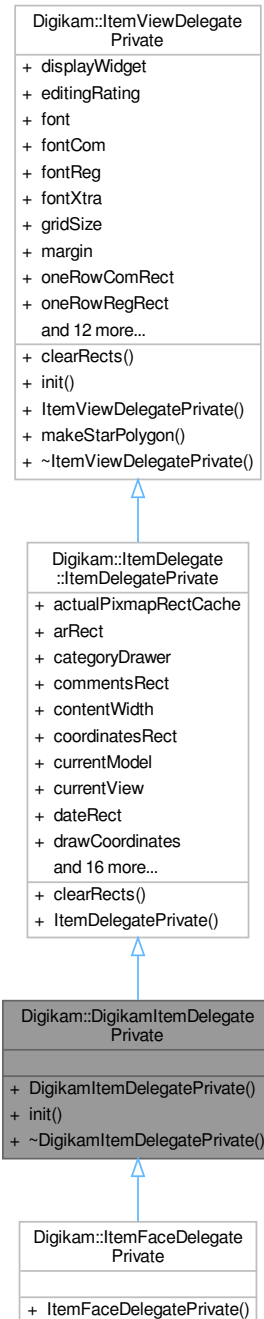
In a subclass, you need to implement this method to set up the rects for drawing. The `paint()` method operates depending on these rects.

Implements [Digikam::ItemDelegate](#).

Reimplemented in [Digikam::ItemFaceDelegate](#).

## 6.386 Digikam::DigikamItemDelegatePrivate Class Reference

Inheritance diagram for Digikam::DigikamItemDelegatePrivate:



### Public Member Functions

- void `init` (`DigikamItemDelegate` \*const q, `ItemCategorizedView` \*const parent)

## Public Member Functions inherited from Digikam::ItemDelegate::ItemDelegatePrivate

- void `clearRects` () override

## Public Member Functions inherited from Digikam::ItemViewDelegatePrivate

- void `init` (ItemViewDelegate \*const \_q, QWidget \*const \_widget)
- void `makeStarPolygon` ()

## Additional Inherited Members

## Public Attributes inherited from Digikam::ItemDelegate::ItemDelegatePrivate

- QCache< int, QRect > `actualPixmapRectCache`
- QRect `arRect`
- ItemCategoryDrawer \* `categoryDrawer`
- QRect `commentsRect`
- int `contentWidth`
- QRect `coordinatesRect`
- QAbstractItemModel \* `currentModel`
- ItemCategorizedView \* `currentView`
- QRect `dateRect`
- bool `drawCoordinates`
- bool `drawFocusFrame`
- bool `drawImageFormat`
- bool `drawImageFormatTop`
- bool `drawMouseOverFrame`
- QRect `groupRect`
- QRect `imageInformationRect`
- QRect `modDateRect`
- QRect `nameRect`
- QRect `pickLabelRect`
- QRect `pixmapRect`
- bool `ratingOverThumbnail`
- QRect `resolutionRect`
- QRect `sizeRect`
- QRect `specialInfoRect`
- QRect `tagRect`
- QRect `titleRect`

## Public Attributes inherited from Digikam::ItemViewDelegatePrivate

- QWidget \* `displayWidget` = nullptr
- QPersistentModelIndex `editingRating`
- QFont `font`
- QFont `fontCom`
- QFont `fontReg`
- QFont `fontXtra`
- QSize `gridSize`
- int `margin` = 5
- QRect `oneRowComRect`

- QRect **oneRowRegRect**
- QRect **oneRowXtraRect**
- [ItemViewDelegate](#) \* **q** = nullptr
- int **radius** = 3
  - constant values for drawing*
- QVector< QPixmap > **ratingPixmap** = QVector< QPixmap >(10)
- QRect **ratingRect**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- QPolygon **starPolygon**
- QSize **starPolygonSize**
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)





## Public Slots

- void **assignRating** (const QList< QModelIndex > &index, int rating)
- void **confirmFaces** (const QList< QModelIndex > &indexes, int tagId)
- void **deleteSelected** (const ItemViewUtilities::DeleteMode deleteMode=ItemViewUtilities::DeleteUseTrash)
- void **deleteSelectedDirectly** (const ItemViewUtilities::DeleteMode deleteMode=ItemViewUtilities::DeleteUseTrash)
- void **dragDropSort** (const ItemInfo &pick, const QList< ItemInfo > &infos)
- void **ignoreFaces** (const QList< QModelIndex > &indexes)
- void **openFile** (const ItemInfo &info)
- void **rejectFaces** (const QList< QModelIndex > &indexes)
- void **removeFaces** (const QList< QModelIndex > &indexes)
- void **rename** ()
- void **setFaceMode** (bool on)
- void **unknownFaces** (const QList< QModelIndex > &indexes)

## Public Slots inherited from [Digikam::ItemCategorizedView](#)

- void **hintAt** (const ItemInfo &info)
- void **openAlbum** (const QList< Album \* > &album)
- void **setCurrentInfo** (const ItemInfo &info)
- void **setCurrentUrl** (const QUrl &url)
- void **setCurrentUrlWhenAvailable** (const QUrl &url)
- void **setCurrentWhenAvailable** (qulonglong imageId)
- void **setSelectedItemInfos** (const QList< ItemInfo > &infos)
- void **setSelectedUrls** (const QList< QUrl > &urlList)
- void **setThumbnailSize** (int size)

## Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void **copy** () override
- void **cut** () override
- void **hideIndexNotification** ()
- void **paste** () override
- void **showIndexNotification** (const QModelIndex &index, const QString &message)

## Public Slots inherited from [Digikam::DCategorizedView](#)

- void **reset** () override

## Signals

- void **previewRequested** (const ItemInfo &info)
- void **signalSeparationModeChanged** (int category)
- void **signalShowContextMenu** (QContextMenuEvent \*event, const QList< QAction \* > &actions=QList< QAction \* >())
- void **signalShowContextMenuOnInfo** (QContextMenuEvent \*event, const ItemInfo &info, const QList< QAction \* > &actions, ItemFilterModel \*filterModel)
- void **signalShowGroupContextMenu** (QContextMenuEvent \*event, const QList< ItemInfo > &selectedInfos, ItemFilterModel \*filterModel)

## Signals inherited from [Digikam::ItemCategorizedView](#)

- void **currentChanged** (const [ItemInfo](#) &info)
- void **deselected** (const QList< [ItemInfo](#) > &nowDeselectedInfos)
 

*Emitted when items are deselected. There may be other selected infos left. This signal is not emitted when the model is reset; then only selectionCleared is emitted.*
- void **imageActivated** (const [ItemInfo](#) &info)
 

*Emitted when the given image is activated. Info is never null.*
- void **modelChanged** ()
 

*Emitted when a new model is set.*
- void **selected** (const QList< [ItemInfo](#) > &newSelectedInfos)
 

*Emitted when new items are selected. The parameter includes only the newly selected infos, there may be other already selected infos.*

## Signals inherited from [Digikam::ItemViewCategorized](#)

- void **clicked** (const QMouseEvent \*e, const QModelIndex &index)
- void **entered** (const QMouseEvent \*e, const QModelIndex &index)
- void **keyPressed** (QKeyEvent \*e)
- void **selectionChanged** ()
- void **selectionCleared** ()
- void **viewportClicked** (const QMouseEvent \*e)
- void **zoomInStep** ()
- void **zoomOutStep** ()

## Public Member Functions

- [ItemInfoList](#) **allItemInfos** (bool grouping=false) const
- bool **allNeedGroupResolving** (const [OperationType](#) type) const
- [DigikamItemView](#) (QWidget \*const parent=nullptr)
- int **fitToWidthIcons** ()
- QList< int > **getFaceIds** (const QList< QModelIndex > &indexes) const
- bool **getFaceMode** () const
- [ItemInfoList](#) **selectedItemInfos** (bool grouping=false) const
- [ItemInfoList](#) **selectedItemInfosCurrentFirst** (bool grouping=false) const
- bool **selectedNeedGroupResolving** (const [OperationType](#) type) const
- void **setThumbnailSize** (const [ThumbnailSize](#) &size) override
- [ItemViewUtilities](#) \* **utilities** () const

## Public Member Functions inherited from [Digikam::ItemCategorizedView](#)

- void **addOverlay** ([ItemDelegateOverlay](#) \*overlay, [ItemDelegate](#) \*delegate=nullptr)
 

*Add and remove an overlay. It will as well be removed automatically when destroyed. Unless you pass a different delegate, the current delegate will be used.*
- void **addSelectionOverlay** ([ItemDelegate](#) \*delegate=nullptr)
- [Album](#) \* **albumAt** (const QPoint &pos) const
- [ItemInfoList](#) **allItemInfos** () const
- QList< QUrl > **allUrls** () const
- [Album](#) \* **currentAlbum** () const
- [ItemInfo](#) **currentInfo** () const
- QUrl **currentUrl** () const
- [ItemDelegate](#) \* **delegate** () const

- `QItemSelectionModel * getSelectionModel () const`
- `ItemAlbumFilterModel * imageAlbumFilterModel () const`
- `ItemAlbumModel * imageAlbumModel () const`  
*Returns 0 if the `ItemModel` is not an `ItemAlbumModel`.*
- `ItemFilterModel * imageFilterModel () const`  
*Returns any `ItemFilterMode` in chain. May not be `sourceModel()`*
- `ItemModel * imageModel () const`
- `ImageSortFilterModel * imageSortFilterModel () const`
- `ItemThumbnailModel * imageThumbnailModel () const`  
*Returns 0 if the `ItemModel` is not an `ItemThumbnailModel`.*
- `QModelIndex indexForInfo (const ItemInfo &info) const`
- `ItemCategorizedView (QWidget *const parent=nullptr)`
- `ItemInfo nextInfo (const ItemInfo &info)`
- `ItemInfo nextInOrder (const ItemInfo &startingPoint, int nth)`
- `ItemInfo previousInfo (const ItemInfo &info)`
- `void removeOverlay (ItemDelegateOverlay *overlay)`
- `ItemInfoList selectedItemInfos () const`
- `ItemInfoList selectedItemInfosCurrentFirst () const`
- `void setModels (ItemModel *model, ImageSortFilterModel *filterModel)`
- `ThumbnailSize thumbnailSize () const`
- `void toIndex (const QUrl &url)`

## Public Member Functions inherited from `Digikam::ItemViewCategorized`

- `void awayFromSelection ()`
- `DItemDelegate * delegate () const`
- `void invertSelection ()`
- `bool isToolTipEnabled () const`
- `ItemViewCategorized (QWidget *const parent=nullptr)`
- `int numberOfSelectedIndexes () const`
- `void scrollTo (const QModelIndex &index, ScrollHint hint=EnsureVisible) override`
- `void scrollToRelaxed (const QModelIndex &index, ScrollHint hint=EnsureVisible)`
- `void setInitialSelectedItem (bool enabled)`
- `void setScrollCurrentToCenter (bool enabled)`
- `void setScrollStepGranularity (int factor)`
- `void setSelectedIndexes (const QList< QModelIndex > &indexes)`
- `void setSpacing (int spacing)`
- `void setToolTipEnabled (bool enabled)`
- `void setUsePointingHandCursor (bool useCursor)`
- `void toFirstIndex ()`
- `void toIndex (const QModelIndex &index)`
- `void toLastIndex ()`
- `void toNextIndex ()`
- `void toPreviousIndex ()`

## Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual QModelIndexList [categorizedIndexesIn](#) (const QRect &rect) const
- virtual QModelIndex [categoryAt](#) (const QPoint &point) const
- [DCategoryDrawer](#) \* [categoryDrawer](#) () const
- virtual QItemSelectionRange [categoryRange](#) (const QModelIndex &index) const
- virtual QRect [categoryVisualRect](#) (const QModelIndex &index) const
- [DCategorizedView](#) (QWidget \*const parent=nullptr)
- QModelIndex [indexAt](#) (const QPoint &point) const override
- void [setCategoryDrawer](#) ([DCategoryDrawer](#) \*categoryDrawer)
- void [setDrawDraggedItems](#) (bool drawDraggedItems)
- void [setGridSize](#) (const QSize &size)
- void [setModel](#) (QAbstractItemModel \*model) override
- QRect [visualRect](#) (const QModelIndex &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void [copy](#) ()
- virtual void [cut](#) ()
- virtual void [paste](#) ()

## Public Member Functions inherited from [Digikam::GroupingViewImplementation](#)

- [ItemInfoList](#) [getHiddenGroupedInfos](#) (const [ItemInfoList](#) &infos) const
- bool [needGroupResolving](#) ([OperationType](#) type, const [ItemInfoList](#) &infos) const
- [ItemInfoList](#) [resolveGrouping](#) (const [ItemInfoList](#) &infos) const

## Protected Slots

- void [groupIndicatorClicked](#) (const QModelIndex &index)
- void [showGroupContextMenu](#) (const QModelIndex &index, QContextMenuEvent \*event)

## Protected Slots inherited from [Digikam::ItemCategorizedView](#)

- void [slotCurrentUrlTimer](#) ()
- void [slotItemInfosAdded](#) ()

## Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void [layoutAboutToBeChanged](#) ()
- void [layoutWasChanged](#) ()
- void [slotActivated](#) (const QModelIndex &index)
- void [slotClicked](#) (const QModelIndex &index)
- void [slotEntered](#) (const QModelIndex &index)
- virtual void [slotThemeChanged](#) ()

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## Protected Member Functions

- void **activated** (const [ItemInfo](#) &info, Qt::KeyboardModifiers modifiers) override  
*Reimplement these in a subclass.*
- void **addAssignNameOverlay** ([ItemDelegate](#) \*delegate=nullptr)
- void **addRejectionOverlay** ([ItemDelegate](#) \*delegate=nullptr)
- bool **hasHiddenGroupedImages** (const [ItemInfo](#) &info) const override  
*must be implemented by parent view*
- [ItemInfoList](#) **imageInfos** (const QList< QModelIndex > &indexes, [OperationType](#) type) const
- void **showContextMenu** (QContextMenuEvent \*event) override
- void **showContextMenuOnInfo** (QContextMenuEvent \*event, const [ItemInfo](#) &info) override
- void **slotSetupChanged** () override

## Protected Member Functions inherited from [Digikam::ItemCategorizedView](#)

- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfoList](#) **imageInfos** (const QList< QModelIndex > &indexes) const
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- void **installDefaultModels** ()  
*install default [ItemAlbumModel](#) and filter model, ready for use*
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([ItemDelegate](#) \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override  
*Reimplement these in a subclass.*
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override

- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** (DItemDelegate \*delegate)
- void **setToolTip** (ItemViewToolTip \*tip)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

### Protected Member Functions inherited from Digikam::DCategorizedView

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

### Protected Member Functions inherited from Digikam::DragDropViewImplementation

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

## 6.387.1 Member Function Documentation

### 6.387.1.1 activated()

```
void Digikam::DigikamItemView::activated (
    const ItemInfo & info,
    Qt::KeyboardModifiers modifiers ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemCategorizedView](#).

### 6.387.1.2 confirmFaces

```
void Digikam::DigikamItemView::confirmFaces (
    const QList< QModelIndex > & indexes,
    int tagId ) [slot]
```

Confirm the face with a face tag (name) in the database. You aren't allowed to "confirm" a person as Ignored. Marking as Ignored is treated as a changeTag() operation.

### 6.387.1.3 hasHiddenGroupedImages()

```
bool Digikam::DigikamItemView::hasHiddenGroupedImages (
    const ItemInfo & ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::GroupingViewImplementation](#).

### 6.387.1.4 ignoreFaces

```
void Digikam::DigikamItemView::ignoreFaces (
    const QList< QModelIndex > & indexes ) [slot]
```

Set Face to Ignore ID.

### 6.387.1.5 rejectFaces

```
void Digikam::DigikamItemView::rejectFaces (
    const QList< QModelIndex > & indexes ) [slot]
```

This slot is connected to the reject signal of [AssignNameOverlay](#), and handles two cases.

If reject is done on an Unknown Face, it will mark the face as Ignored.

If reject is done on Unconfirmed suggestions, the suggestion is rejected and the face is marked as Unknown.

### 6.387.1.6 removeFaces

```
void Digikam::DigikamItemView::removeFaces (
    const QList< QModelIndex > & indexes ) [slot]
```

Removes the face from the database. You will have to run face detection again, to recover the face.

### 6.387.1.7 setThumbnailSize()

```
void Digikam::DigikamItemView::setThumbnailSize (
    const ThumbnailSize & size ) [override], [virtual]
```

Reimplemented from [Digikam::ItemCategorizedView](#).



### 6.387.1.8 showContextMenu()

```
void Digikam::DigikamItemView::showContextMenu (
    QContextMenuEvent * event ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

### 6.387.1.9 showContextMenuOnInfo()

```
void Digikam::DigikamItemView::showContextMenuOnInfo (
    QContextMenuEvent * event,
    const ItemInfo & info ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemCategorizedView](#).

### 6.387.1.10 slotSetupChanged()

```
void Digikam::DigikamItemView::slotSetupChanged ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

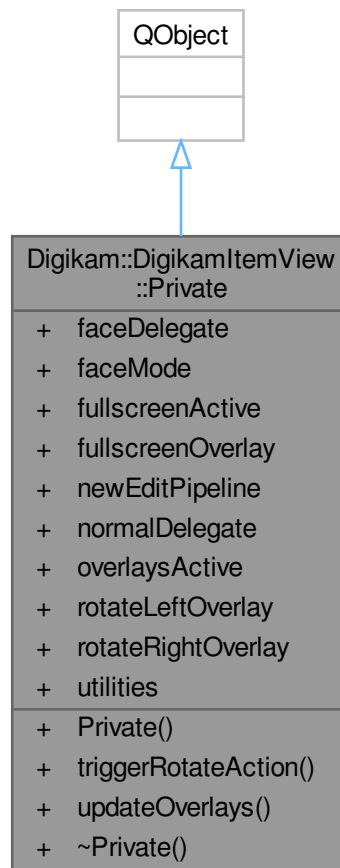
### 6.387.1.11 unknownFaces

```
void Digikam::DigikamItemView::unknownFaces (
    const QList< QModelIndex > & indexes ) [slot]
```

Ignored Face back to Unknown face.

## 6.388 Digikam::DigikamItemView::Private Class Reference

Inheritance diagram for Digikam::DigikamItemView::Private:



### Public Member Functions

- `Private` (`DigikamItemView *const qq`)
- void `triggerRotateAction` (`const char *actionName`)
- void `updateOverlays` ()

### Public Attributes

- `ItemFaceDelegate * faceDelegate` = nullptr
- bool `faceMode` = false
- bool `fullscreenActive` = false
- `ItemFullScreenOverlay * fullscreenOverlay` = nullptr
- `FacePipelineEdit * newEditPipeline` = nullptr
- `DigikamItemDelegate * normalDelegate` = nullptr
- bool `overlaysActive` = false
- `ItemRotateOverlay * rotateLeftOverlay` = nullptr
- `ItemRotateOverlay * rotateRightOverlay` = nullptr
- `ItemViewUtilities * utilities` = nullptr

## 6.389 Digikam::DImageHistory Class Reference

### Classes

- class [Entry](#)

### Public Member Functions

- const [FilterAction](#) & **action** (int i) const
- int **actionCount** () const  
*Returns the number of non-null actions.*
- void **adjustCurrentUuid** (const QString &uuid)  
*Changes the UUID of the current (last added current) referred image.*
- void **adjustReferredImages** ()
- QList< [FilterAction](#) > **allActions** () const  
*Gets all actions which are not null.*
- QList< [HistoryImageId](#) > **allReferredImages** () const
- void **appendReferredImage** (const [HistoryImageId](#) &id)
- void **clearReferredImages** ()  
*Remove all referredImages, leaving the entries list untouched.*
- [HistoryImageId](#) **currentReferredImage** () const
- DImageHistory** (const [DImageHistory](#) &other)
- QList< [DImageHistory::Entry](#) > & **entries** ()
- const QList< [DImageHistory::Entry](#) > & **entries** () const
- bool **hasActions** () const  
*Returns if there is any non-null action.*
- bool **hasCurrentReferredImage** () const
- bool **hasFilters** () const
- bool **hasOriginalReferredImage** () const
- bool **hasReferredImageOfType** ([HistoryImageId::Type](#) type) const
- bool **hasReferredImages** () const
- void **insertReferredImage** (int entryIndex, const [HistoryImageId](#) &id)
- bool **isEmpty** () const
- bool **isNull** () const
- bool **isValid** () const
- void **moveCurrentReferredImage** (const QString &newPath, const QString &newFileName)
- bool **operator!=** (const [DImageHistory](#) &other) const
- bool **operator<** (const [DImageHistory](#) &other) const
- [DImageHistory](#) & **operator<<** (const [FilterAction](#) &action)
- [DImageHistory](#) & **operator<<** (const [HistoryImageId](#) &imageId)
- [DImageHistory](#) & **operator=** (const [DImageHistory](#) &other)
- bool **operator==** (const [DImageHistory](#) &other) const
- bool **operator>** (const [DImageHistory](#) &other) const
- [Entry](#) & **operator[]** (int i)
- const [Entry](#) & **operator[]** (int i) const
- [HistoryImageId](#) **originalReferredImage** () const
- void **purgePathFromReferredImages** (const QString &path, const QString &fileName)
- QList< [HistoryImageId](#) > & **referredImages** (int i)
- const QList< [HistoryImageId](#) > & **referredImages** (int i) const
- QList< [HistoryImageId](#) > **referredImagesOfType** ([HistoryImageId::Type](#) type) const
- void **removeLast** ()  
*Removes the last entry from the history.*
- int **size** () const  
*Returns the number of entries.*
- QString **toXml** () const

## Static Public Member Functions

- static [DImageHistory](#) `fromXml` (const QString &xml)

## 6.389.1 Member Function Documentation

### 6.389.1.1 `adjustReferredImages()`

```
void Digikam::DImageHistory::adjustReferredImages ( )
```

Adjusts the type of a Current [HistoryImageId](#): If it is the first entry, it becomes Original, if it is in an intermediate entry, it becomes Intermediate, if in the last entry, it stays current.

### 6.389.1.2 `clearReferredImages()`

```
void Digikam::DImageHistory::clearReferredImages ( )
```

Edit referred images

### 6.389.1.3 `entries()`

```
QList< DImageHistory::Entry > & Digikam::DImageHistory::entries ( )
```

Access entries. There are [size\(\)](#) entries.

### 6.389.1.4 `hasActions()`

```
bool Digikam::DImageHistory::hasActions ( ) const
```

Access actions.

There is one action per entry, but the action may be null.

### 6.389.1.5 `isEmpty()`

```
bool Digikam::DImageHistory::isEmpty ( ) const
```

A history is considered empty if there are no entries.

### 6.389.1.6 `isNull()`

```
bool Digikam::DImageHistory::isNull ( ) const
```

A history is null if it is constructed with the default constructor

### 6.389.1.7 isValid()

```
bool Digikam::DImageHistory::isValid ( ) const
```

A history is a valid history (telling something about the past), if the history is not empty, and there is at least one referred image other than the "Current" entry, or there is a valid action.

### 6.389.1.8 moveCurrentReferredImage()

```
void Digikam::DImageHistory::moveCurrentReferredImage (
    const QString & newPath,
    const QString & newFileName )
```

Change file path entries of the current referred image

### 6.389.1.9 operator<<() [1/2]

```
DImageHistory & Digikam::DImageHistory::operator<< (
    const FilterAction & action )
```

Appends a new filter action to the history.

### 6.389.1.10 operator<<() [2/2]

```
DImageHistory & Digikam::DImageHistory::operator<< (
    const HistoryImageId & imageId )
```

Appends a new referred image, representing the current state of the history. If you add an id of type Current, [adjustReferredImages\(\)](#) will be called.

### 6.389.1.11 purgePathFromReferredImages()

```
void Digikam::DImageHistory::purgePathFromReferredImages (
    const QString & path,
    const QString & fileName )
```

Remove file path entries pointing to the given absolute path from any referred images. This is useful when said file is about to be overwritten. All other [HistoryImageId](#) fields remain unchanged, no [HistoryImageId](#) is removed. path: directory path, without filename.

### 6.389.1.12 referredImages()

```
QList< HistoryImageId > & Digikam::DImageHistory::referredImages (
    int i )
```

Access referred images

### 6.389.1.13 toXml()

```
QString Digikam::DImageHistory::toXml ( ) const
```

Serialize to and from XML.

Note: The "Current" entry is skipped when writing to XML, so make sure the file into the metadata of which you write the XML, is the file marked as "Current" in this history.

## 6.390 Digikam::DImageHistory::Entry Class Reference

### Public Attributes

- [FilterAction](#) `action`
- `QList< HistoryImageId >` `referredImages`

### 6.390.1 Member Data Documentation

#### 6.390.1.1 action

```
FilterAction Digikam::DImageHistory::Entry::action
```

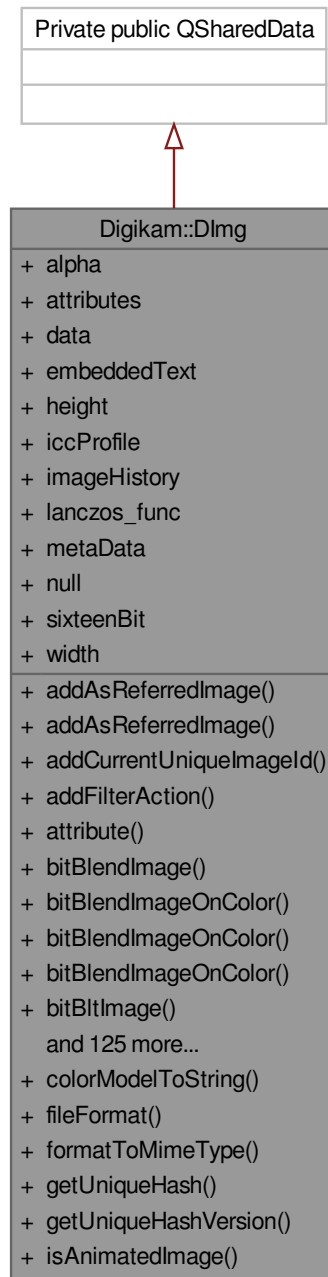
A [DImageHistory](#) is a list of entries.

Each entry has one action. The action can be null, but it shall be null only if it is the action of the first entry, with the "Original" as referred image, representing the action of digitization.

There can be zero, one or any number of referred images per entry. A referred image is a file in the state after the action is applied.

## 6.391 Digikam::DImg Class Reference

Inheritance diagram for Digikam::DImg:



### Public Types

- enum **ANGLE** { **ROT90** = 0 , **ROT180** , **ROT270** , **ROTNONE** }

- enum **COLORMODEL** {  
**COLORMODELUNKNOWN** = 0 , **RGB** , **GRAYSCALE** , **MONOCHROME** ,  
**INDEXED** , **YCBCR** , **CMYK** , **CIELAB** ,  
**COLORMODELRAW** }
- enum **FLIP** { **HORIZONTAL** = 0 , **VERTICAL** }
- enum **FORMAT** {  
**NONE** = 0 , **JPEG** , **PNG** , **TIFF** ,  
**JP2K** , **PGF** , **HEIF** , **RAW** ,  
**QIMAGE** }
- enum **PrepareMetadataFlag** {  
**RemoveOldMetadataPreviews** = 1 << 0 , **CreateNewMetadataPreview** = 1 << 1 , **ResetExifOrientationTag**  
= 1 << 2 , **CreateNewImageHistoryUUID** = 1 << 3 ,  
**PrepareMetadataFlagsAll** }

## Public Member Functions

- void **addAsReferredImage** (const [HistoryImageId](#) &id)
- [HistoryImageId](#) **addAsReferredImage** (const QString &filePath, [HistoryImageId::Type](#) type=[HistoryImageId::Intermediate](#))
- void **addCurrentUniqueImageId** (const QString &uuid)
- void **addFilterAction** (const [FilterAction](#) &action)
- QVariant **attribute** (const QString &key) const
- void **bitBlendImage** ([DColorComposer](#) \*const composer, const [DImg](#) \*const src, int sx, int sy, int w, int h, int dx, int dy, [DColorComposer::MultiplicationFlags](#) multiplicationFlags=[DColorComposer::NoMultiplication](#))
- void **bitBlendImageOnColor** (const [DColor](#) &color)
- void **bitBlendImageOnColor** (const [DColor](#) &color, int x, int y, int w, int h)
- void **bitBlendImageOnColor** ([DColorComposer](#) \*const composer, const [DColor](#) &color, int x, int y, int w, int h, [DColorComposer::MultiplicationFlags](#) multiplicationFlags=[DColorComposer::NoMultiplication](#))
- void **bitBitImage** (const [DImg](#) \*const src, int dx, int dy)
- void **bitBitImage** (const [DImg](#) \*const src, int sx, int sy, int dx, int dy)
- void **bitBitImage** (const [DImg](#) \*const src, int sx, int sy, int w, int h, int dx, int dy)
- void **bitBitImage** (const uchar \*const src, int sx, int sy, int w, int h, int dx, int dy, uint swidth, uint sheight, int sdepth)
- uchar \* **bits** () const
- int **bitsDepth** () const
- int **bytesDepth** () const
- void **convertDepth** (int depth)
- void **convertToDepthOfImage** (const [DImg](#) \*const otherImage)
- void **convertToEightBit** ()
- QPixmap **convertToPixmap** () const
- QPixmap **convertToPixmap** ([IccTransform](#) &monitorICctrans) const
- void **convertToSixteenBit** ()
- [DImg](#) **copy** () const
- [DImg](#) **copy** (const QRect &rect) const
- [DImg](#) **copy** (const QRectF &relativeRect) const
- [DImg](#) **copy** (int x, int y, int w, int h) const
- uchar \* **copyBits** () const
- [DImg](#) **copyImageData** () const
- [DImg](#) **copyMetaData** () const
- QImage **copyQImage** () const
- QImage **copyQImage** (const QRect &rect) const
- QImage **copyQImage** (const QRectF &relativeRect) const
- QImage **copyQImage** (int x, int y, int w, int h) const
- QImage **copyQImage32** () const
- [HistoryImageId](#) **createHistoryImageId** (const QString &filePath, [HistoryImageId::Type](#) type)



- QByteArray [createImageUniqueId](#) ()
- void [crop](#) (const QRect &rect)
- void [crop](#) (int x, int y, int w, int h)
- void [detach](#) ()
- [FORMAT](#) [detectedFormat](#) () const
- [DImg](#) ()
- [DImg](#) (const [DImg](#) &image)
- [DImg](#) (const QByteArray &filePath, [DImgLoaderObserver](#) \*const observer=nullptr, const [DRawDecoding](#) &[rawDecodingSettings](#)=[DRawDecoding](#)())
- [DImg](#) (const QImage &image)
- [DImg](#) (const QString &filePath, [DImgLoaderObserver](#) \*const observer=nullptr, const [DRawDecoding](#) &[rawDecodingSettings](#)=[DRawDecoding](#)())
- [DImg](#) (uint width, uint height, bool sixteenBit, bool alpha=false, uchar \*const data=nullptr, bool copy←Data=true)
- QString [embeddedText](#) (const QString &key) const
- int [exifOrientation](#) (const QString &filePath)
- bool [exifRotate](#) (const QString &filePath)
- QVariant [fileOriginData](#) () const
 

*When loaded from a file, some attributes like format and isReadOnly still depend on this originating file. When saving in a different format to a different file, you may wish to switch these attributes to the new file.*
- void [fill](#) (const [DColor](#) &color)
- void [flip](#) (FLIP direction)
- QString [format](#) () const
- [IccProfile](#) [getIccProfile](#) () const
- [DImageHistory](#) & [getItemHistory](#) ()
- const [DImageHistory](#) & [getItemHistory](#) () const
- [MetaEngineData](#) [getMetadata](#) () const
- [DImageHistory](#) [getOriginalImageHistory](#) () const
- [DColor](#) [getPixelColor](#) (uint x, uint y) const
- [DColor](#) [getSubPixelColor](#) (float x, float y) const
- [DColor](#) [getSubPixelColorFast](#) (float x, float y) const
- QByteArray [getUniqueHash](#) ()
- QByteArray [getUniqueHashVersion](#) (int version)
- bool [hasAlpha](#) () const
- bool [hasAttribute](#) (const QString &key) const
- bool [hasImageHistory](#) () const
- bool [hasTransparentPixels](#) () const
- uint [height](#) () const
- void [imageSavedAs](#) (const QString &savePath)
- void [insertAsReferredImage](#) (int afterHistoryStep, const [HistoryImageId](#) &otherImageId)
- bool [isNull](#) () const
- bool [isReadOnly](#) () const
- QVariant [lastSavedFileOriginData](#) () const
- QString [lastSavedFilePath](#) () const
- bool [load](#) (const QString &filePath, bool loadMetadata, bool loadICCDData, bool loadUniqueHash, bool load←History, [DImgLoaderObserver](#) \*const observer=nullptr, const [DRawDecoding](#) &[rawDecodingSettings](#)=[DRawDecoding](#)())
- bool [load](#) (const QString &filePath, [DImgLoaderObserver](#) \*const observer=nullptr, const [DRawDecoding](#) &[rawDecodingSettings](#)=[DRawDecoding](#)())
- bool [load](#) (const QString &filePath, int loadFlags, [DImgLoaderObserver](#) \*const observer, const [DRawDecoding](#) &[rawDecodingSettings](#)=[DRawDecoding](#)())
- bool [loadItemInfo](#) (const QString &filePath, bool loadMetadata=true, bool loadICCDData=true, bool load←UniqueHash=true, bool loadImageHistory=true)
- quint64 [numBytes](#) () const
- quint64 [numPixels](#) () const
- [DImg](#) & [operator=](#) (const [DImg](#) &image)

- bool `operator==` (const `DImg` &image) const
- int `orientation` () const
- int `originalBitDepth` () const
- COLORMODEL `originalColorModel` () const
- QString `originalFilePath` () const
- QSize `originalRatioSize` () const
- QSize `originalSize` () const
- void `prepareMetadataToSave` (const QString &intendedDestPath, const QString &destMimeType, bool resetExifOrientationTag)
- void `prepareMetadataToSave` (const QString &intendedDestPath, const QString &destMimeType, const QString &originalFileName=QString(), PrepareMetadataFlags flags=PrepareMetadataFlagsAll)
- void `prepareSubPixelAccess` ()
- **Private** ()=default
- QImage `pureColorMask` (`ExposureSettingsContainer` \*const expoSettings) const
- void `putImageData` (uchar \*const data, bool copyData=true)
- void `putImageData` (uint width, uint height, bool sixteenBit, bool alpha, uchar \*const data, bool copyData=true)
- `DRawDecoding` `rawDecodingSettings` () const
- void `removeAlphaChannel` ()
- void `removeAlphaChannel` (const `DColor` &destColor)
- void `removeAttribute` (const QString &key)
- void `reset` ()
- void `resetMetaData` ()
- void `resize` (int w, int h)
- bool `reverseExifRotate` (const QString &filePath)
- bool `reverseRotateAndFlip` (int `orientation`)
- void `rotate` (ANGLE angle)
- bool `rotateAndFlip` (int `orientation`)
- bool `save` (const QString &filePath, const QString &format, `DImgLoaderObserver` \*const observer=nullptr)
- bool `save` (const QString &filePath, `FORMAT` frm, `DImgLoaderObserver` \*const observer=nullptr)
- QString `savedFormat` () const
- uchar \* `scanLine` (uint i) const
- void `setAttribute` (const QString &key, const QVariant &value)
- void `setEmbeddedText` (const QString &key, const QString &text)
- void `setFileOriginData` (const QVariant &data)
- void `setHistoryBranch` (bool isBranch=true)
- void `setHistoryBranchAfter` (const `DImageHistory` &historyBeforeBranch, bool isBranch=true)
- void `setHistoryBranchForLastSteps` (int numberOfLastHistorySteps, bool isBranch=true)
- void `setIccProfile` (const `IccProfile` &profile)
- void `setItemHistory` (const `DImageHistory` &history)
- void `setMetadata` (const `MetaEngineData` &data)
- void `setPixelColor` (uint x, uint y, const `DColor` &color)
- bool `sixteenBit` () const
- QSize `size` () const
- `DImg` `smoothScale` (const QSize &destSize, Qt::AspectRatioMode aspectRatioMode=Qt::IgnoreAspectRatio) const
- `DImg` `smoothScale` (int width, int height, Qt::AspectRatioMode aspectRatioMode=Qt::IgnoreAspectRatio) const
- `DImg` `smoothScaleClipped` (const QSize &destSize, const QRect &clip, bool smooth=true) const
- `DImg` `smoothScaleClipped` (int width, int height, int clipx, int clipy, int clipwidth, int clipheight, bool smooth=true) const
- `DImg` `smoothScaleSection` (const QRect &sourceRect, const QSize &destSize) const
- `DImg` `smoothScaleSection` (int sx, int sy, int sw, int sh, int dw, int dh) const
- uchar \* `stripImageData` ()
- void `switchOriginToLastSaved` ()
- bool `transform` (int transformAction)
- bool `wasExifRotated` ()
- uint `width` () const

### Static Public Member Functions

- static QString [colorModelToString](#) (COLORMODEL colorModel)
- static [FORMAT](#) [fileFormat](#) (const QString &filePath)
- static QString [formatToMimeType](#) ([FORMAT](#) frm)
- static QByteArray [getUniqueHash](#) (const QString &filePath)
- static QByteArray [getUniqueHashVersion](#) (const QString &filePath, int version)
- static bool [isAnimatedImage](#) (const QString &filePath)

### Public Attributes

- bool **alpha** = false
- QMap< QString, QVariant > **attributes**
- unsigned char \* **data** = nullptr
- QMap< QString, QString > **embeddedText**
- unsigned int **height** = 0
- [IccProfile](#) **iccProfile**
- [DImageHistory](#) **imageHistory**
- LANCZOS\_DATA\_TYPE \* **lanczos\_func** = nullptr
- [MetaEngineData](#) **metaData**
- bool **null** = true
- bool **sixteenBit** = false
- unsigned int **width** = 0

### Friends

- class [DImgLoader](#)

## 6.391.1 Member Enumeration Documentation

### 6.391.1.1 FORMAT

```
enum Digikam::DImg::FORMAT
```

#### Enumerator

QIMAGE	QImage or ImageMagick.
--------	------------------------

### 6.391.1.2 PrepareMetadataFlag

```
enum Digikam::DImg::PrepareMetadataFlag
```

When saving, several changes to the image metadata are necessary before it can safely be written to the new file. This method updates the stored meta engine object in preparation to a subsequent call to `save()` with the same target file. 'intendedDestPath' is the finally intended file name. Do not give the temporary file name if you are going to `save()` to a temp file. 'destMimeType' is destination type mime. In some cases, metadata is updated depending on this value. 'originalFileName' is the original file's name, for simplistic history tracking in metadata. This is completely independent from the [DImageHistory](#) framework. For the 'flags' see below. Not all steps are optional and can be controlled with flags.

## Enumerator

RemoveOldMetadataPreviews	A small preview can be stored in the metadata. Remove old preview entries
CreateNewMetadataPreview	Create a new preview from current image data.
ResetExifOrientationTag	Set the exif orientation tag to "normal" Applicable if the image data was rotated according to the tag
CreateNewImageHistoryUUID	Creates a new UUID for the image history. Applicable if the file was changed.

## 6.391.2 Constructor & Destructor Documentation

### 6.391.2.1 DImg() [1/6]

```
DigiKam::DImg::DImg ( )
```

Create null image

[DImg](#) is a framework to support 16bits color depth image. it doesn't aim to be a complete imaging library; it uses QImage/ImageMagick for load/save files which are not supported natively by it. some of the features:

- Native Image Loaders, for some imageformats which are of interest to us: JPEG (complete), TIFF (mostly complete), PNG (complete), JPEG2000 (complete), RAW (complete through libraw), PGF (complete). For the rest ImageMAGick codecs or qimageloader are used.
- Metadata preservation: when a file is loaded, its metadata like XMP, IPTC, EXIF, JFIF are read and held in memory. now when you save back the file to the original file or to a different file, the metadata is automatically written. All is delegate to Exiv2 library.
- Explicitly Shared Container format (see qt docs): this is necessary for performance reasons.
- 8 bits and 16 bits support: if the file format is 16 bits, it will load up the image in 16bits format (TIFF/PNG/↔ JPEG2000/RAW/PGF support) and all operations are done in 16 bits format, except when the rendering to screen is done, when its converted on the fly to a temporary 8 bits image and then rendered.
- Basic image manipulation: rotate, flip, color modifications, crop, scale. This has been ported from Imlib2 with 16 bits scaling support and support for scaling of only a section of the image.
- Rendering to Pixmap: using QImage/QPixmap. (see above for rendering of 16 bits images).
- Pixel format: the pixel format is different from QImage pixel format. In QImage the pixel data is stored as unsigned ints and to access the individual colors you need to use bit-shifting to ensure endian correctness. in [DImg](#), the pixel data is stored as unsigned char. the color layout is B,G,R,A (blue, green, red, alpha)

for 8 bits images: you can access individual color components like this:

```
uchar* const pixels = image.bits();
```

```
for (int i = 0 ; i < image.width() * image.height() ; ++i) { pixel[0] // blue pixel[1] // green pixel[2] // red pixel[3] // alpha
pixel += 4; // go to next pixel }
```

and for 16 bits images:

```
ushort* const pixels = (ushort*)image.bits();
```

```
for (int i = 0 ; i < image.width() * image.height() ; ++i) { pixel[0] // blue pixel[1] // green pixel[2] // red pixel[3] // alpha
pixel += 4; // go to next pixel }
```

The above is true for both big and little endian platforms. What this also means is that the pixel format is different from that of QImage for big endian machines. Functions are provided if you want to get a copy of the [DImg](#) as a QImage.

### 6.391.2.2 DImg() [2/6]

```
Digikam::DImg::DImg (
    const QByteArray & filePath,
    DImgLoaderObserver *const observer = nullptr,
    const DRawDecoding & rawDecodingSettings = DRawDecoding() ) [explicit]
```

Load image using QByteArray as file path

### 6.391.2.3 DImg() [3/6]

```
Digikam::DImg::DImg (
    const QString & filePath,
    DImgLoaderObserver *const observer = nullptr,
    const DRawDecoding & rawDecodingSettings = DRawDecoding() ) [explicit]
```

Load image using QString as file path

### 6.391.2.4 DImg() [4/6]

```
Digikam::DImg::DImg (
    const DImg & image )
```

Copy image: Creates a shallow copy that refers to the same shared data. The two images will be equal. Call [detach\(\)](#) or [copy\(\)](#) to create deep copies.

### 6.391.2.5 DImg() [5/6]

```
Digikam::DImg::DImg (
    const QImage & image ) [explicit]
```

Copy image: Creates a copy of a QImage object. If the QImage is null, a null [DImg](#) will be created.

### 6.391.2.6 DImg() [6/6]

```
Digikam::DImg::DImg (
    uint width,
    uint height,
    bool sixteenBit,
    bool alpha = false,
    uchar *const data = nullptr,
    bool copyData = true )
```

Create image from data. If data is 0, a new buffer will be allocated, otherwise the given data will be used: If copydata is true, the data will be copied to a newly allocated buffer. If copyData is false, this [DImg](#) object will take ownership of the data pointer. If there is an alpha channel, the data shall be in non-premultiplied form (unassociated alpha).

## 6.391.3 Member Function Documentation

### 6.391.3.1 addAsReferredImage()

```
HistoryImageId Digikam::DImg::addAsReferredImage (
    const QString & filePath,
    HistoryImageId::Type type = HistoryImageId::Intermediate )
```

If you have saved this [DImg](#) to filePath, and want to continue using this [DImg](#) object to add further changes to the image history, you can call this method to add to the image history a reference to the just saved image. First call [updateMetadata\(\)](#), then call [save\(\)](#), then call [addAsReferredImage\(\)](#). Do not call this directly after loading, before applying any changes: The history is correctly initialized when loading. If you need to insert the referred file to an entry which is not the last entry, which may happen if the added image was saved after this image's history was created, you can use [insertAsReferredImage](#). The added id is returned.

### 6.391.3.2 addCurrentUniqueImageId()

```
void Digikam::DImg::addCurrentUniqueImageId (
    const QString & uuid )
```

In the history, adjusts the UUID of the ImageHistoryId of the current file. Call this if you have associated a UUID with this file which is not written to the metadata. If there is already a UUID present, read from metadata, it will not be replaced.

### 6.391.3.3 bitBlendImage()

```
void Digikam::DImg::bitBlendImage (
    DColorComposer *const composer,
    const DImg *const src,
    int sx,
    int sy,
    int w,
    int h,
    int dx,
    int dy,
    DColorComposer::MultiplicationFlags multiplicationFlags = DColorComposer::NoMultiplication
)
```

Blend src image on this image (this is dest) with the specified composer and multiplication flags. See documentation of [DColorComposer](#) for more info. For the other arguments, see documentation of [bitBlendImage](#) above.

### 6.391.3.4 bitBlendImageOnColor()

```
void Digikam::DImg::bitBlendImageOnColor (
    DColorComposer *const composer,
    const DColor & color,
    int x,
    int y,
    int w,
    int h,
    DColorComposer::MultiplicationFlags multiplicationFlags = DColorComposer::NoMultiplication
)
```

For the specified region, blend this image on the given color with the specified composer and multiplication flags. See documentation of [DColorComposer](#) for more info. Note that the result pixel is again written to this image, which is, for the blending, source.

### 6.391.3.5 bitBltImage()

```
void Digikam::DImg::bitBltImage (
    const DImg *const src,
    int dx,
    int dy )
```

Copy a region of pixels from a source image to this image. Parameters: sx|sy Coordinates in the source image of the rectangle to be copied w h Width and height of the rectangle (Default, or when both are -1: whole source image) dx|dy Coordinates in this image of the rectangle in which the region will be copied (Default: 0|0) The bit depth of source and destination must be identical.

### 6.391.3.6 bitsDepth()

```
int Digikam::DImg::bitsDepth ( ) const
```

Return the number of bits depth of one color component for one pixel : 8 (non sixteenBit) or 16 (sixteen)

### 6.391.3.7 bytesDepth()

```
int Digikam::DImg::bytesDepth ( ) const
```

Return the number of bytes depth of one pixel : 4 (non sixteenBit) or 8 (sixteen)

### 6.391.3.8 colorModelToString()

```
QString Digikam::DImg::colorModelToString (
    COLORMODEL colorModel ) [static]
```

Helper method to translate enum values to user presentable strings

### 6.391.3.9 convertDepth()

```
void Digikam::DImg::convertDepth (
    int depth )
```

Convert depth of image. Depth is bytesDepth \* bitsDepth. If depth is 32, converts to 8 bits, if depth is 64, converts to 16 bits.

### 6.391.3.10 convertToSixteenBit()

```
void Digikam::DImg::convertToSixteenBit ( )
```

Wrapper methods for convertDepth

### 6.391.3.11 copy() [1/2]

```
DImg Digikam::DImg::copy ( ) const
```

Return a deep copy of full image

### 6.391.3.12 copy() [2/2]

```
DImg Digikam::DImg::copy (
    const QRect & rect ) const
```

Return a region of image

### 6.391.3.13 copyImageData()

```
DImg Digikam::DImg::copyImageData ( ) const
```

Return a deep copy of the image, but do not include metadata.

### 6.391.3.14 copyMetaData()

```
DImg Digikam::DImg::copyMetaData ( ) const
```

Return an image that contains a deep copy of this image's metadata and the information associated with the image data (width, height, hasAlpha, sixteenBit), but no image data, i.e. `isNull()` is true.

### 6.391.3.15 copyQImage()

```
QImage Digikam::DImg::copyQImage ( ) const
```

QImage wrapper methods

### 6.391.3.16 createHistoryImageId()

```
HistoryImageId Digikam::DImg::createHistoryImageId (
    const QString & filePath,
    HistoryImageId::Type type )
```

Create a [HistoryImageId](#) for *this* image *already* saved at the given file path.

### 6.391.3.17 createImageUniqueId()

```
QByteArray Digikam::DImg::createImageUniqueId ( )
```

This method creates a new 256-bit UUID meant to be globally unique. The UUID will be returned as a 64-byte hexadecimal string. At least 128bits of the UUID will be created by the platform random number generator. The rest may be created from a content-based hash similar to the `uniqueHash`, see above. This method only generates a new UUID for this image without in any way changing this image object or saving the UUID anywhere.



### 6.391.3.18 crop()

```
void Digikam::DImg::crop (
    const QRect & rect )
```

Crop image to the specified region

### 6.391.3.19 detach()

```
void Digikam::DImg::detach ( )
```

Detaches from shared data and makes sure that this image is the only one referring to the data. If multiple images share common data, this image makes a copy of the data and detaches itself from the sharing mechanism. Nothing is done if there is just a single reference.

### 6.391.3.20 detectedFormat()

```
DImg::FORMAT Digikam::DImg::detectedFormat ( ) const
```

Returns the file format in form of the FORMAT enum that was detected in the load() method. Other than the format attribute which is written by the [DImgLoader](#), this can include the QIMAGE or NONE values. Returns NONE for images that have not been loaded. For unknown image formats, a value of QIMAGE can be returned to indicate that the QImage-based loader will have been used. To find out if this has worked, check the return value you got from load().

### 6.391.3.21 exifOrientation()

```
int Digikam::DImg::exifOrientation (
    const QString & filePath )
```

Retrieves the Exif orientation, either from the [LoadSaveThread](#) info provider if available, or from the metadata

### 6.391.3.22 fileFormat()

```
DImg::FORMAT Digikam::DImg::fileFormat (
    const QString & filePath ) [static]
```

Identify file format

### 6.391.3.23 fileOriginData()

```
QVariant Digikam::DImg::fileOriginData ( ) const
```

#### See also

[fileOriginData](#) returns the current origin data, bundled in the returned QVariant.  
[setFileOriginData](#) takes such a variant and adjusts the properties.  
[lastSavedFileOriginData](#) returns the origin data as if the image was loaded from the last saved image.  
[switchOriginToLastSaved](#) is equivalent to setting origin data returned from [lastSavedFileOriginData](#).

For example, an image loaded from a RAW and saved to PNG will be read-only and format RAW. After calling

#### See also

[switchOriginToLastSaved](#), it will not be read-only, [format](#) will be PNG, and [rawDecodingSettings](#) will be null.  
[detectedFormat](#) will not change. In the history, the last referred image that was added (as intermediate) is made the new Current image.

#### Note

Set the saved image path with

#### See also

[imageSavedAs](#) before!

### 6.391.3.24 fill()

```
void Digikam::DImg::fill (
    const DColor & color )
```

Fill whole image with specified color. The bit depth of the color must be identical to the depth of this image.

### 6.391.3.25 format()

```
QString Digikam::DImg::format ( ) const
```

Returns the format string as written by the image loader this image was originally loaded from. Format strings used include JPEG, PNG, TIFF, PGF, JP2K, RAW, PPM. For images loaded with the platform QImage loader, the file suffix is used. Returns null if this [DImg](#) was not loaded from a file, but created in memory.

### 6.391.3.26 getMetadata()

```
MetaEngineData Digikam::DImg::getMetadata ( ) const
```

Metadata manipulation methods

### 6.391.3.27 getPixelColor()

```
DColor Digikam::DImg::getPixelColor (
    uint x,
    uint y ) const
```

Access a single pixel of the image. These functions add some safety checks and then use the methods from [DColor](#). In optimized code working directly on the data, better use the inline methods from [DColor](#).

### 6.391.3.28 getUniqueHash()

```
QByteArray Digikam::DImg::getUniqueHash ( )
```

This methods return a 128-bit MD5 hex digest which is meant to uniquely identify the file. The hash is calculated on parts of the file and the file metadata. It cannot be used to find similar images. It is not calculated from the image data. The hash will be returned as a 32-byte hexadecimal string.

If you already have a [DImg](#) object of the file, use the member method. The object does not need to have the full image data loaded, but it shall at least have been loaded with `loadItemInfo` with `loadMetadata = true`, or have the metadata set later with `setComments`, `setExif`, `setIptc`, `setXmp`. If the object does not have the metadata loaded, a non-null, but invalid hash will be returned! In this case, use the static method. If the image has been loaded with `loadUniqueHash = true`, the hash can be retrieved with the member method.

You do not need a [DImg](#) object of the file to retrieve the unique hash; Use the static method and pass just the file path.

### 6.391.3.29 getUniqueHashVersion()

```
QByteArray Digikam::DImg::getUniqueHashVersion (
    int version )
```

Version 2: This methods return a 128-bit MD5 hex digest which is meant to uniquely identify the file. The hash is calculated on parts of the file. It cannot be used to find similar images. It is not calculated from the image data. The hash will be returned as a 32-byte hexadecimal string.

Version 3: This methods return a 128-bit MD5 hex digest which is meant to uniquely identify the file. It cannot be used to find similar images. The hash is calculated from 6 blocks distributed across the file, the first block has a size of 100 kB (capture metadata), all other possible 5 blocks up to 25 kB. The hash will be returned as a 32-byte hexadecimal string.

If you already have a [DImg](#) object loaded from the file, use the member method. If the image has been loaded with `loadUniqueHash = true`, the hash will already be available.

You do not need a [DImg](#) object of the file to retrieve the unique hash; Use the static method and pass just the file path and version.

### 6.391.3.30 hasTransparentPixels()

```
bool Digikam::DImg::hasTransparentPixels ( ) const
```

If the image has an alpha channel, check if there exist pixels which actually have non-opaque color, that is `alpha < 1.0`. Note that all pixels are scanned to reach a return value of "false". If `hasAlpha()` is false, always returns false.

### 6.391.3.31 imageSavedAs()

```
void Digikam::DImg::imageSavedAs (
    const QString & savePath )
```

It is common that images are not directly saved to the destination path. For this reason, `save()` does not call `addAsReferredImage()`, and the stored save path may be wrong. Call this method after `save()` with the final destination path. This path will be stored in the image history as well.

### 6.391.3.32 isAnimatedImage()

```
bool Digikam::DImg::isAnimatedImage (
    const QString & filePath ) [static]
```

Return true if image file is an animation, as GIFa or NMG

### 6.391.3.33 isReadOnly()

```
bool Digikam::DImg::isReadOnly ( ) const
```

Return true if the original image file format cannot be saved. This is depending of `DImgLoader::save()` implementation. For example RAW file formats are supported by `DImg` using `dcrw` than cannot support writing operations.

### 6.391.3.34 lastSavedFilePath()

```
QString Digikam::DImg::lastSavedFilePath ( ) const
```

Returns the file path to which this `DImg` was saved. Returns the file path set with `imageSavedAs()`, if that was not called, `save()`, if that was not called, a null string.

### 6.391.3.35 loadItemInfo()

```
bool Digikam::DImg::loadItemInfo (
    const QString & filePath,
    bool loadMetadata = true,
    bool loadICCDData = true,
    bool loadUniqueHash = true,
    bool loadImageHistory = true )
```

Loads most parts of the meta information, but never the image data. If `loadMetadata` is true, the metadata will be available with `getComments`, `getExif`, `getIptc`, `getXmp` . If `loadICCDData` is true, the ICC profile will be available with `getICCProfile`.

### 6.391.3.36 operator=()

```
DImg & Digikam::DImg::operator= (
    const DImg & image )
```

Equivalent to the copy constructor

### 6.391.3.37 operator==( )

```
bool Digikam::DImg::operator==(
    const DImg & image ) const
```

Returns whether two images are equal. Two images are equal if and only if they refer to the same shared data. (Thus, `DImg() == DImg()` is not true, both instances refer to their own shared data. `image == DImg(image)` is true.) If two or more images refer to the same data, they have the same image data, `bits()` returns the same data, they have the same metadata, and a change to one image also affects the others. Call `detach()` to split one image from the group of equal images.

### 6.391.3.38 orientation( )

```
int Digikam::DImg::orientation ( ) const
```

Returns current `DMetadata::Orientation` from `DImg`

### 6.391.3.39 originalBitDepth( )

```
int Digikam::DImg::originalBitDepth ( ) const
```

Returns the bit depth (in bits per channel, e.g. 8 or 16) of the original file.

### 6.391.3.40 originalColorModel( )

```
DImg::COLORMODEL Digikam::DImg::originalColorModel ( ) const
```

Returns the color model in which the image was stored in the file. The color space of the loaded image data is always RGB.

### 6.391.3.41 originalFilePath( )

```
QString Digikam::DImg::originalFilePath ( ) const
```

Returns the file path from which this `DImg` was originally loaded. Returns a null string if the `DImg` was not loaded from a file.

### 6.391.3.42 originalRatioSize( )

```
QSize Digikam::DImg::originalRatioSize ( ) const
```

Returns the size of the original file in the same aspect ratio as `size()`.

### 6.391.3.43 originalSize( )

```
QSize Digikam::DImg::originalSize ( ) const
```

Returns the size of the original file.

**6.391.3.44 prepareMetadataToSave()**

```
void Digikam::DImg::prepareMetadataToSave (
    const QString & intendedDestPath,
    const QString & destMimeType,
    bool resetExifOrientationTag )
```

For convenience: Including all flags, except for ResetExifOrientationTag which can be selected. Uses [originalFilePath\(\)](#) to fill the original file name.

**6.391.3.45 pureColorMask()**

```
QImage Digikam::DImg::pureColorMask (
    ExposureSettingsContainer *const expoSettings ) const
```

Return a mask image where pure white and pure black pixels are over-colored. This way is used to identify over and under exposed pixels.

**6.391.3.46 putImageData() [1/2]**

```
void Digikam::DImg::putImageData (
    uchar *const data,
    bool copyData = true )
```

Overloaded function, provided for convenience, behaves essentially like the function above if data is not nullptr. Uses current width, height, sixteenBit, and alpha values. If data is nullptr, the current data are deleted and the image is set to null (But metadata are unchanged).

**6.391.3.47 putImageData() [2/2]**

```
void Digikam::DImg::putImageData (
    uint width,
    uint height,
    bool sixteenBit,
    bool alpha,
    uchar *const data,
    bool copyData = true )
```

Replaces image data of this object. Metadata are unchanged. Parameters like constructor above.

**6.391.3.48 rawDecodingSettings()**

```
DRawDecoding Digikam::DImg::rawDecodingSettings ( ) const
```

Returns the [DRawDecoding](#) options that this [DImg](#) was loaded with. If this is not a RAW image or no options were specified, returns [DRawDecoding\(\)](#).

### 6.391.3.49 removeAlphaChannel()

```
void Digikam::DImg::removeAlphaChannel (
    const DColor & destColor )
```

If the image has an alpha channel and transparent pixels, it will be blended on the specified color and the alpha channel will be removed. This is a no-op if [hasTransparentPixels\(\)](#) is false, but this method can be expensive, therefore it is *not* checked inside `removeAlphaChannel()`. (the trivial `hasAlpha()` is checked)

### 6.391.3.50 reset()

```
void Digikam::DImg::reset ( )
```

Reset metadata and image data to null image

### 6.391.3.51 resetMetaData()

```
void Digikam::DImg::resetMetaData ( )
```

Reset metadata, but do not change image data

### 6.391.3.52 resize()

```
void Digikam::DImg::resize (
    int w,
    int h )
```

Set width and height of this image, `smoothScale` it to the given size

### 6.391.3.53 reverseExifRotate()

```
bool Digikam::DImg::reverseExifRotate (
    const QString & filePath )
```

Reverses the previous function

### 6.391.3.54 reverseRotateAndFlip()

```
bool Digikam::DImg::reverseRotateAndFlip (
    int orientation )
```

Reverses the previous function.

### 6.391.3.55 rotateAndFlip()

```
bool Digikam::DImg::rotateAndFlip (
    int orientation )
```

Rotates and/or flip the [DImg](#) according to the given `DMetadata::Orientation`, so that the current state is orientation and the resulting step is normal orientation. Returns true if the image was actually rotated or flipped (e.g. if `ORIENTATION_NORMAL` is given, returns false, because no action is taken).

**6.391.3.56 savedFormat()**

```
QString Digikam::DImg::savedFormat ( ) const
```

Returns the format string of the format that this image was last saved to. An image can be loaded from a file - retrieve that format with [fileFormat\(\)](#) and [loadedFormat\(\)](#) - and can the multiple times be saved to different formats. Format strings used include JPG, PGF, PNG, TIFF and JP2K. If this file was not save, a null string is returned.

**6.391.3.57 setHistoryBranchAfter()**

```
void Digikam::DImg::setHistoryBranchAfter (
    const DImageHistory & historyBeforeBranch,
    bool isBranch = true )
```

Sets a step in the history to constitute the beginning of a branch. Use [setHistoryBranch\(\)](#) to take [getOriginalImageHistory\(\)](#) and set the first added step as a branch. Use [setHistoryBranchForLastSteps\(n\)](#) to start the branch before the last n steps in the history. (Assume the history had 3 steps and you added 2, call [setHistoryBranchForLastSteps\(2\)](#)) Use [setHistoryBranchAfter\(\)](#) if have a copy of the history before branching, the first added step on top of that history will be made a branch.

**6.391.3.58 smoothScale()**

```
DImg Digikam::DImg::smoothScale (
    int width,
    int height,
    Qt::AspectRatioMode aspectRatioMode = Qt::IgnoreAspectRatio ) const
```

Return a version of this image scaled to the specified size with the specified mode. See [QSize](#) documentation for information on available modes

**6.391.3.59 smoothScaleClipped()**

```
DImg Digikam::DImg::smoothScaleClipped (
    int width,
    int height,
    int clipx,
    int clipy,
    int clipwidth,
    int clipheight,
    bool smooth = true ) const
```

Executes the same scaling as [smoothScale\(width, height\)](#), but from the result of this call, returns only the section specified by *clipx*, *clipy*, *clipwidth*, *clipheight*. This is thus equivalent to calling `Dimg scaled = smoothScale(width, height); scaled.crop(clipx, clipy, clipwidth, clipheight);` but potentially much faster. In [smoothScaleSection](#), you specify the source region, here, the result region. It will often not be possible to find *integer* source coordinates for a result region!



### 6.391.3.60 smoothScaleSection()

```
DImg Digikam::DImg::smoothScaleSection (
    int sx,
    int sy,
    int sw,
    int sh,
    int dw,
    int dh ) const
```

Take the region specified by the rectangle *sx|sy*, width and height *sw \* sh*, and scale it to an image with size *dw \* dh*

### 6.391.3.61 stripImageData()

```
uchar * Digikam::DImg::stripImageData ( )
```

Returns the data of this image. Ownership of the buffer is passed to the caller, this image will be null afterwards.

### 6.391.3.62 transform()

```
bool Digikam::DImg::transform (
    int transformAction )
```

Rotates and/or flip the [DImg](#) according to the given transform action, which is a [MetaEngineRotation::TransformAction](#). Returns true if the image was actually rotated or flipped.

### 6.391.3.63 wasExifRotated()

```
bool Digikam::DImg::wasExifRotated ( )
```

Utility to make sure that an image is rotated according to Exif tag. Detects if an image has previously already been rotated: You can call this method more than one time on the same image. Returns true if the image has actually been rotated or flipped. Returns false if a rotation was not needed.

## 6.392 Digikam::DImgBuiltinFilter Class Reference

### Public Types

- enum [Type](#) {  
[NoOperation](#) , [Rotate90](#) , [Rotate180](#) , [Rotate270](#) ,  
[FlipHorizontally](#) , [FlipVertically](#) , [Crop](#) , [Resize](#) ,  
[ConvertTo8Bit](#) , [ConvertTo16Bit](#) }

## Public Member Functions

- void `apply` (`DImg` &image) const
- `DImgThreadedFilter` \* `createThreadedFilter` (`DImg` \*const orgImage, `QObject` \*const parent=nullptr) const
- `DImgThreadedFilter` \* `createThreadedFilter` (`QObject` \*const parent=nullptr) const
- `DImgBuiltinFilter` ()=default
- `DImgBuiltinFilter` (const `FilterAction` &action)
- `DImgBuiltinFilter` (`Type` type, const `QVariant` &arg=QVariant())
- `QString` `displayName` () const
- `FilterAction` `filterAction` () const
- `QString` `filterIcon` () const
- `QString` `i18nDisplayName` () const
- bool `isReversible` () const
- bool `isValid` () const
- `DImgBuiltinFilter` `reverseFilter` () const
- void `setAction` (const `FilterAction` &action)
- void `setAction` (`Type` type, const `QVariant` &arg=QVariant())

## Static Public Member Functions

- static `QString` `filterIcon` (const `QString` &filterIdentifier)
- static `QString` `i18nDisplayName` (const `QString` &filterIdentifier)
- static bool `isSupported` (const `QString` &filterIdentifier)
- static bool `isSupported` (const `QString` &filterIdentifier, int version)
- static `QStringList` `supportedFilters` ()
- static `QList`< int > `supportedVersions` (const `QString` &filterIdentifier)

## Protected Attributes

- `QVariant` `m_arg`
- `Type` `m_type` = NoOperation

## 6.392.1 Member Enumeration Documentation

### 6.392.1.1 Type

```
enum Digikam::DImgBuiltinFilter::Type
```

Enumerator

Crop	Argument: QRect.
Resize	Argument: QSize.

## 6.392.2 Constructor & Destructor Documentation

### 6.392.2.1 DImgBuiltinFilter() [1/3]

```
Digikam::DImgBuiltinFilter::DImgBuiltinFilter ( ) [default]
```

Create a filter performing no operation

### 6.392.2.2 DImgBuiltinFilter() [2/3]

```
Digikam::DImgBuiltinFilter::DImgBuiltinFilter (
    const FilterAction & action ) [explicit]
```

Create a filter for the given action. If the action is not supported, the filter will perform no operation.

### 6.392.2.3 DImgBuiltinFilter() [3/3]

```
Digikam::DImgBuiltinFilter::DImgBuiltinFilter (
    Type type,
    const QVariant & arg = QVariant\(\) ) [explicit]
```

Create a filter of the given type. See documentation of [Type](#) for required arguments.

## 6.392.3 Member Function Documentation

### 6.392.3.1 apply()

```
void Digikam::DImgBuiltinFilter::apply (
    DImg & image ) const
```

Apply the described change to the given image reference

### 6.392.3.2 createThreadedFilter()

```
DImgThreadedFilter * Digikam::DImgBuiltinFilter::createThreadedFilter (
    QObject *const parent = nullptr ) const
```

Returns a [DImgThreadedFilter](#) which executes this builtin action.

### 6.392.3.3 displayableName()

```
QString Digikam::DImgBuiltinFilter::displayableName ( ) const
```

Returns a displayableName for this filter

### 6.392.3.4 filterAction()

```
FilterAction Digikam::DImgBuiltinFilter::filterAction ( ) const
```

#### Note

The following methods are also accessed by the more general [DImgFilterManager](#) methods, so you usually do not need to call these directly. Returns the [FilterAction](#) describing this filter.

### 6.392.3.5 isSupported()

```
bool Digikam::DImgBuiltinFilter::isSupported (
    const QString & filterIdentifier ) [static]
```

Returns if the given filter and version are supported by [DImgBuiltinFilter](#)

### 6.392.3.6 isValid()

```
bool Digikam::DImgBuiltinFilter::isValid ( ) const
```

Checks that the action is supported and valid arguments are set

### 6.392.3.7 reverseFilter()

```
DImgBuiltinFilter Digikam::DImgBuiltinFilter::reverseFilter ( ) const
```

Returns the reverse action of this filter. If the current action is not revertible, returns an invalid filter.

### 6.392.3.8 setAction()

```
void Digikam::DImgBuiltinFilter::setAction (
    const FilterAction & action )
```

same as constructor

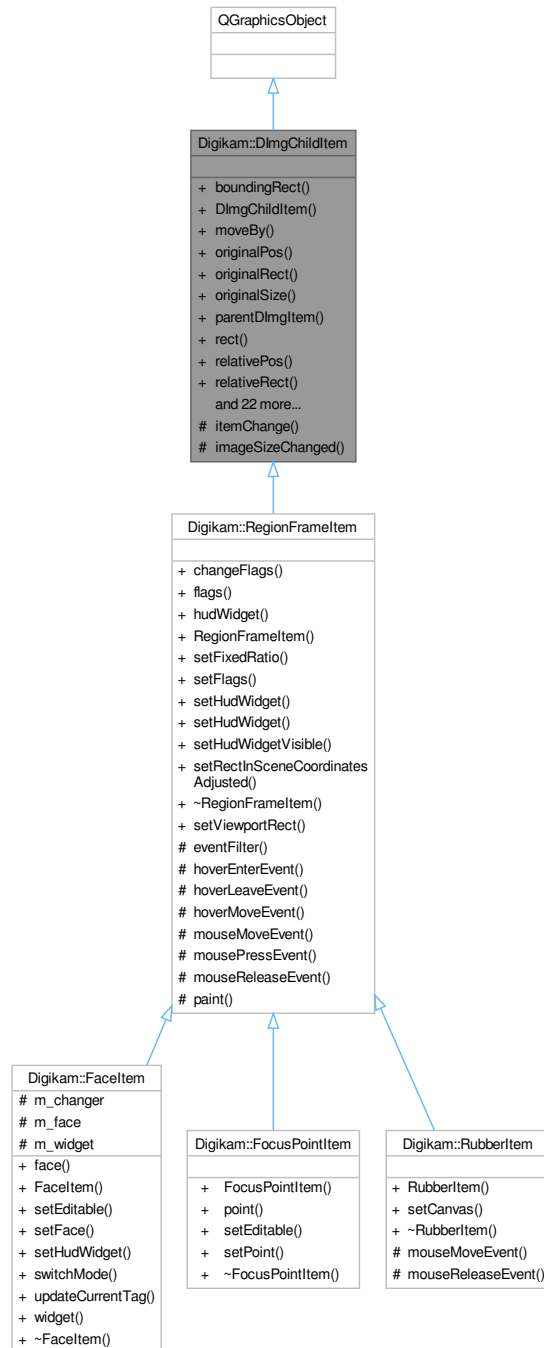
### 6.392.3.9 supportedVersions()

```
QList< int > Digikam::DImgBuiltinFilter::supportedVersions (
    const QString & filterIdentifier ) [static]
```

Returns a list of supported versions of the given filter

## 6.393 Digikam::DImgChildItem Class Reference

Inheritance diagram for Digikam::DImgChildItem:



### Signals

- void **geometryChanged** ()
- void **geometryOnImageChanged** ()

- void [positionChanged](#) ()
- void [positionOnImageChanged](#) ()
- void **sizeChanged** ()
- void **sizeOnImageChanged** ()

### Public Member Functions

- QRectF [boundingRect](#) () const override
- [DImgChildItem](#) (QGraphicsItem \*const parent=nullptr)
- void **moveBy** (qreal dx, qreal dy)
- QPoint **originalPos** () const
- QRect [originalRect](#) () const
- QSize **originalSize** () const
- [GraphicsDImgItem](#) \* [parentDImgItem](#) () const
- QRectF [rect](#) () const
- QPointF **relativePos** () const
- QRectF [relativeRect](#) () const
- QSizeF **relativeSize** () const
- void [setOriginalPos](#) (const QPointF &posInOriginal)
- void **setOriginalPos** (qreal x, qreal y)
- void **setOriginalRect** (const QRectF &[rect](#))
- void **setOriginalRect** (qreal x, qreal y, qreal width, qreal height)
- void **setOriginalSize** (const QSizeF &sizeInOriginal)
- void **setOriginalSize** (qreal width, qreal height)
- void [setPos](#) (const QPointF &zoomedPos)
- void **setPos** (qreal x, qreal y)
- void **setRect** (const QRectF &[rect](#))
- void **setRect** (qreal x, qreal y, qreal width, qreal height)
- void [setRectInSceneCoordinates](#) (const QRectF &[rect](#))
- void [setRelativePos](#) (const QPointF &relativePosition)
- void **setRelativePos** (qreal x, qreal y)
- void **setRelativeRect** (const QRectF &[rect](#))
- void **setRelativeRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRelativeSize** (const QSizeF &relativeSize)
- void **setRelativeSize** (qreal width, qreal height)
- void **setSize** (const QSizeF &zoomedSize)
- void **setSize** (qreal width, qreal height)
- QSizeF **size** () const

### Protected Slots

- void **imageSizeChanged** (const QSizeF &)

### Protected Member Functions

- QVariant **itemChange** (GraphicsItemChange change, const QVariant &value) override

## 6.393.1 Constructor & Destructor Documentation

### 6.393.1.1 DImgChildItem()

```
Digikam::DImgChildItem::DImgChildItem (
    QGraphicsItem *const parent = nullptr ) [explicit]
```

This is a base class for items that are positioned on top of a [GraphicsDImgItem](#), positioned in relative coordinates, i.e. [0;1], on the image. From the set relative size, the [boundingRect\(\)](#) is calculated.

## 6.393.2 Member Function Documentation

### 6.393.2.1 boundingRect()

```
QRectF Digikam::DImgChildItem::boundingRect ( ) const [override]
```

Reimplemented. Returns a rectangle starting at (0,0) (pos() in parent coordinates) and has a size determined by the relative size.

### 6.393.2.2 originalRect()

```
QRect Digikam::DImgChildItem::originalRect ( ) const
```

Returns the position and size in coordinates of the original image. Note that the return value is integer based. At high zoom rates, different values of [relativeRect\(\)](#) or [zoomedRect\(\)](#) may result in the same [originalRect\(\)](#), when one pixel in the original is represented by more than one pixel on screen.

### 6.393.2.3 parentDImgItem()

```
GraphicsDImgItem * Digikam::DImgChildItem::parentDImgItem ( ) const
```

If the parent item is a [GraphicsDImgItem](#), return it, if the parent item is null or of a different class, returns 0.

### 6.393.2.4 positionChanged

```
void Digikam::DImgChildItem::positionChanged ( ) [signal]
```

These signals are emitted in any case when the geometry changed: Either after changing the geometry relative to the original image, or when the size of the parent [GraphicsDImgItem](#) changed (zooming). [positionChanged\(\)](#) is equivalent to listening to [xChanged\(\)](#) and [yChanged\(\)](#).

### 6.393.2.5 positionOnImageChanged

```
void Digikam::DImgChildItem::positionOnImageChanged ( ) [signal]
```

These signals are emitted when the geometry, relative to the original image, of this item has changed. This happens by calling any of the methods above.

### 6.393.2.6 rect()

```
QRectF Digikam::DImgChildItem::rect ( ) const
```

Returns position and size of this item, in coordinates of the parent [DImg](#) with the current zoom. This is the same result as `QRectF(pos(), boundingRect())`, `boundingRect` is virtual and may be overridden by base classes.

### 6.393.2.7 relativeRect()

```
QRectF Digikam::DImgChildItem::relativeRect ( ) const
```

Returns the position and size relative to the [DImg](#) displayed in the parent item. All four values are in the interval [0;1].

### 6.393.2.8 setOriginalPos()

```
void Digikam::DImgChildItem::setOriginalPos (
    const QPointF & posInOriginal )
```

Sets the position and size of this item, in coordinates of the original image. Requires a valid parent item.

### 6.393.2.9 setPos()

```
void Digikam::DImgChildItem::setPos (
    const QPointF & zoomedPos )
```

Sets the position and size of this item, in coordinates of the parent [DImg](#) item. This is accepting unscaled parent coordinates, just like the "normal" [setPos\(\)](#) does. Requires a valid parent item.

### 6.393.2.10 setRectInSceneCoordinates()

```
void Digikam::DImgChildItem::setRectInSceneCoordinates (
    const QRectF & rect )
```

Equivalent to mapping the scene coordinates to the parent item, and calling `setRect()`.

### 6.393.2.11 setRelativePos()

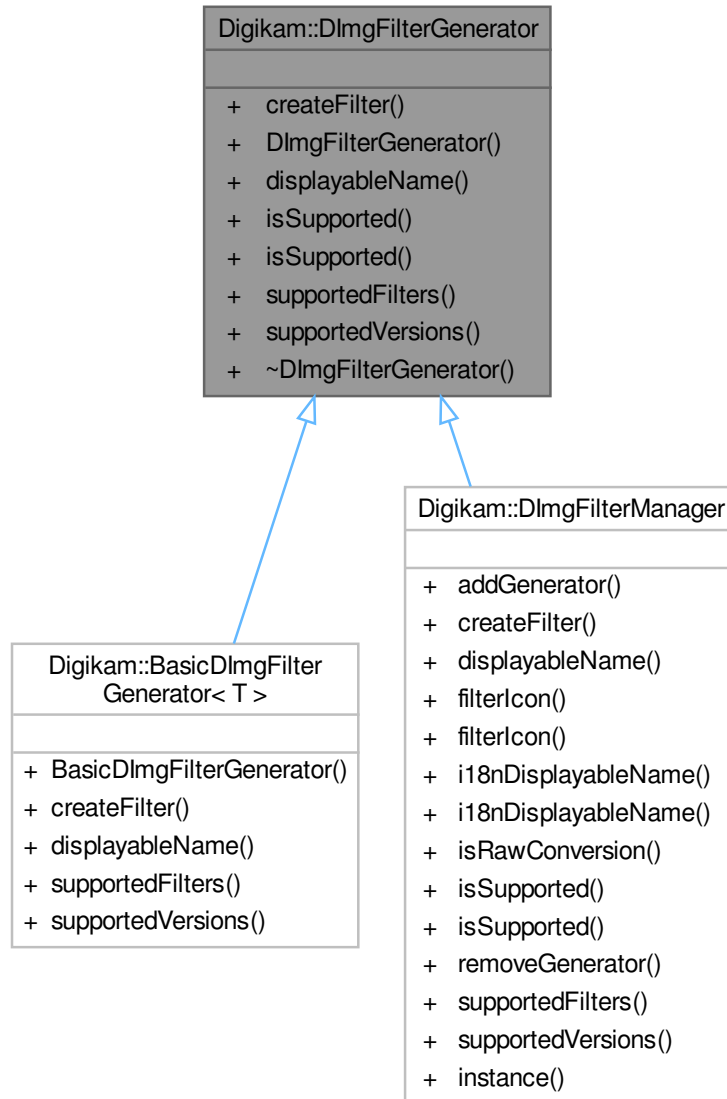
```
void Digikam::DImgChildItem::setRelativePos (
    const QPointF & relativePosition )
```

Sets the position and size of this item, relative to the [DImg](#) displayed in the parent item. The values of `relativePosition` must be in the interval [0;1].



## 6.394 Digikam::DImgFilterGenerator Class Reference

Inheritance diagram for Digikam::DImgFilterGenerator:



### Public Member Functions

- virtual `DImgThreadedFilter * createFilter` (const QString &filterIdentifier, int version)=0  
*Create the filter for the given combination of identifier and version.*
- virtual QString `displayName` (const QString &filterIdentifier)=0  
*Returns a QString with filter name for displaying in views.*
- virtual bool `isSupported` (const QString &filterIdentifier)  
*Convenience methods.*
- virtual bool `isSupported` (const QString &filterIdentifier, int version)

- virtual QStringList [supportedFilters](#) ()=0  
*Returns a list with identifiers of supported filters.*
- virtual QList< int > [supportedVersions](#) (const QString &filterIdentifier)=0  
*Returns a list with the supported versions for the given identifier.*

## 6.394.1 Member Function Documentation

### 6.394.1.1 createFilter()

```
virtual DImgThreadedFilter * Digikam::DImgFilterGenerator::createFilter (
    const QString & filterIdentifier,
    int version ) [pure virtual]
```

Implemented in [Digikam::BasicDImgFilterGenerator< T >](#), and [Digikam::DImgFilterManager](#).

### 6.394.1.2 displayableName()

```
virtual QString Digikam::DImgFilterGenerator::displayableName (
    const QString & filterIdentifier ) [pure virtual]
```

Implemented in [Digikam::BasicDImgFilterGenerator< T >](#), and [Digikam::DImgFilterManager](#).

### 6.394.1.3 isSupported()

```
bool Digikam::DImgFilterGenerator::isSupported (
    const QString & filterIdentifier ) [virtual]
```

Reimplemented in [Digikam::DImgFilterManager](#).

### 6.394.1.4 supportedFilters()

```
virtual QStringList Digikam::DImgFilterGenerator::supportedFilters ( ) [pure virtual]
```

Implemented in [Digikam::BasicDImgFilterGenerator< T >](#), and [Digikam::DImgFilterManager](#).

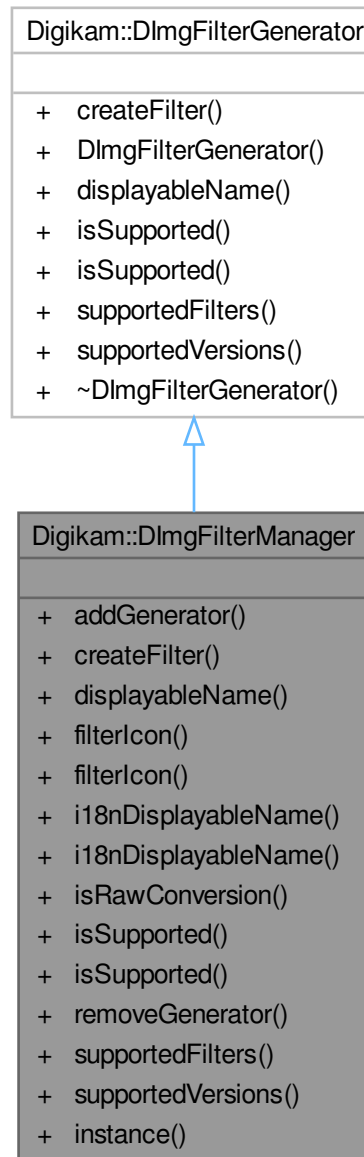
### 6.394.1.5 supportedVersions()

```
virtual QList< int > Digikam::DImgFilterGenerator::supportedVersions (
    const QString & filterIdentifier ) [pure virtual]
```

Implemented in [Digikam::BasicDImgFilterGenerator< T >](#), and [Digikam::DImgFilterManager](#).

## 6.395 Digikam::DImgFilterManager Class Reference

Inheritance diagram for Digikam::DImgFilterManager:



### Public Member Functions

- void `addGenerator` (`DImgFilterGenerator` \*const generator)
- `DImgThreadedFilter` \* `createFilter` (const QString &filterIdentifier, int version) override
- QString `displayName` (const QString &filterIdentifier) override
- QString `filterIcon` (const `FilterAction` &action)
- QString `filterIcon` (const QString &filterIdentifier)

- QString **i18nDisplayName** (const [FilterAction](#) &action)
- QString **i18nDisplayName** (const QString &filterIdentifier)
- bool **isRawConversion** (const QString &filterIdentifier)
- bool **isSupported** (const QString &filterIdentifier) override
- bool **isSupported** (const QString &filterIdentifier, int version) override
- void **removeGenerator** ([DImgFilterGenerator](#) \*const generator)
- QStringList **supportedFilters** () override
- QList< int > **supportedVersions** (const QString &filterIdentifier) override

### Static Public Member Functions

- static [DImgFilterManager](#) \* **instance** ()

### Friends

- class [DImgFilterManagerCreator](#)

## 6.395.1 Member Function Documentation

### 6.395.1.1 addGenerator()

```
void Digikam::DImgFilterManager::addGenerator (
    DImgFilterGenerator *const generator )
```

Registers all filter provided by this generator.

### 6.395.1.2 createFilter()

```
DImgThreadedFilter * Digikam::DImgFilterManager::createFilter (
    const QString & filterIdentifier,
    int version ) [override], [virtual]
```

Create a filter from an installed manager. Returns 0 if no filter could be created. This is true if identifier/version is not supported, or the filter is builtin. Note: You probably want to use [FilterActionFilter](#).

Implements [Digikam::DImgFilterGenerator](#).

### 6.395.1.3 displayName()

```
QString Digikam::DImgFilterManager::displayName (
    const QString & filterIdentifier ) [override], [virtual]
```

Returns the (untranslated) displayable name for the given identifier. This is only possible for supported filters. If you have a [FilterAction](#), it may already contain a displayable name.

Implements [Digikam::DImgFilterGenerator](#).

#### 6.395.1.4 filterIcon()

```
QString Digikam::DImgFilterManager::filterIcon (
    const QString & filterIdentifier )
```

Returns an icon for the given filter. If no icon is known, returns a null string.

#### 6.395.1.5 i18nDisplayableName()

```
QString Digikam::DImgFilterManager::i18nDisplayableName (
    const QString & filterIdentifier )
```

Returns the translated displayable name

#### 6.395.1.6 isRawConversion()

```
bool Digikam::DImgFilterManager::isRawConversion (
    const QString & filterIdentifier )
```

Returns true if the given filter is to be considered as a step converting a RAW image to a normal image.

#### 6.395.1.7 isSupported() [1/2]

```
bool Digikam::DImgFilterManager::isSupported (
    const QString & filterIdentifier ) [override], [virtual]
```

Returns true if the given filter, or, more specifically, the given filter in the given version is supported.

Reimplemented from [Digikam::DImgFilterGenerator](#).

#### 6.395.1.8 isSupported() [2/2]

```
bool Digikam::DImgFilterManager::isSupported (
    const QString & filterIdentifier,
    int version ) [override], [virtual]
```

Reimplemented from [Digikam::DImgFilterGenerator](#).

#### 6.395.1.9 supportedFilters()

```
QStringList Digikam::DImgFilterManager::supportedFilters ( ) [override], [virtual]
```

Returns a list of the supported filter identifiers

Implements [Digikam::DImgFilterGenerator](#).

### 6.395.1.10 supportedVersions()

```
QList< int > Digikam::DImgFilterManager::supportedVersions (
    const QString & filterIdentifier ) [override], [virtual]
```

Returns a list of supported versions of the given filter

Implements [Digikam::DImgFilterGenerator](#).

## 6.396 Digikam::DImgLoader Class Reference

### Public Types

- enum [LoadFlag](#) {
 [LoadItemInfo](#) = 1 , [LoadMetadata](#) = 2 , [LoadICCDData](#) = 4 , [LoadImageData](#) = 8 ,
 [LoadUniqueHash](#) = 16 , [LoadImageHistory](#) = 32 , [LoadPreview](#) = 64 , [LoadAll](#) = LoadItemInfo | LoadMetadata
 | LoadICCDData | LoadImageData | LoadUniqueHash | LoadImageHistory }

### Public Member Functions

- virtual bool **hasAlpha** () const =0
- virtual bool **hasLoadedData** () const
- virtual bool **isReadOnly** () const =0
- virtual bool **load** (const QString &filePath, [DImgLoaderObserver](#) \*const observer)=0
- template<typename Type >
 Q\_INLINE\_TEMPLATE Type \* [new\\_failureTolerant](#) (qint64 w, quint64 h, uint typesPerPixel)
- template<typename Type >
 Q\_INLINE\_TEMPLATE Type \* [new\\_failureTolerant](#) (size\_t size)
- virtual bool **save** (const QString &filePath, [DImgLoaderObserver](#) \*const observer)=0
- void **setLoadFlags** (LoadFlags flags)
- virtual bool **sixteenBit** () const =0

### Static Public Member Functions

- static qint64 [checkAllocation](#) (qint64 fullSize)
- static int [convertCompressionForLibJpeg](#) (int value)
- static int [convertCompressionForLibPng](#) (int value)
- static unsigned char \* [new\\_failureTolerant](#) (qint64 w, quint64 h, uint typesPerPixel)
- template<typename Type >
 static Type \* [new\\_failureTolerant](#) (qint64 w, quint64 h, uint typesPerPixel)
- static unsigned char \* [new\\_failureTolerant](#) (size\_t unsecureSize)
- template<typename Type >
 static Type \* [new\\_failureTolerant](#) (size\_t unsecureSize)
- static unsigned short \* [new\\_short\\_failureTolerant](#) (qint64 w, quint64 h, uint typesPerPixel)
- static unsigned short \* [new\\_short\\_failureTolerant](#) (size\_t unsecureSize)

## Protected Member Functions

- bool **checkExifWorkingColorSpace** () const
- **DImgLoader** ([DImg](#) \*const image)
- virtual int **granularity** ([DImgLoaderObserver](#) \*const observer, int total, float progressSlice=1.0F)
- int **imageBitsDepth** () const
- int **imageBytesDepth** () const
- unsigned char \*& **imageData** ()
- QMap< QString, QString > & **imageEmbeddedText** () const
- QVariant **imageGetAttribute** (const QString &key) const
- QString **imageGetEmbeddedText** (const QString &key) const
- bool **imageHasAlpha** () const
- unsigned int & **imageHeight** ()
- quint64 **imageNumBytes** () const
- void **imageSetAttribute** (const QString &key, const QVariant &value)
- void **imageSetEmbeddedText** (const QString &key, const QString &text)
- void **imageSetIccProfile** (const [IccProfile](#) &profile)
- bool **imageSixteenBit** () const
- unsigned int & **imageWidth** ()
- void **loadingFailed** ()
- void **purgeExifWorkingColorSpace** ()
- virtual bool **readMetadata** (const QString &filePath)
- virtual bool **saveMetadata** (const QString &filePath)
- void **storeColorProfileInMetadata** ()

## Protected Attributes

- [DImg](#) \* **m\_image** = nullptr
- LoadFlags **m\_loadFlags** = [LoadAll](#)

## 6.396.1 Member Enumeration Documentation

### 6.396.1.1 LoadFlag

enum [Digikam::DImgLoader::LoadFlag](#)

This is the list of loading modes usable by [DImg](#) image plugins

#### Enumerator

LoadItemInfo	Load image information without image data. Image info as width and height
LoadMetadata	Image metadata.
LoadICCDData	Image color profile.
LoadImageData	Full image data.
LoadUniqueHash	Image unique hash.
LoadImageHistory	Image version history.
LoadPreview	Special mode to load reduced image data. Load embedded preview image instead full size image
LoadAll	Helper to load all information, metadata and full image.

## 6.396.2 Member Function Documentation

### 6.396.2.1 checkAllocation()

```
qint64 Digikam::DImgLoader::checkAllocation (
    qint64 fullSize ) [static]
```

Value returned : -1 : unsupported platform 0 : parse failure from supported platform 1 : parse done with success from supported platform

### 6.396.2.2 new\_failureTolerant()

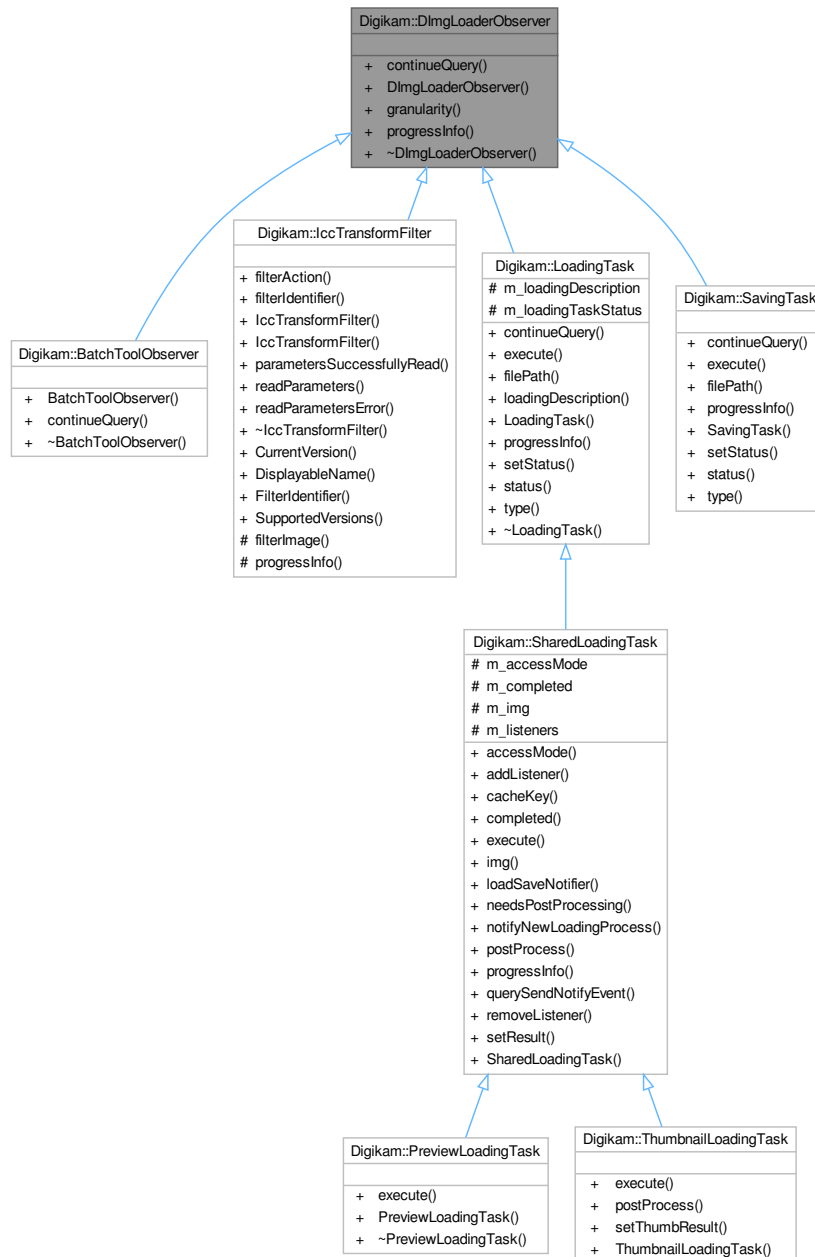
```
template<typename Type >
Q_INLINE_TEMPLATE Type * Digikam::DImgLoader::new_failureTolerant (
    quint64 w,
    quint64 h,
    uint typesPerPixel )
```

Allows safe multiplication of requested pixel number and bytes per pixel, avoiding particularly 32 bits overflow and exceeding the `size_t` type



## 6.397 Digikam::DImgLoaderObserver Class Reference

Inheritance diagram for Digikam::DImgLoaderObserver:



### Public Member Functions

- virtual bool `continueQuery` ()
- virtual float `granularity` ()
- virtual void `progressInfo` (float progress)

## 6.397.1 Member Function Documentation

### 6.397.1.1 `continueQuery()`

```
virtual bool Digikam::DImgLoaderObserver::continueQuery ( ) [inline], [virtual]
```

Queries whether the image IO operation shall be continued

### 6.397.1.2 `granularity()`

```
virtual float Digikam::DImgLoaderObserver::granularity ( ) [inline], [virtual]
```

Return a relative value which determines the granularity, the frequency with which the `DImgLoaderObserver` is checked and progress is posted. Standard is 1.0. Values < 1 mean less granularity (fewer checks), values > 1 mean higher granularity (more checks).

### 6.397.1.3 `progressInfo()`

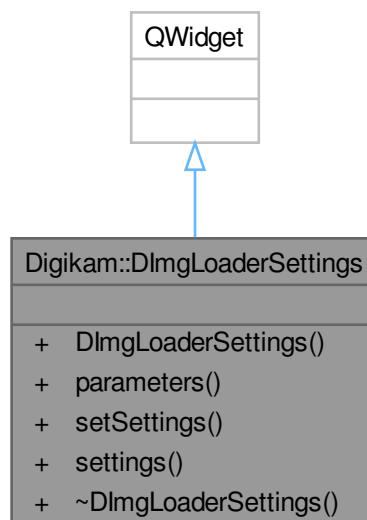
```
virtual void Digikam::DImgLoaderObserver::progressInfo (
    float progress ) [inline], [virtual]
```

Posts progress information about image IO

Reimplemented in [Digikam::lccTransformFilter](#).

## 6.398 Digikam::DImgLoaderSettings Class Reference

Inheritance diagram for Digikam::DImgLoaderSettings:



## Signals

- void [signalSettingsChanged](#) ()

## Public Member Functions

- **DImgLoaderSettings** (QWidget \*const parent=nullptr)
- QStringList [parameters](#) () const
- virtual void [setSettings](#) (const [DImgLoaderPrms](#) &set)=0
- virtual [DImgLoaderPrms](#) [settings](#) () const =0

## 6.398.1 Member Function Documentation

### 6.398.1.1 parameters()

```
QStringList Digikam::DImgLoaderSettings::parameters ( ) const
```

Return the list of supported parameter names.

### 6.398.1.2 setSettings()

```
virtual void Digikam::DImgLoaderSettings::setSettings (
    const DImgLoaderPrms & set ) [pure virtual]
```

Set the parameters values in the widget from DImgLoaderPrms map container.

### 6.398.1.3 settings()

```
virtual DImgLoaderPrms Digikam::DImgLoaderSettings::settings ( ) const [pure virtual]
```

Return the DImgLoaderPrms map container of parameters/values from the Widget.

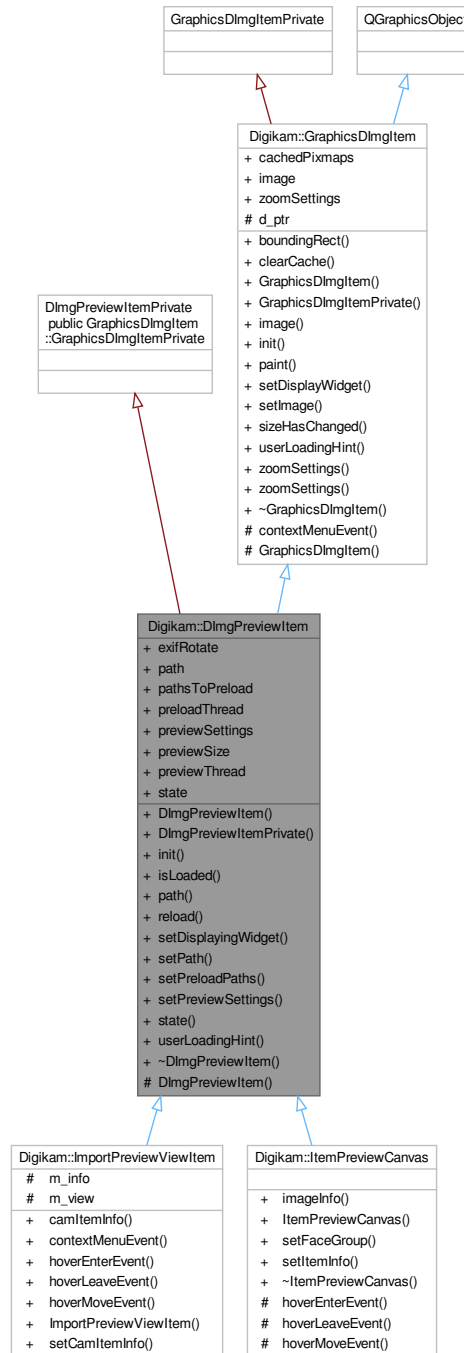
### 6.398.1.4 signalSettingsChanged

```
void Digikam::DImgLoaderSettings::signalSettingsChanged ( ) [signal]
```

Signal to emit when a settings is changed from the widget.

## 6.399 Digikam::DImgPreviewItem Class Reference

Inheritance diagram for Digikam::DImgPreviewItem:



### Public Types

- enum **State** { **NoImage** , **Loading** , **ImageLoaded** , **ImageLoadingFailed** }

## Signals

- void **loaded** ()
- void **loadingFailed** ()
- void **stateChanged** (int state)

## Signals inherited from [Digikam::GraphicsDImgItem](#)

- void **imageChanged** ()
- void **imageSizeChanged** (const QSizeF &size)
- void **showContextMenu** (QGraphicsSceneContextMenuEvent \*e)

## Public Member Functions

- **DImgPreviewItem** (QGraphicsItem \*const parent=nullptr)
- **DImgPreviewItemPrivate** ()=default
- void **init** ([DImgPreviewItem](#) \*const q)
- bool **isLoading** () const
- QString **path** () const
- void **reload** ()
- void **setDisplayingWidget** (QWidget \*const widget)
- void **setPath** (const QString &path, bool rePreview=false)
- void **setPreloadPaths** (const QStringList &pathsToPreload)
- void **setPreviewSettings** (const [PreviewSettings](#) &settings)
- State **state** () const
- QString **userLoadingHint** () const override

## Public Member Functions inherited from [Digikam::GraphicsDImgItem](#)

- QRectF **boundingRect** () const override
- void **clearCache** ()
- **GraphicsDImgItem** (QGraphicsItem \*const parent=nullptr)
- **GraphicsDImgItemPrivate** ()=default
- [DImg](#) **image** () const
- void **init** ([GraphicsDImgItem](#) \*const q)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override
- void **setDisplayWidget** (QWidget \*const widget)
- void **setImage** (const [DImg](#) &img)
- void **sizeHasChanged** ()
- [ImageZoomSettings](#) \* **zoomSettings** ()
- const [ImageZoomSettings](#) \* **zoomSettings** () const

## Public Attributes

- bool **exifRotate** = false
- QString **path**
- QStringList **pathsToPreload**
- [PreviewLoadThread](#) \* **preloadThread** = nullptr
- [PreviewSettings](#) **previewSettings**
- int **previewSize** = 1024
- [PreviewLoadThread](#) \* **previewThread** = nullptr
- [DImgPreviewItem::State](#) **state** = [DImgPreviewItem::NoImage](#)

## Public Attributes inherited from [Digikam::GraphicsDImgItem](#)

- [CachedPixmaps](#) `cachedPixmaps`
- [DImg](#) `image`
- [ImageZoomSettings](#) `zoomSettings`

## Protected Member Functions

- [DImgPreviewItem](#) (`DImgPreviewItemPrivate &dd, QGraphicsItem *const parent=nullptr`)

## Protected Member Functions inherited from [Digikam::GraphicsDImgItem](#)

- void `contextMenuEvent` (`QGraphicsSceneContextMenuEvent *e`) override
- [GraphicsDImgItem](#) (`GraphicsDImgItemPrivate &dd, QGraphicsItem *const parent`)

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::GraphicsDImgItem](#)

- `GraphicsDImgItemPrivate *const d_ptr`

## 6.399.1 Member Function Documentation

### 6.399.1.1 `userLoadingHint()`

```
QString Digikam::DImgPreviewItem::userLoadingHint ( ) const [override], [virtual]
```

Reimplemented from [Digikam::GraphicsDImgItem](#).

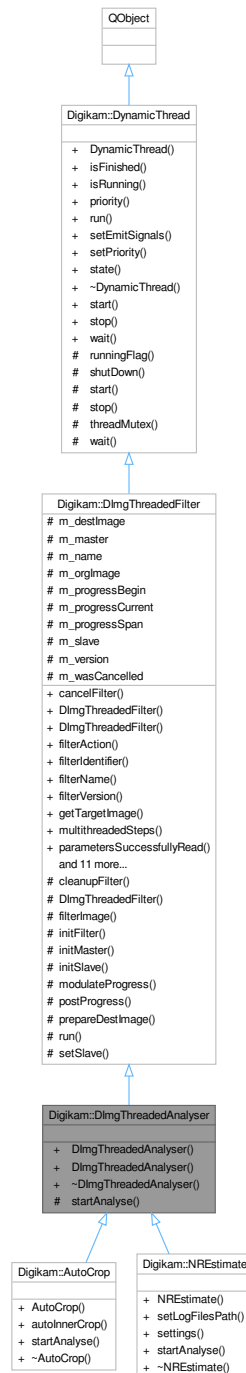
## 6.400 Digikam::DImgStaticPriv Class Reference

### Static Public Member Functions

- static `QStringList fileOriginAttributes` ()
- static [DImg::FORMAT](#) `loaderNameToFormat` (const `QString &name`)
- static [DPluginDImg](#) \* `pluginForFile` (const `QFileInfo &fileInfo, bool magic`)
- static [DPluginDImg](#) \* `pluginForFormat` (const `QString &format`)

## 6.401 Digikam::DImgThreadedAnalyser Class Reference

Inheritance diagram for Digikam::DImgThreadedAnalyser:



### Public Member Functions

- **DImgThreadedAnalyser** (`DImg *const orgImage`, `QObject *const parent=nullptr`, `const QString &name=QString()`)

Constructs an image analyser with all arguments (ready to use). The given original image will be copied. You need to call [startFilter\(\)](#) to start the threaded computation. To run analyser without to use multithreading, call [startFilterDirectly\(\)](#).

- [DImgThreadedAnalyser](#) (QObject \*const parent=nullptr, const QString &name=QString())

Constructs a filter without argument. You need to call [setupFilter\(\)](#) and [startFilter\(\)](#) to start the threaded computation. To run filter without to use multithreading, call [startFilterDirectly\(\)](#).

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) (DImg \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

## Protected Member Functions

- virtual void [startAnalyse](#) ()=0

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cleanupFilter](#) ()
- [DImgThreadedFilter](#) ([DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void [initFilter](#) ()
- void [initMaster](#) ()
- void [initSlave](#) ([DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- virtual int [modulateProgress](#) (int progress)
- void [postProgress](#) (int progress)
- void [run](#) () override
- void [setSlave](#) ([DImgThreadedFilter](#) \*const slave)



## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void [start](#) ()
- void [stop](#) ()
- void [wait](#) ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void [finished](#) (bool success)
- void [progress](#) (int progress)
- void [started](#) ()

## Signals inherited from [Digikam::DynamicThread](#)

- void [finished](#) ()
- void [starting](#) ()

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- [DImg](#) [m\\_destImage](#)
- [DImgThreadedFilter](#) \* [m\\_master](#) = nullptr
- [QString](#) [m\\_name](#)
- [DImg](#) [m\\_orgImage](#)
- int [m\\_progressBegin](#) = 0
- int [m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int [m\\_progressSpan](#) = 0
- [DImgThreadedFilter](#) \* [m\\_slave](#) = nullptr
- int [m\\_version](#) = 1
- bool [m\\_wasCancelled](#) = false

## 6.401.1 Constructor & Destructor Documentation

### 6.401.1.1 DImgThreadedAnalyser()

```
Digikam::DImgThreadedAnalyser::DImgThreadedAnalyser (
    QObject *const parent = nullptr,
    const QString & name = QString() ) [explicit]
```

#### Warning

Versioning is not supported in this class

## 6.401.2 Member Function Documentation

### 6.401.2.1 startAnalyse()

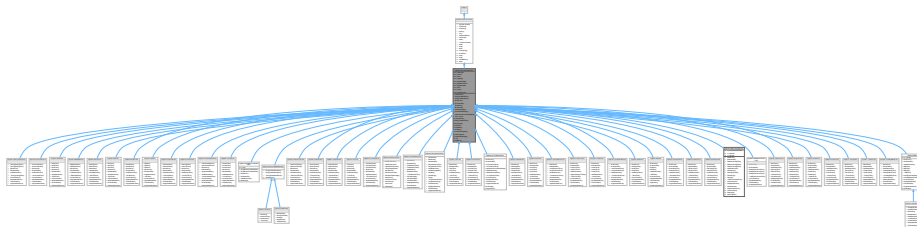
```
virtual void Digikam::DImgThreadedAnalyser::startAnalyse ( ) [protected], [pure virtual]
```

Main image analys method. Override in subclass.

Implemented in [Digikam::NREstimate](#), and [Digikam::AutoCrop](#).

## 6.402 Digikam::DImgThreadedFilter Class Reference

Inheritance diagram for Digikam::DImgThreadedFilter:



#### Classes

- class [DefaultFilterAction](#)

#### Signals

- void [finished](#) (bool success)
- void [progress](#) (int progress)
- void [started](#) ()

#### Signals inherited from [Digikam::DynamicThread](#)

- void [finished](#) ()
- void [starting](#) ()

**Public Member Functions**

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- virtual [FilterAction](#) [filterAction](#) ()=0
- virtual QString [filterIdentifier](#) () const =0
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int [stop](#), int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual void [readParameters](#) (const [FilterAction](#) &)=0
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

**Public Member Functions inherited from [Digikam::DynamicThread](#)**

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

**Protected Member Functions**

- virtual void [cleanupFilter](#) ()
- [DImgThreadedFilter](#) ([DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void [filterImage](#) ()=0
- virtual void [initFilter](#) ()
- void [initMaster](#) ()
- void [initSlave](#) ([DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- virtual int [modulateProgress](#) (int [progress](#))
- void [postProgress](#) (int [progress](#))
- virtual void [prepareDestImage](#) ()
- void [run](#) () override
- void [setSlave](#) ([DImgThreadedFilter](#) \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

## Protected Attributes

- [DImg m\\_destImage](#)
- [DImgThreadedFilter \\* m\\_master](#) = nullptr
- [QString m\\_name](#)
- [DImg m\\_orgImage](#)
- int [m\\_progressBegin](#) = 0
- int [m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int [m\\_progressSpan](#) = 0
- [DImgThreadedFilter \\* m\\_slave](#) = nullptr
- int [m\\_version](#) = 1
- bool [m\\_wasCancelled](#) = false

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void [start](#) ()
- void [stop](#) ()
- void [wait](#) ()

## 6.402.1 Constructor & Destructor Documentation

### 6.402.1.1 [DImgThreadedFilter\(\)](#) [1/3]

```
Digikam::DImgThreadedFilter::DImgThreadedFilter (
    QObject *const parent = nullptr,
    const QString & name = QString() ) [explicit]
```

Constructs a filter without argument. You need to call [setupFilter\(\)](#) and [startFilter\(\)](#) to start the threaded computation. To run filter without to use multithreading, call [startFilterDirectly\(\)](#).

### 6.402.1.2 DImgThreadedFilter() [2/3]

```
Digikam::DImgThreadedFilter::DImgThreadedFilter (
    DImg *const orgImage,
    QObject *const parent,
    const QString & name = QString() )
```

Constructs a filter with all arguments (ready to use). The given original image will be copied. You need to call [startFilter\(\)](#) to start the threaded computation. To run filter without to use multithreading, call [startFilterDirectly\(\)](#).

### 6.402.1.3 DImgThreadedFilter() [3/3]

```
Digikam::DImgThreadedFilter::DImgThreadedFilter (
    DImgThreadedFilter *const master,
    const DImg & orgImage,
    const DImg & destImage,
    int progressBegin = 0,
    int progressEnd = 100,
    const QString & name = QString() ) [protected]
```

Support for chaining two filters as master and thread.

Do not call [startFilter\(\)](#) or [startFilterDirectly\(\)](#) on this. The computation will be started from [initFilter\(\)](#) which you must call from the derived class constructor.

Constructor for slave mode: Constructs a new slave filter with the specified master. The filter will be executed in the current thread. orgImage and destImage will not be copied. Note that the slave is still free to reallocate his destImage. progressBegin and progressEnd can indicate the progress span that the slave filter uses in the parent filter's progress. Any derived filter class that is publicly available to other filters should implement an additional constructor using this constructor.

## 6.402.2 Member Function Documentation

### 6.402.2.1 cancelFilter()

```
void Digikam::DImgThreadedFilter::cancelFilter ( ) [virtual]
```

Cancel the threaded computation.

Reimplemented in [Digikam::GreycstorationFilter](#).

### 6.402.2.2 cleanupFilter()

```
virtual void Digikam::DImgThreadedFilter::cleanupFilter ( ) [inline], [protected], [virtual]
```

Clean up filter data if necessary, called by stopComputation() method. Override in subclass.

### 6.402.2.3 filterAction()

```
virtual FilterAction Digikam::DImgThreadedFilter::filterAction ( ) [pure virtual]
```

Returns the action description corresponding to currently set options.

Implemented in [Digikam::AutoExpoFilter](#), [Digikam::AutoLevelsFilter](#), [Digikam::EqualizeFilter](#), [Digikam::NormalizeFilter](#), [Digikam::StretchFilter](#), [Digikam::BCGFilter](#), [Digikam::BWSepiaFilter](#), [Digikam::InfraredFilter](#), [Digikam::MixerFilter](#), [Digikam::TonalityFilter](#), [Digikam::CBFilter](#), [Digikam::CurvesFilter](#), [Digikam::BorderFilter](#), [Digikam::TextureFilter](#), [Digikam::FilmFilter](#), [Digikam::FilterActionFilter](#), [Digikam::BlurFilter](#), [Digikam::BlurFXFilter](#), [Digikam::CharcoalFilter](#), [Digikam::ColorFXFilter](#), [Digikam::DistortionFXFilter](#), [Digikam::EmbossFilter](#), [Digikam::FilmGrainFilter](#), [Digikam::InvertFilter](#), [Digikam::OilPaintFilter](#), [Digikam::RainDropFilter](#), [Digikam::GreycstorationFilter](#), [Digikam::HotPixelFixer](#), [Digikam::HSLFilter](#), [Digikam::lccTransformFilter](#), [Digikam::LocalContrastFilter](#), [Digikam::AntiVignettingFilter](#), [Digikam::LensDistortionFilter](#), [Digikam::LensFunFilter](#), [Digikam::LevelsFilter](#), [Digikam::NRFilter](#), [Digikam::RawProcessingFilter](#), [Digikam::RedEyeCorrectionFilter](#), [Digikam::RefocusFilter](#), [Digikam::SharpenFilter](#), [Digikam::UnsharpMaskFilter](#), [Digikam::ContentAwareFilter](#), [Digikam::FreeRotationFilter](#), [Digikam::ShearFilter](#), and [Digikam::WBFilter](#).

### 6.402.2.4 filterIdentifier()

```
virtual QString Digikam::DImgThreadedFilter::filterIdentifier( ) const [pure virtual]
```

Return the identifier for this filter in the image history.

Implemented in [Digikam::AutoExpoFilter](#), [Digikam::AutoLevelsFilter](#), [Digikam::EqualizeFilter](#), [Digikam::NormalizeFilter](#), [Digikam::StretchFilter](#), [Digikam::BCGFilter](#), [Digikam::BWSepiaFilter](#), [Digikam::InfraredFilter](#), [Digikam::MixerFilter](#), [Digikam::TonalityFilter](#), [Digikam::CBFilter](#), [Digikam::CurvesFilter](#), [Digikam::BorderFilter](#), [Digikam::TextureFilter](#), [Digikam::FilmFilter](#), [Digikam::FilterActionFilter](#), [Digikam::BlurFilter](#), [Digikam::BlurFXFilter](#), [Digikam::CharcoalFilter](#), [Digikam::ColorFXFilter](#), [Digikam::DistortionFXFilter](#), [Digikam::EmbossFilter](#), [Digikam::FilmGrainFilter](#), [Digikam::InvertFilter](#), [Digikam::OilPaintFilter](#), [Digikam::RainDropFilter](#), [Digikam::GreycstorationFilter](#), [Digikam::HotPixelFixer](#), [Digikam::HSLFilter](#), [Digikam::lccTransformFilter](#), [Digikam::LocalContrastFilter](#), [Digikam::AntiVignettingFilter](#), [Digikam::LensDistortionFilter](#), [Digikam::LensFunFilter](#), [Digikam::LevelsFilter](#), [Digikam::NRFilter](#), [Digikam::RawProcessingFilter](#), [Digikam::RedEyeCorrectionFilter](#), [Digikam::RefocusFilter](#), [Digikam::SharpenFilter](#), [Digikam::UnsharpMaskFilter](#), [Digikam::ContentAwareFilter](#), [Digikam::FreeRotationFilter](#), [Digikam::ShearFilter](#), and [Digikam::WBFilter](#).

### 6.402.2.5 filterImage()

```
virtual void Digikam::DImgThreadedFilter::filterImage ( ) [protected], [pure virtual]
```

Main image filter method. Override in subclass.

Implemented in [Digikam::FilterActionFilter](#), [Digikam::lccTransformFilter](#), [Digikam::RawProcessingFilter](#), and [Digikam::WBFilter](#).

### 6.402.2.6 finished

```
void Digikam::DImgThreadedFilter::finished (
    bool success ) [signal]
```

Emitted when the computation has completed.

## Parameters

<i>success</i>	True if computation finished without interruption on valid data False if the thread was canceled, or no data is available.
----------------	--

**6.402.2.7 initFilter()**

```
void Digikam::DImgThreadedFilter::initFilter ( ) [protected], [virtual]
```

Start filter operation before threaded method. Must be called by your constructor.

**6.402.2.8 initSlave()**

```
void Digikam::DImgThreadedFilter::initSlave (
    DImgThreadedFilter *const master,
    int progressBegin = 0,
    int progressEnd = 100 ) [protected]
```

Initialize the filter for use as a slave - reroutes progress info to master. Note: Computation will be started from [setupFilter\(\)](#).

**6.402.2.9 modulateProgress()**

```
int Digikam::DImgThreadedFilter::modulateProgress (
    int progress ) [protected], [virtual]
```

This method modulates the progress value from the 0..100 span to the span of this slave. Called by postProgress if master is not null.

**6.402.2.10 multithreadedSteps()**

```
QList< int > Digikam::DImgThreadedFilter::multithreadedSteps (
    int stop,
    int start = 0 ) const
```

This method return a list of steps to process parallelized operation in filter using QtConcurrents API. Usually, start and stop are rows or columns from image to process. By default, whole image will be processed and start value is 0. In this case stop will be last row or column to process. Between range [start,stop], this method will divide by equal steps depending of number of CPU cores available. To be sure that all values will be processed, in case of CPU core division give rest, the last step compensate the difference. See Blur filter loop implementation for example to see how to use this method with QtConcurrents API.

**6.402.2.11 parametersSuccessfullyRead()**

```
bool Digikam::DImgThreadedFilter::parametersSuccessfullyRead ( ) const [virtual]
```

Optional: error handling for readParameters. When readParameters() has been called, this method will return true if the call was successful, and false if not. If returning false, readParametersError() will give an error message. The default implementation always returns success. You only need to reimplement when a filter is likely to fail in a different environment, e.g. depending on availability of installed files. These methods have an undefined return value if readParameters() was not called previously.

Reimplemented in [Digikam::lccTransformFilter](#).

#### 6.402.2.12 postProgress()

```
void Digikam::DImgThreadedFilter::postProgress (
    int progress ) [protected]
```

Emit progress info.

#### 6.402.2.13 progress

```
void Digikam::DImgThreadedFilter::progress (
    int progress ) [signal]
```

Emitted when progress info from the calculation is available.

#### 6.402.2.14 run()

```
void Digikam::DImgThreadedFilter::run ( ) [override], [protected], [virtual]
```

List of threaded operations by filter.

Implements [Digikam::DynamicThread](#).

#### 6.402.2.15 setFilterVersion()

```
void Digikam::DImgThreadedFilter::setFilterVersion (
    int version )
```

Replaying a filter action: Set the filter version. A filter may implement different versions, to preserve image history when the algorithm is changed. Any value set here must be contained in supportedVersions, otherwise this call will be ignored. Default value is 1. (Note: If you intend to *record* a filter action, please look at [FilterAction](#)'s m\_version)

#### 6.402.2.16 setSlave()

```
void Digikam::DImgThreadedFilter::setSlave (
    DImgThreadedFilter *const slave ) [protected]
```

Inform the master that there is currently a slave. At destruction of the slave, call with slave=0.

#### 6.402.2.17 setupAndStartDirectly()

```
void Digikam::DImgThreadedFilter::setupAndStartDirectly (
    const DImg & orgImage,
    DImgThreadedFilter *const master,
    int progressBegin = 0,
    int progressEnd = 100 )
```

Initializes the filter for use as a slave and directly starts computation (in-thread)



### 6.402.2.18 setupFilter()

```
void Digikam::DImgThreadedFilter::setupFilter (
    const DImg & orgImage )
```

You need to call this and then start filter if you used the constructor not setting an original image. The original image's data will not be copied.

### 6.402.2.19 started

```
void Digikam::DImgThreadedFilter::started ( ) [signal]
```

This signal is emitted when image data is available and the computation has started.

### 6.402.2.20 startFilter()

```
void Digikam::DImgThreadedFilter::startFilter ( ) [virtual]
```

Start the threaded computation.

### 6.402.2.21 startFilterDirectly()

```
void Digikam::DImgThreadedFilter::startFilterDirectly ( ) [virtual]
```

Start computation of this filter, directly in this thread.

## 6.402.3 Member Data Documentation

### 6.402.3.1 m\_destImage

```
DImg Digikam::DImgThreadedFilter::m_destImage [protected]
```

Output image data.

### 6.402.3.2 m\_master

```
DImgThreadedFilter* Digikam::DImgThreadedFilter::m_master = nullptr [protected]
```

The master of this slave filter. Progress info will be routed to this one.

### 6.402.3.3 m\_name

```
QString Digikam::DImgThreadedFilter::m_name [protected]
```

[Filter](#) name.

#### 6.402.3.4 m\_orgImage

`DImg Digikam::DImgThreadedFilter::m_orgImage` [protected]

Copy of original Image data.

#### 6.402.3.5 m\_progressBegin

`int Digikam::DImgThreadedFilter::m_progressBegin = 0` [protected]

The progress span that a slave filter uses in the parent filter's progress.

#### 6.402.3.6 m\_slave

`DImgThreadedFilter* Digikam::DImgThreadedFilter::m_slave = nullptr` [protected]

The current slave. Any filter might want to use another filter while processing.

## 6.403 Digikam::DImgThreadedFilter::DefaultFilterAction< Filter > Class Template Reference

Inheritance diagram for Digikam::DImgThreadedFilter::DefaultFilterAction< Filter >:



### Public Member Functions

- **DefaultFilterAction** (bool isReproducible)
- **DefaultFilterAction** ([FilterAction::Category](#) category=[FilterAction::ReproducibleFilter](#))
- void [supportOlderVersionIf](#) (int [version](#), bool condition)

## Public Member Functions inherited from [Digikam::FilterAction](#)

- void **addFlag** (Flags flags)
- void **addParameter** (const QString &key, const QVariant &value)
  - Sets parameter, removing all other values for the same key.*
- [Category](#) **category** () const
- void **clearParameters** ()
  - Clear all parameters.*
- QString **description** () const
- QString **displayName** () const
- [FilterAction](#) (const QString &identifier, int version, [Category](#) category=[ReproducibleFilter](#))
- Flags **flags** () const
- bool **hasParameter** (const QString &key) const
- bool **hasParameters** () const
- QString **identifier** () const
- bool **isNull** () const
- bool **operator==** (const [FilterAction](#) &other) const
- QVariant & **parameter** (const QString &key)
- const QVariant **parameter** (const QString &key) const
- template<typename T >
  - T **parameter** (const QString &key) const
- template<typename T >
  - T **parameter** (const QString &key, const T &defaultValue) const
- QHash< QString, QVariant > & **parameters** ()
- const QHash< QString, QVariant > & **parameters** () const
- void **removeFlag** (Flags flags)
- void **removeParameters** (const QString &key)
  - Removes all parameters for key.*
- void **setDescription** (const QString &description)
- void **setDisplayName** (const QString &displayName)
- void **setFlags** (Flags flags)
- void **setParameters** (const QHash< QString, QVariant > &params)
  - Replaces parameters.*
- int **version** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::FilterAction](#)

- enum [Category](#) {
  - [ReproducibleFilter](#) = 0 , [ComplexFilter](#) = 1 , [DocumentedHistory](#) = 2 , **CategoryFirst** = [ReproducibleFilter](#) ,
  - CategoryLast** = [DocumentedHistory](#) }
- enum [Flag](#) { [ExplicitBranch](#) = 1 << 0 }

## Protected Attributes inherited from [Digikam::FilterAction](#)

- [Category](#) **m\_category** = [ReproducibleFilter](#)
  - NOTE: Value class, do not create a d-pointer.*
- QString **m\_description**
- QString **m\_displayableName**
- Flags **m\_flags**
- QString **m\_identifier**
- QHash< QString, QVariant > **m\_params**
- int **m\_version** = 0

## 6.403.1 Detailed Description

```
template<class Filter>
class Digikam::DImgThreadedFilter::DefaultFilterAction< Filter >
```

Convenience class to spare the few repeating lines of code

## 6.403.2 Member Function Documentation

### 6.403.2.1 supportOlderVersionIf()

```
template<class Filter >
void Digikam::DImgThreadedFilter::DefaultFilterAction< Filter >::supportOlderVersionIf (
    int version,
    bool condition ) [inline]
```

Preserve backwards compatibility: If a given condition (some new feature is not used) is true, decrease the version so that older digikam versions can still replay the action

## 6.404 Digikam::DInfoInterface Class Reference

Inheritance diagram for Digikam::DInfoInterface:



### Public Types

- typedef `QList< int >` **DAAlbumIDs**  
List of *Album ids*.

- typedef QMap< QString, QVariant > **DInfoMap**  
*Map of properties name and value.*
- enum **SetupPage** { **ExifToolPage** = 0 , **ImageQualityPage** }

### Public Member Functions

- virtual void **deleteImage** (const QUrl &url)  
*Manipulate with item.*
  - **DInfoInterface** (QObject \*const parent)
  - virtual void **openSetupPage** (SetupPage page)  
*Open configuration dialog page.*
  - virtual QMap< QString, QString > **passShortcutActionsToWidget** (QWidget \*const) const  
*Pass extra shortcut actions to widget and return prefixes of shortcuts.*
  - Q\_SIGNAL void **signalAlbumItemsRecursiveCompleted** (const QList< QUrl > &imageList)
  - Q\_SIGNAL void **signalSetupChanged** ()
  - Q\_SIGNAL void **signalShortcutPressed** (const QString &shortcut, int val)
  - virtual Q\_SLOT void **slotDateTimeForUrl** (const QUrl &url, const QDateTime &dt, bool updModDate)  
*Slot to call when date time stamp from item is changed.*
  - virtual Q\_SLOT void **slotMetadataChangedForUrl** (const QUrl &url)  
*Slot to call when something in metadata from item is changed.*
  - virtual QAbstractItemModel \* **tagFilterModel** ()  
*Return an instance of tag filter model if host application support this feature, else null pointer.*
- 
- virtual QList< QUrl > **currentSelectedItems** () const  
*Low level items and albums methods.*
  - virtual QList< QUrl > **currentAlbumItems** () const
  - virtual QUrl **currentActiveItem** () const
  - virtual void **parseAlbumItemsRecursive** ()
  - virtual QList< QUrl > **albumItems** (int) const
  - virtual QList< QUrl > **albumsItems** (const DAlbumIDs &) const
  - virtual QList< QUrl > **allAlbumItems** () const
  - virtual **DInfoMap** **albumInfo** (int) const
  - virtual void **setAlbumInfo** (int, const **DInfoMap** &) const
  - virtual **DInfoMap** **itemInfo** (const QUrl &) const
  - virtual void **setItemInfo** (const QUrl &, const **DInfoMap** &)
  - Q\_SIGNAL void **signalLastItemUrl** (const QUrl &)
- 
- virtual QWidget \* **albumChooser** (QWidget \*const parent) const  
*Albums chooser view methods (to use items from albums before to process).*
  - virtual **DAlbumIDs** **albumChooserItems** () const
  - virtual bool **supportAlbums** () const
  - Q\_SIGNAL void **signalAlbumChooserSelectionChanged** ()
- 
- virtual QWidget \* **uploadWidget** (QWidget \*const parent) const  
*Album selector view methods (to upload items from an external place).*
  - virtual QUrl **uploadUrl** () const
  - Q\_SIGNAL void **signalUploadUrlChanged** ()
  - virtual QUrl **defaultUploadUrl** () const  
*Url to upload new items without to use album selector.*
  - Q\_SIGNAL void **signalImportedImage** (const QUrl &)

## Public Attributes

- bool **forceAlbumSelection** = false

## 6.404.1 Member Function Documentation

### 6.404.1.1 albumChooser()

```
QWidget * Digikam::DInfoInterface::albumChooser (
    QWidget *const parent ) const [virtual]
```

Reimplemented in [Digikam::DBInfoface](#).

### 6.404.1.2 currentSelectedItems()

```
QList< QUrl > Digikam::DInfoInterface::currentSelectedItems ( ) const [virtual]
```

Reimplemented in [Digikam::DBInfoface](#), and [Digikam::DMetaInfoface](#).

### 6.404.1.3 defaultUploadUrl()

```
QUrl Digikam::DInfoInterface::defaultUploadUrl ( ) const [virtual]
```

Reimplemented in [Digikam::DBInfoface](#), and [Digikam::DMetaInfoface](#).

### 6.404.1.4 deleteImage()

```
void Digikam::DInfoInterface::deleteImage (
    const QUrl & url ) [virtual]
```

Reimplemented in [Digikam::DBInfoface](#), and [Digikam::DMetaInfoface](#).

### 6.404.1.5 openSetupPage()

```
void Digikam::DInfoInterface::openSetupPage (
    SetupPage page ) [virtual]
```

Reimplemented in [Digikam::DBInfoface](#), and [ShowFoto::ShowfotoInfoface](#).

### 6.404.1.6 passShortcutActionsToWidget()

```
QMap< QString, QString > Digikam::DInfoInterface::passShortcutActionsToWidget (
    QWidget * const ) const [virtual]
```

Reimplemented in [Digikam::DBInfoface](#).



#### 6.404.1.7 slotDateTimeForUrl()

```
void Digikam::DInfoInterface::slotDateTimeForUrl (
    const QUrl & url,
    const QDateTime & dt,
    bool updModDate ) [virtual]
```

Reimplemented in [Digikam::DMetaInfoInterface](#).

#### 6.404.1.8 slotMetadataChangedForUrl()

```
void Digikam::DInfoInterface::slotMetadataChangedForUrl (
    const QUrl & url ) [virtual]
```

Reimplemented in [Digikam::DMetaInfoInterface](#).

#### 6.404.1.9 tagFilterModel()

```
QAbstractItemModel * Digikam::DInfoInterface::tagFilterModel ( ) [virtual]
```

Reimplemented in [Digikam::DBInfoInterface](#).

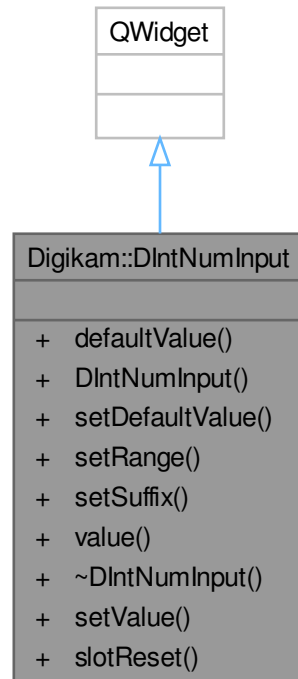
#### 6.404.1.10 uploadWidget()

```
QWidget * Digikam::DInfoInterface::uploadWidget (
    QWidget *const parent ) const [virtual]
```

Reimplemented in [Digikam::DBInfoInterface](#), and [Digikam::DMetaInfoInterface](#).

## 6.405 Digikam::DIntNumInput Class Reference

Inheritance diagram for Digikam::DIntNumInput:



### Public Slots

- void **setValue** (int d)
- void **slotReset** ()

### Signals

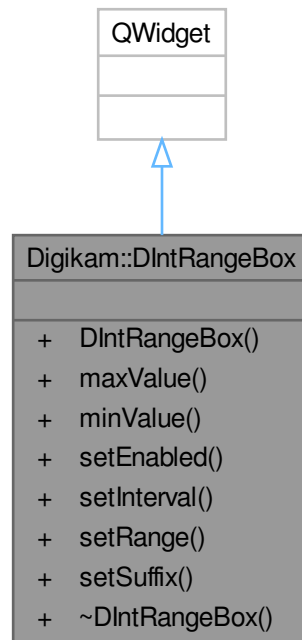
- void **reset** ()
- void **valueChanged** (int)

### Public Member Functions

- int **defaultValue** () const
- **DIntNumInput** (QWidget \*const parent=nullptr)
- void **setDefaultValue** (int d)
- void **setRange** (int min, int max, int step)
- void **setSuffix** (const QString &suffix)
- int **value** () const

## 6.406 Digikam::DIntRangeBox Class Reference

Inheritance diagram for Digikam::DIntRangeBox:



### Signals

- void `maxChanged` (int)
- void `minChanged` (int)

### Public Member Functions

- `DIntRangeBox` (`QWidget *const parent=nullptr`)
- int `maxValue` ()
- int `minValue` ()
- void `setEnabled` (bool enabled)
- void `setInterval` (int min, int max)
- void `setRange` (int min, int max)
- void `setSuffix` (const `QString` &suffix)

### 6.406.1 Member Function Documentation

#### 6.406.1.1 `maxValue()`

```
int Digikam::DIntRangeBox::maxValue ( )
```

This method returns the maximum value of the interval.

#### Returns

the maximum value.

### 6.406.1.2 `minValue()`

```
int Digikam::DIntRangeBox::minValue ( )
```

This method returns the minimum value of the interval.

#### Returns

the minimum value.

### 6.406.1.3 `setEnabled()`

```
void Digikam::DIntRangeBox::setEnabled (
    bool enabled )
```

This method enables or disables the embedded spinboxes.

#### Parameters

<i>enabled</i>	If the interval boxes should be enabled.
----------------	--

### 6.406.1.4 `setInterval()`

```
void Digikam::DIntRangeBox::setInterval (
    int min,
    int max )
```

This method sets the minimum and maximum of the interval.

#### Parameters

<i>min</i>	The minimum value of the interval.
<i>max</i>	The maximum value of the interval.

### 6.406.1.5 `setRange()`

```
void Digikam::DIntRangeBox::setRange (
    int min,
    int max )
```

This method sets the lower and upper threshold of possible interval minimum and maximum values.

#### Parameters

<i>min</i>	the lowest value to which the interval can be expanded.
<i>max</i>	the highest value to which the interval can be expanded.

### 6.406.1.6 setSuffix()

```
void Digikam::DIntRangeBox::setSuffix (
    const QString & suffix )
```

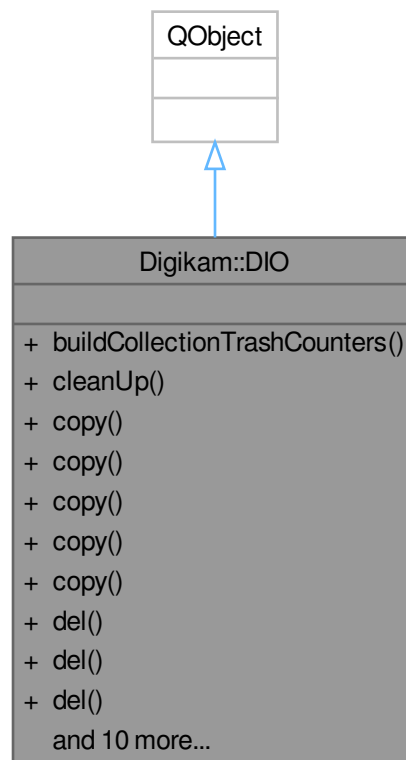
This method sets the suffix for the minimum and maximum value boxes.

#### Parameters

<i>suffix</i>	The suffix.
---------------	-------------

## 6.407 Digikam::DIO Class Reference

Inheritance diagram for Digikam::DIO:



#### Signals

- void **signalRenameFailed** (const QUrl &url)
- void **signalRenameFinished** ()
- void **signalTrashCounters** ()
- void **signalTrashFinished** ()

## Static Public Member Functions

- static void **buildCollectionTrashCounters** ()
- static void **cleanUp** ()
- static void **copy** (const QList< [ItemInfo](#) > &infos, const QUrl &dest)  
*Copy items to external folder.*
- static void **copy** (const QList< [ItemInfo](#) > &infos, [PAlbum](#) \*const dest)  
*Copy items to another album.*
- static void **copy** (const QList< QUrl > &srcList, [PAlbum](#) \*const dest)  
*Copy external files to another album.*
- static void **copy** (const QUrl &src, [PAlbum](#) \*const dest)  
*Copy an external file to another album.*
- static void **copy** ([PAlbum](#) \*const src, [PAlbum](#) \*const dest)  
*Copy an album to another album.*
- static void **del** (const [ItemInfo](#) &info, bool useTrash)
- static void **del** (const QList< [ItemInfo](#) > &infos, bool useTrash)
- static void **del** ([PAlbum](#) \*const album, bool useTrash)
- static void **emptyTrash** (const DTrashItemInfoList &infos)
- static int **getTrashCounter** (const QString &albumRootPath)  
*Trash operations.*
- static [DIO](#) \* **instance** ()
- static bool **itemsUnderProcessing** ()
- static void **move** (const QList< [ItemInfo](#) > &infos, [PAlbum](#) \*const dest)  
*Move items to another album.*
- static void **move** (const QList< QUrl > &srcList, [PAlbum](#) \*const dest)  
*Move external files into another album.*
- static void **move** (const QUrl &src, [PAlbum](#) \*const dest)  
*Move external files another album.*
- static void **move** ([PAlbum](#) \*const src, [PAlbum](#) \*const dest)  
*Move an album into another album.*
- static void **rename** (const QUrl &src, const QString &newName, bool overwrite=false)  
*Rename item to new name.*
- static void **restoreTrash** (const DTrashItemInfoList &infos)

## Friends

- class [DIOCreator](#)

## 6.407.1 Member Function Documentation

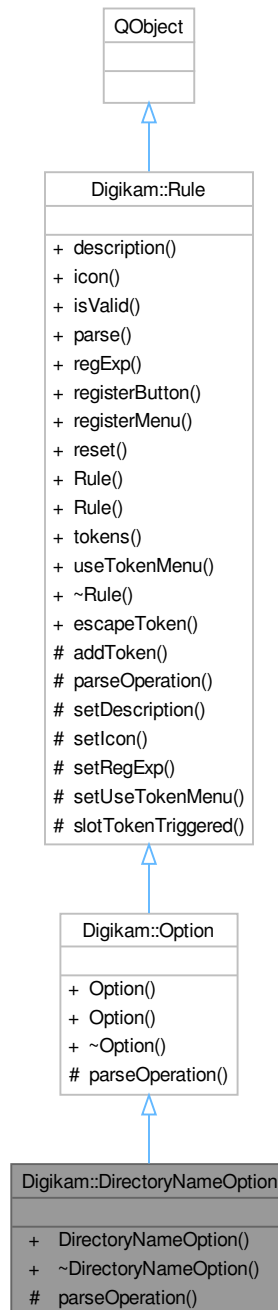
### 6.407.1.1 copy()

```
void Digikam::DIO::copy (
    PAlbum *const src,
    PAlbum *const dest ) [static]
```

All [DIO](#) methods will take care for sidecar files, if they exist

## 6.408 Digikam::DirectoryNameOption Class Reference

Inheritance diagram for Digikam::DirectoryNameOption:



### Protected Member Functions

- QString [parseOperation](#) ([ParseSettings](#) &settings, const QRegularExpressionMatch &match) override

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void [setDescription](#) (const QString &desc)
- void [setIcon](#) (const QString &pixmap)
- void [setRegExp](#) (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum [IconType](#) { [Action](#) = 0 , [Dialog](#) }

## Signals inherited from [Digikam::Rule](#)

- void [signalTokenTriggered](#) (const QString &)

## Public Member Functions inherited from [Digikam::Option](#)

- [Option](#) (const QString &name, const QString &description)
- [Option](#) (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString [description](#) () const
- QPixmap [icon](#) (Rule::IconType type=Rule::Action) const
- bool [isValid](#) () const
- [ParseResults](#) [parse](#) ([ParseSettings](#) &settings)
- QRegularExpression & [regExp](#) () const
- QPushButton \* [registerButton](#) (QWidget \*parent)
- QAction \* [registerMenu](#) (QMenu \*parent)
- virtual void [reset](#) ()
- [Rule](#) (const QString &name)
- [Rule](#) (const QString &name, const QString &icon)
- TokenList & [tokens](#) () const
- bool [useTokenMenu](#) () const

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString [escapeToken](#) (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void [slotTokenTriggered](#) (const QString &)

## 6.408.1 Member Function Documentation

### 6.408.1.1 [parseOperation\(\)](#)

```
QString Digikam::DirectoryNameOption::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [virtual]
```

TODO: describe me



## Parameters

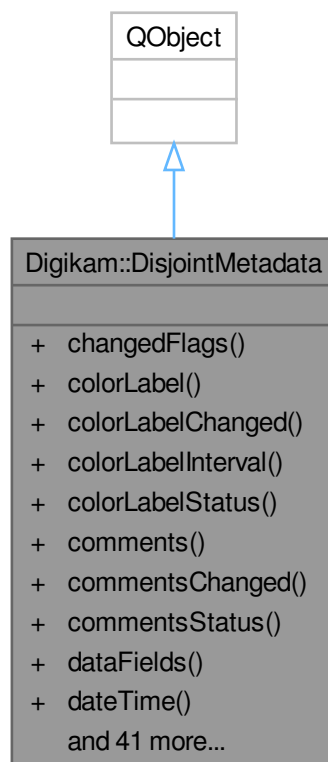
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Option](#).

## 6.409 Digikam::DisjointMetadata Class Reference

Inheritance diagram for Digikam::DisjointMetadata:



## Classes

- class [Private](#)

## Public Types

- enum [WriteMode](#) { [FullWrite](#) , [FullWriteIfChanged](#) , [PartialWrite](#) }

## Public Member Functions

- int [changedFlags](#) ()  
*changedFlags - used for selective metadata write. The result will be passed to metadatahub and it will*
- int [colorLabel](#) () const
- bool [colorLabelChanged](#) () const
- void [colorLabelInterval](#) (int &lowest, int &highest) const
- [DisjointMetadataDataFields::Status](#) [colorLabelStatus](#) () const
- [CaptionsMap](#) [comments](#) () const
- bool [commentsChanged](#) () const
- [DisjointMetadataDataFields::Status](#) [commentsStatus](#) () const
- [DisjointMetadataDataFields](#) [dataFields](#) () const
- QDateTime [dateTime](#) () const
- bool [dateTimeChanged](#) () const
- void [dateTimeInterval](#) (QDateTime &lowest, QDateTime &highest) const
- [DisjointMetadataDataFields::Status](#) [dateTimeStatus](#) () const
- QStringList [keywords](#) () const
- void [load](#) (const [ItemInfo](#) &info)
- [Template](#) [metadataTemplate](#) () const
- int [pickLabel](#) () const
- bool [pickLabelChanged](#) () const
- void [pickLabelInterval](#) (int &lowest, int &highest) const
- [DisjointMetadataDataFields::Status](#) [pickLabelStatus](#) () const
- int [rating](#) () const
- bool [ratingChanged](#) () const
- void [ratingInterval](#) (int &lowest, int &highest) const
- [DisjointMetadataDataFields::Status](#) [ratingStatus](#) () const
- void [replaceColorLabel](#) (int colorId)
- void [replacePickLabel](#) (int pickId)
- void [replaceRating](#) (int rating)
- void [reset](#) ()
- void [resetChanged](#) ()
- void [setColorLabel](#) (int colorId, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- void [setComments](#) (const [CaptionsMap](#) &comments, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- void [setDataFields](#) (const [DisjointMetadataDataFields](#) &data)
- void [setDateTime](#) (const QDateTime &dateTime, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- void [setMetadataTemplate](#) (const [Template](#) &t, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- void [setPickLabel](#) (int pickId, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- void [setRating](#) (int rating, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- void [setTag](#) (int albumId, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- void [setTitles](#) (const [CaptionsMap](#) &titles, [DisjointMetadataDataFields::Status](#) status=[DisjointMetadataDataFields::MetadataAvailable](#))
- QMap< int, [DisjointMetadataDataFields::Status](#) > [tags](#) () const
- bool [tagsChanged](#) () const
- [DisjointMetadataDataFields::Status](#) [tagStatus](#) (const QString &tagPath) const
- [DisjointMetadataDataFields::Status](#) [tagStatus](#) (int albumId) const
- bool [templateChanged](#) () const
- [DisjointMetadataDataFields::Status](#) [templateStatus](#) () const
- [CaptionsMap](#) [titles](#) () const
- bool [titlesChanged](#) () const
- [DisjointMetadataDataFields::Status](#) [titlesStatus](#) () const
- bool [willWriteMetadata](#) ([WriteMode](#) writeMode, const [MetaEngineSettingsContainer](#) &settings=[MetaEngineSettings::instance](#)() ->settings()) const
- bool [write](#) ([ItemInfo](#) info, [WriteMode](#) writeMode=[FullWrite](#))

## 6.409.1 Member Enumeration Documentation

### 6.409.1.1 WriteMode

enum `Digikam::DisjointMetadata::WriteMode`

Enumerator

FullWrite	Write all available information
FullWriteIfChanged	Do a full write if and only if <ul style="list-style-type: none"> <li>• metadata fields changed</li> <li>• the changed fields shall be written according to write settings "Changed" in this context means changed by one of the set... methods, the load() methods are ignored for this attribute. This mode allows to avoid write operations when e.g. the user does not want keywords to be written and only changes keywords.</li> </ul>
PartialWrite	Write only the changed parts. Metadata fields which cannot be changed from <a href="#">MetadataHub</a> (photographer ID etc.) will never be written

## 6.409.2 Member Function Documentation

### 6.409.2.1 changedFlags()

```
int Digikam::DisjointMetadata::changedFlags ( )
```

- write it to disk

Returns

- metadatahub flags encoded as int

### 6.409.2.2 colorLabel()

```
int Digikam::DisjointMetadata::colorLabel ( ) const
```

Returns the Color Label id (see ColorLabel values in globals.h). If status is [DisjointMetadataDataFields::MetadataDisjoint](#), the None Label is returned. (see [colorLabelInterval\(\)](#)) If status is [DisjointMetadataDataFields::MetadataInvalid](#), -1 is returned.

### 6.409.2.3 colorLabelInterval()

```
void Digikam::DisjointMetadata::colorLabelInterval (
    int & lowest,
    int & highest ) const
```

Returns the lowest and highest Color Label id (see ColorLabel values from globals.h). If status is [DisjointMetadataDataFields::MetadataAvailable](#), the values are the same. If status is [DisjointMetadataDataFields::MetadataInvalid](#), -1 is returned.

#### 6.409.2.4 comments()

```
CaptionsMap Digikam::DisjointMetadata::comments ( ) const
```

Returns a map all alternate language omments . If status is [DisjointMetadataDataFields::MetadataDisjoint](#), the first loaded map is returned. If status is [DisjointMetadataDataFields::MetadataInvalid](#), CaptionMap() is returned.

#### 6.409.2.5 dateTime()

```
QDateTime Digikam::DisjointMetadata::dateTime ( ) const
```

Returns the dateTime. If status is [DisjointMetadataDataFields::MetadataDisjoint](#), the earliest date is returned. (see [dateTimeInterval\(\)](#)) If status is [DisjointMetadataDataFields::MetadataInvalid](#), an invalid date is returned.

#### 6.409.2.6 dateTimeChanged()

```
bool Digikam::DisjointMetadata::dateTimeChanged ( ) const
```

Returns if the metadata field has been changed with the corresponding setter method.

#### 6.409.2.7 dateTimeInterval()

```
void Digikam::DisjointMetadata::dateTimeInterval (
    QDateTime & lowest,
    QDateTime & highest ) const
```

Returns the earliest and latest date. If status is [DisjointMetadataDataFields::MetadataAvailable](#), the values are the same. If status is [DisjointMetadataDataFields::MetadataInvalid](#), invalid dates are returned.

#### 6.409.2.8 dateTimeStatus()

```
DisjointMetadataDataFields::Status Digikam::DisjointMetadata::dateTimeStatus ( ) const
```

Returns the metadata field Status.

#### 6.409.2.9 keywords()

```
QStringList Digikam::DisjointMetadata::keywords ( ) const
```

Returns a QStringList with all tags with status [DisjointMetadataDataFields::MetadataAvailable](#). (i.e., the intersection of tags from all loaded metadata sets)

#### 6.409.2.10 metadataTemplate()

```
Template Digikam::DisjointMetadata::metadataTemplate ( ) const
```

Returns the metadata template. If status is [DisjointMetadataDataFields::MetadataDisjoint](#), the first loaded template is returned. If status is [DisjointMetadataDataFields::MetadataInvalid](#), 0 is returned.

### 6.409.2.11 pickLabel()

```
int Digikam::DisjointMetadata::pickLabel ( ) const
```

Returns the Pick Label id (see PickLabel values in globals.h). If status is [DisjointMetadataDataFields::MetadataDisjoint](#), the None Label is returned. (see [pickLabelInterval\(\)](#)) If status is [DisjointMetadataDataFields::MetadataInvalid](#), -1 is returned.

### 6.409.2.12 pickLabelInterval()

```
void Digikam::DisjointMetadata::pickLabelInterval (
    int & lowest,
    int & highest ) const
```

Returns the lowest and highest Pick Label id (see PickLabel values from globals.h). If status is [DisjointMetadataDataFields::MetadataAvailable](#), the values are the same. If status is [DisjointMetadataDataFields::MetadataInvalid](#), -1 is returned.

### 6.409.2.13 rating()

```
int Digikam::DisjointMetadata::rating ( ) const
```

Returns the rating. If status is [DisjointMetadataDataFields::MetadataDisjoint](#), the lowest rating is returned. (see [ratingInterval\(\)](#)) If status is [DisjointMetadataDataFields::MetadataInvalid](#), -1 is returned.

### 6.409.2.14 ratingInterval()

```
void Digikam::DisjointMetadata::ratingInterval (
    int & lowest,
    int & highest ) const
```

Returns the lowest and highest rating. If status is [DisjointMetadataDataFields::MetadataAvailable](#), the values are the same. If status is [DisjointMetadataDataFields::MetadataInvalid](#), -1 is returned.

### 6.409.2.15 replaceColorLabel()

```
void Digikam::DisjointMetadata::replaceColorLabel (
    int colorId )
```

Special case if the metadata of color, pick or rating has already been changed outside. Replace with current values as if there is no change.

### 6.409.2.16 setDateTime()

```
void Digikam::DisjointMetadata::setDateTime (
    const QDateTime & dateTime,
    DisjointMetadataDataFields::Status status = DisjointMetadataDataFields::MetadataAvailable
)
```

Set metadata field to the given value, and the metadata field status to the corresponding [DisjointMetadataDataFields::MetadataAvailable](#)

**6.409.2.17 tags()**

```
QMap< int, DisjointMetadataDataFields::Status > Digikam::DisjointMetadata::tags ( ) const
```

Returns a map with the status for each tag. Any tag that was set on one of the loaded images is contained in the map. (If a tag is not contained in the map, it was not set on any of the loaded images) If the tag was set on all loaded images, the status is [DisjointMetadataDataFields::MetadataAvailable](#). If the tag was set on at least one, but not all of the loaded images, the status is [DisjointMetadataDataFields::MetadataDisjoint](#).

**6.409.2.18 titles()**

```
CaptionsMap Digikam::DisjointMetadata::titles ( ) const
```

Returns a map all alternate language titles. If status is [DisjointMetadataDataFields::MetadataDisjoint](#), the first loaded map is returned. If status is [DisjointMetadataDataFields::MetadataInvalid](#), CaptionMap() is returned.

**6.409.2.19 willWriteMetadata()**

```
bool Digikam::DisjointMetadata::willWriteMetadata (
    DisjointMetadata::WriteMode writeMode,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
) const
```

With the currently applied changes, the given writeMode and settings, returns if write(DMetadata), write(QString) or write(DImg) will actually apply any changes.

**6.409.2.20 write()**

```
bool Digikam::DisjointMetadata::write (
    ItemInfo info,
    WriteMode writeMode = FullWrite )
```

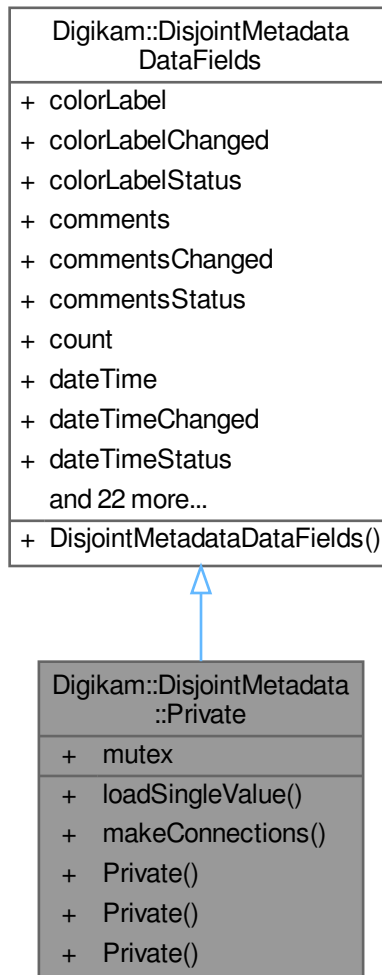
Applies the set of metadata contained in this [MetadataHub](#) to the given [ItemInfo](#) object.

**Returns**

Returns true if the info object has been changed

## 6.410 Digikam::DisjointMetadata::Private Class Reference

Inheritance diagram for Digikam::DisjointMetadata::Private:



### Public Member Functions

- `template<class T >`  
void **loadSingleValue** (const T &data, T &storage, [DisjointMetadataDataFields::Status](#) &status)
- void **makeConnections** ([DisjointMetadata](#) \*const q)
- **Private** (const [DisjointMetadataDataFields](#) &other)
- **Private** (const [Private](#) &other)

### Public Attributes

- QMutex **mutex**

## Public Attributes inherited from [Digikam::DisjointMetadataDataFields](#)

- int **colorLabel** = -1
- bool **colorLabelChanged** = false
- [Status](#) **colorLabelStatus** = [MetadataInvalid](#)
- [CaptionsMap](#) **comments**
- bool **commentsChanged** = false
- [Status](#) **commentsStatus** = [MetadataInvalid](#)
- int **count** = 0
- QDateTime **dateTime**
- bool **dateTimeChanged** = false
- [Status](#) **dateTimeStatus** = [MetadataInvalid](#)
- int **highestColorLabel** = -1
- int **highestPickLabel** = -1
- int **highestRating** = -1
- bool **invalid** = false
- QDateTime **lastDateTime**
- [Template](#) **metadataTemplate**
- int **pickLabel** = -1
- bool **pickLabelChanged** = false
- [Status](#) **pickLabelStatus** = [MetadataInvalid](#)
- int **rating** = -1
- bool **ratingChanged** = false
- [Status](#) **ratingStatus** = [MetadataInvalid](#)
- QList< int > **tagIds**
- QStringList **tagList**
- QMap< int, [Status](#) > **tags**
- bool **tagsChanged** = false
- bool **templateChanged** = false
- [Status](#) **templateStatus** = [MetadataInvalid](#)
- [CaptionsMap](#) **titles**
- bool **titlesChanged** = false
- [Status](#) **titlesStatus** = [MetadataInvalid](#)
- bool **withoutTags** = false

## Additional Inherited Members

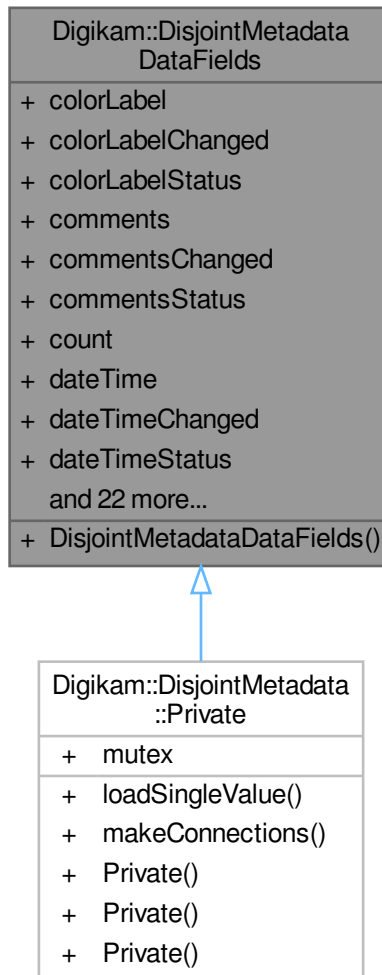
## Public Types inherited from [Digikam::DisjointMetadataDataFields](#)

- enum [Status](#) { [MetadataInvalid](#) , [MetadataAvailable](#) , [MetadataDisjoint](#) }



## 6.411 Digikam::DisjointMetadataDataFields Class Reference

Inheritance diagram for Digikam::DisjointMetadataDataFields:



### Public Types

- enum [Status](#) { [MetadataInvalid](#) , [MetadataAvailable](#) , [MetadataDisjoint](#) }

### Public Attributes

- int `colorLabel` = -1
- bool `colorLabelChanged` = false
- [Status](#) `colorLabelStatus` = [MetadataInvalid](#)
- [CaptionsMap](#) `comments`
- bool `commentsChanged` = false

- [Status](#) `commentsStatus` = [MetadataInvalid](#)
- int `count` = 0
- QDateTime `dateTime`
- bool `dateTimeChanged` = false
- [Status](#) `dateTimeStatus` = [MetadataInvalid](#)
- int `highestColorLabel` = -1
- int `highestPickLabel` = -1
- int `highestRating` = -1
- bool `invalid` = false
- QDateTime `lastDateTime`
- [Template](#) `metadataTemplate`
- int `pickLabel` = -1
- bool `pickLabelChanged` = false
- [Status](#) `pickLabelStatus` = [MetadataInvalid](#)
- int `rating` = -1
- bool `ratingChanged` = false
- [Status](#) `ratingStatus` = [MetadataInvalid](#)
- QList< int > `tagIds`
- QStringList `tagList`
- QMap< int, [Status](#) > `tags`
- bool `tagsChanged` = false
- bool `templateChanged` = false
- [Status](#) `templateStatus` = [MetadataInvalid](#)
- [CaptionsMap](#) `titles`
- bool `titlesChanged` = false
- [Status](#) `titlesStatus` = [MetadataInvalid](#)
- bool `withoutTags` = false

### 6.411.1 Detailed Description

This class was split from [DisjointMetadata::Private](#) to allow to use the automatic C++ copy constructor ([DisjointMetadata::Private](#) contains a QMutex and is thus non-copyable)

### 6.411.2 Member Enumeration Documentation

#### 6.411.2.1 Status

```
enum Digikam::DisjointMetadataDataFields::Status
```

The status enum describes the result of joining several metadata sets. If only one set has been added, the status is always `MetadataAvailable`. If no set has been added, the status is always `MetadataInvalid`

#### Enumerator

<code>MetadataInvalid</code>	Not yet filled with any value.
<code>MetadataAvailable</code>	Only one data set has been added, or a common value is available.
<code>MetadataDisjoint</code>	No common value is available. For rating and dates, the interval is available.

## 6.412 Digikam::DistortionFXFilter Class Reference

Inheritance diagram for Digikam::DistortionFXFilter:



### Public Types

- enum **DistortionFXTypes** {  
**FishEye =0** , **Twirl** , **CilindricalHor** , **CilindricalVert** ,

**CylindricalHV** , **Caricature** , **MultipleCorners** , **WavesHorizontal** , **WavesVertical** , **BlockWaves1** , **BlockWaves2** , **CircularWaves1** , **CircularWaves2** , **PolarCoordinates** , **UnpolarCoordinates** , **Tile** }

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Member Functions

- **DistortionFXFilter** ([DImg](#) \*const orgImage, [QObject](#) \*const parent=nullptr, int effectType=0, int level=0, int iteration=0, bool antialiasing=true)
- **DistortionFXFilter** ([QObject](#) \*const parent=nullptr)
- [FilterAction](#) filterAction () override
- [QString](#) filterIdentifier () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > **supportedVersions** () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

### Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

### Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

### Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- [DImg m\\_destImage](#)
- [DImgThreadedFilter \\* m\\_master](#) = nullptr
- [QString m\\_name](#)
- [DImg m\\_orgImage](#)
- [int m\\_progressBegin](#) = 0
- [int m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- [int m\\_progressSpan](#) = 0
- [DImgThreadedFilter \\* m\\_slave](#) = nullptr
- [int m\\_version](#) = 1
- [bool m\\_wasCancelled](#) = false

### 6.412.1 Member Function Documentation

#### 6.412.1.1 filterAction()

```
FilterAction Digikam::DistortionFXFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

#### 6.412.1.2 filterIdentifier()

```
QString Digikam::DistortionFXFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

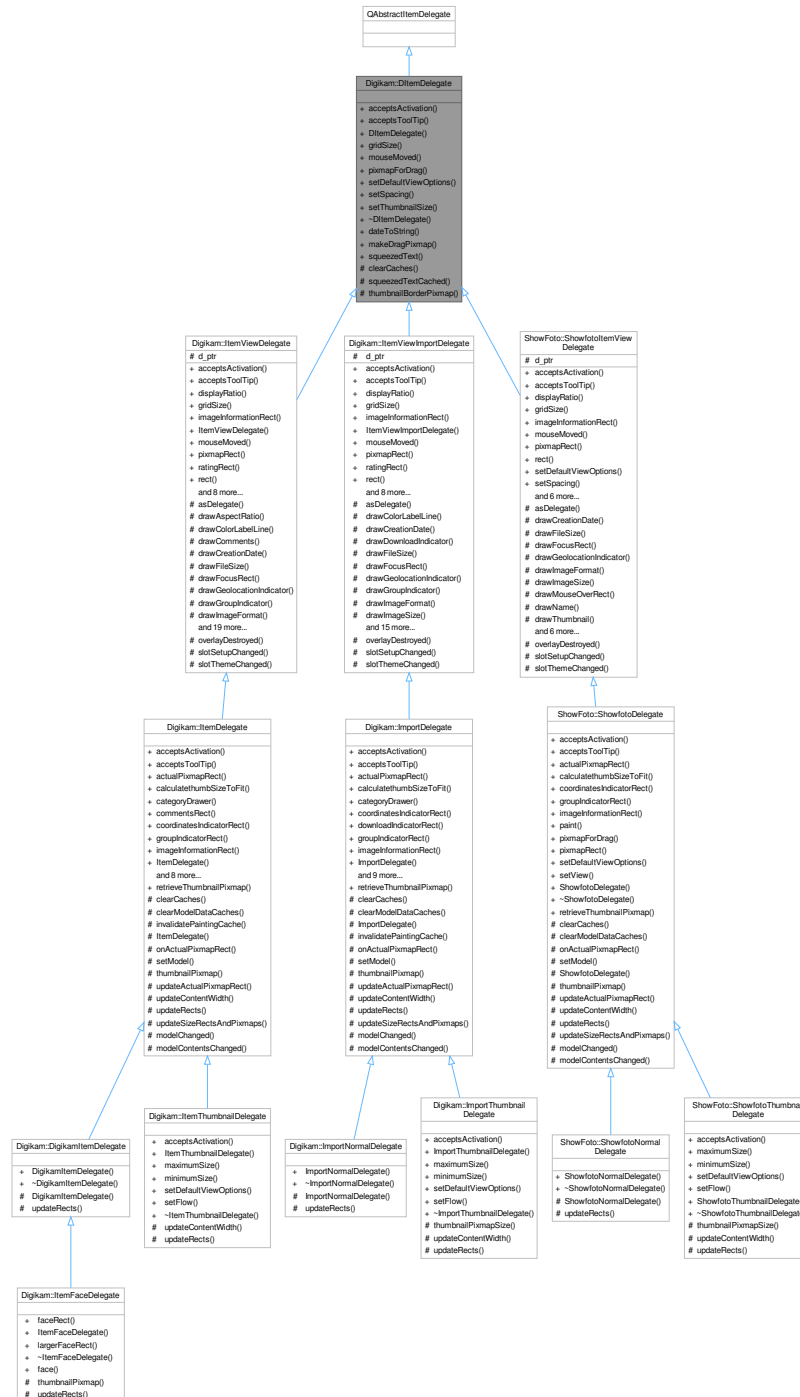
#### 6.412.1.3 readParameters()

```
void Digikam::DistortionFXFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.413 Digikam::DItemDelegate Class Reference

Inheritance diagram for Digikam::DItemDelegate:



### Signals

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Public Member Functions

- virtual bool **acceptsActivation** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const =0
- virtual bool **acceptsToolTip** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const =0
- **DItemDelegate** (QObject \*const parent=nullptr)
- virtual QSize **gridSize** () const =0
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)=0
- virtual QPixmap **pixmapForDrag** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const =0
- virtual void **setDefaultViewOptions** (const QStyleOptionViewItem &option)=0
- virtual void **setSpacing** (int spacing)=0
- virtual void **setThumbnailSize** (const ThumbnailSize &thumbSize)=0

## Static Public Member Functions

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

## Protected Member Functions

- virtual void **clearCaches** ()
- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

## 6.413.1 Member Function Documentation

### 6.413.1.1 acceptsToolTip()

```
virtual bool Digikam::DItemDelegate::acceptsToolTip (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * tooltipRect = nullptr ) const [pure virtual]
```

These methods take four parameters: The position on viewport, the rect on viewport, the index, and optionally a parameter into which, if the return value is true, a rectangle can be written for which the return value will be true as well.

Implemented in [Digikam::ItemDelegate](#), [Digikam::ItemViewDelegate](#), [ShowFoto::ShowfotoDelegate](#), [ShowFoto::ShowfotoItemViewDe](#), [Digikam::ImportDelegate](#), and [Digikam::ItemViewImportDelegate](#).

### 6.413.1.2 gridSize()

```
virtual QSize Digikam::DItemDelegate::gridSize ( ) const [pure virtual]
```

Returns the gridsizes to be set by the view. It's sizeHint plus spacing.

Implemented in [Digikam::ItemViewDelegate](#), [ShowFoto::ShowfotoItemViewDelegate](#), and [Digikam::ItemViewImportDelegate](#).



### 6.413.1.3 mouseMoved()

```
virtual void Digikam::DItemDelegate::mouseMoved (
    QMouseEvent * e,
    const QRect & visualRect,
    const QModelIndex & index ) [pure virtual]
```

#### Note

to be called by [ItemViewCategorized](#) only

Implemented in [Digikam::ItemViewDelegate](#), [ShowFoto::ShowfotoItemViewDelegate](#), and [Digikam::ItemViewImportDelegate](#).

### 6.413.1.4 setDefaultViewOptions()

```
virtual void Digikam::DItemDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [pure virtual]
```

Style option with standard values to use for cached rendering. option.rect shall be the viewport rectangle. Call on resize, font change.

Implemented in [Digikam::ItemDelegate](#), [Digikam::ItemThumbnailDelegate](#), [Digikam::ItemViewDelegate](#), [ShowFoto::ShowfotoDelegate](#), [ShowFoto::ShowfotoThumbnailDelegate](#), [ShowFoto::ShowfotoItemViewDelegate](#), [Digikam::ImportDelegate](#), [Digikam::ImportThumbnailDelegate](#), and [Digikam::ItemViewImportDelegate](#).

### 6.413.1.5 setThumbnailSize()

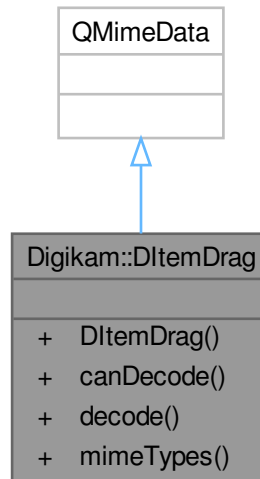
```
virtual void Digikam::DItemDelegate::setThumbnailSize (
    const ThumbnailSize & thumbSize ) [pure virtual]
```

You must set these options from the view

Implemented in [Digikam::ItemViewDelegate](#), [ShowFoto::ShowfotoItemViewDelegate](#), and [Digikam::ItemViewImportDelegate](#).

## 6.414 Digikam::DItemDrag Class Reference

Inheritance diagram for Digikam::DItemDrag:



### Public Member Functions

- **DItemDrag** (const QList< QUrl > &urls, const QList< int > &albumIDs, const QList< qlonglong > &imageIDs)

### Static Public Member Functions

- static bool **canDecode** (const QMimeData \*e)
- static bool **decode** (const QMimeData \*e, QList< QUrl > &urls, QList< int > &albumIDs, QList< qlonglong > &imageIDs)
- static QStringList **mimeTypeypes** ()

### 6.414.1 Detailed Description

Provides a drag object with additional information for internal drag&drop

Images can be moved through `ItemDrag`. It is possible to move them on another application which is supported through QT to e.g. copy the images. `digiKam` can use the IDs, if `ItemDrag` is dropped on `digiKam` itself. The urls set via `setUrls()` are used for external drops (k3b, gimp, ...)

## 6.415 Digikam::DItemInfo Class Reference

### Public Member Functions

- int **albumId** () const
- double **altitude** () const
- QString **aperture** () const
- [CaptionsMap](#) **captions** () const
- int **colorLabel** () const
- QString **comment** () const
- [MetaEngine::AltLangMap](#) **copyrightNotices** () const
- [MetaEngine::AltLangMap](#) **copyrights** () const
- QStringList **creators** () const
- QString **credit** () const
- QDateTime **dateTime** () const
- QSize **dimensions** () const
- **DItemInfo** (const [DInfoInterface::DInfoMap](#) &)
- QString **exposureTime** () const
- qlonglong **fileSize** () const
- QString **focalLength** () const
- QString **focalLength35mm** () const
- bool **hasGeolocationInfo** () const
- [DInfoInterface::DInfoMap](#) **infoMap** () const
- QStringList **keywords** () const
- double **latitude** () const
- QString **lens** () const
- double **longitude** () const
- QString **make** () const
- QString **model** () const
- QString **name** () const
- int **orientation** () const
- int **pickLabel** () const
- int **rating** () const
- QString **rights** () const
- QString **sensitivity** () const
- void **setCaptions** (const [CaptionsMap](#) &)
- void **setColorLabel** (int)
- void **setCopyrightNotices** (const [MetaEngine::AltLangMap](#) &map)
- void **setCopyrights** (const [MetaEngine::AltLangMap](#) &map)
- void **setOrientation** (int)
- void **setPickLabel** (int)
- void **setRating** (int)
- void **setTitles** (const [CaptionsMap](#) &)
- QString **source** () const
- QStringList **tagsPath** () const
- QString **title** () const
- [CaptionsMap](#) **titles** () const
- QString **videoCodec** () const

### 6.415.1 Detailed Description

[DItemInfo](#) is a class to get item information from host application (Showfoto or digiKam) The interface is re-implemented in host and depend how item information must be retrieved (from a database or by file metadata). The easy way to use this container is given below:

```
// READ INFO FROM HOST -----
```

```
QUrl itemUrl; // The item url that you want to retrieve information. DInfoInterface* hostface; // The host application interface instance.
```

```
DInfoInterface::DInfoMap info = hostface->itemInfo(itemUrl); // First stage is to get the information map from host application. DItemInfo item(info); // Second stage, is to create the DItemInfo instance for this item by url. QString title = item.name(); // Now you can retrieve the title, QString description = item.comment(); // The comment, QDateTime time = item.dateTime(); // The time stamp, etc.
```

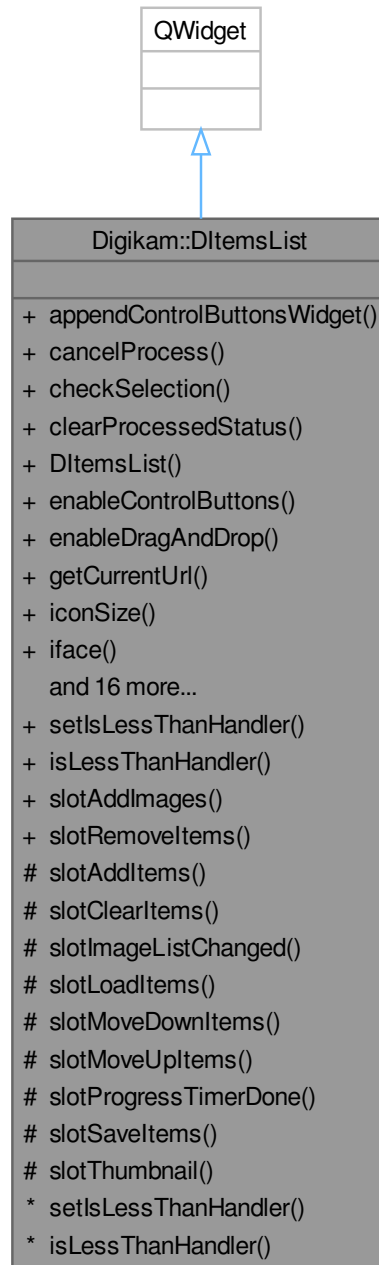
```
// WRITE INFO TO HOST -----
```

```
QUrl itemUrl; // The item url that you want to retrieve information. DInfoInterface* hostface; // The host application interface instance.
```

```
DItemInfo item; // Create the DItemInfo instance for this item with an empty internal info map. item.setRating(3); // Store rating to internal info map. item.setColorLabel(1); // Store color label to internal info map. hostface->setItemInfo(url, item.infoMap()); // Update item information to host using internal info map.
```

## 6.416 Digikam::DItemsList Class Reference

Inheritance diagram for Digikam::DItemsList:



### Public Types

- enum `ControlButton` {  
**Add** = 0x1 , **Remove** = 0x2 , **MoveUp** = 0x4 , **MoveDown** = 0x8 ,  
**Clear** = 0x10 , **Load** = 0x20 , **Save** = 0x40 }

- enum **ControlButtonPlacement** {  
**NoControlButtons** = 0 , **ControlButtonsLeft** , **ControlButtonsRight** , **ControlButtonsAbove** ,  
**ControlButtonsBelow** }

### Public Slots

- virtual void **slotAddImages** (const QList< QUrl > &list)
- virtual void **slotRemoveItems** ()

### Signals

- void **signalAddItems** (const QList< QUrl > &)
- void **signalContextMenuRequested** ()
- void **signalFoundRAWImages** (bool)
- void **signalImageListChanged** ()
- void **signalItemClicked** (QTreeWidgetItem \*)
- void **signalMoveDownItem** ()
- void **signalMoveUpItem** ()
- void **signalRemovedItems** (const QList< int > &)
- void **signalXMLCustomElements** (QXmlStreamReader &)
- void **signalXMLCustomElements** (QXmlStreamWriter &)
- void **signalXMLLoadImageElement** (QXmlStreamReader &)
- void **signalXMLSaveItem** (QXmlStreamWriter &, int)

### Public Member Functions

- void [appendControlButtonsWidget](#) (QWidget \*const widget)
  - void **cancelProcess** ()
  - bool [checkSelection](#) ()
  - void **clearProcessedStatus** ()
  - **DItemsList** (QWidget \*const parent)
  - void **enableControlButtons** (bool enable=true)
  - void **enableDragAndDrop** (const bool enable=true)
  - QUrl **getCurrentUrl** () const
  - int **iconSize** () const
  - [DInfoInterface](#) \* **iface** () const
  - virtual QList< QUrl > **imageUrls** (bool onlyUnprocessed=false) const
  - [DItemsListView](#) \* **listView** () const
  - void [loadImagesFromCurrentAlbum](#) ()
  - void **loadImagesFromCurrentSelection** ()
  - void **processed** (const QUrl &url, bool success)
  - void **processing** (const QUrl &url)
  - virtual void **removeItemByUrl** (const QUrl &url)
  - void **setAllowDuplicate** (bool allow)
  - void **setAllowRAW** (bool allow)
  - void **setControlButtons** (ControlButtons buttonMask)
  - QBoxLayout \* [setControlButtonsPlacement](#) (ControlButtonPlacement placement)
  - void **setCurrentUrl** (const QUrl &url)
  - void **setIconSize** (int size)
  - void **setIface** ([DInfoInterface](#) \*const iface)
  - void **updateThumbnail** (const QUrl &url)
- 
- void [setIsLessThanHandler](#) ([DItemsListIsLessThanHandler](#) fncptr)
  - [DItemsListIsLessThanHandler](#) **isLessThanHandler** () const

## Protected Slots

- virtual void **slotAddItems** ()
- virtual void **slotClearItems** ()
- virtual void **slotImageListChanged** ()
- virtual void **slotLoadItems** ()
- virtual void **slotMoveDownItems** ()
- virtual void **slotMoveUpItems** ()
- void **slotProgressTimerDone** ()
- virtual void **slotSaveItems** ()
- virtual void **slotThumbnail** (const [LoadingDescription](#) &, const QPixmap &)

## 6.416.1 Member Function Documentation

### 6.416.1.1 `appendControlButtonsWidget()`

```
void Digikam::DItemsList::appendControlButtonsWidget (
    QWidget *const widget )
```

Append a extra widget to the end of Control Button layout (as a progress bar for exemple). This method must be call before [setControlButtonsPlacement\(\)](#). Ownership of the widget is not transferred to the DItemList.

### 6.416.1.2 `checkSelection()`

```
bool Digikam::DItemsList::checkSelection ( )
```

a function to check whether an image has been selected or not.

### 6.416.1.3 `loadImagesFromCurrentAlbum()`

```
void Digikam::DItemsList::loadImagesFromCurrentAlbum ( )
```

A function to load all the images from the album if no image has been selected by user.

### 6.416.1.4 `setControlButtonsPlacement()`

```
QBoxLayout * Digikam::DItemsList::setControlButtonsPlacement (
    ControlButtonPlacement placement )
```

Plug the control buttons near to the list, following 'placement' position. Return the instance of the layout supporting the control buttons, if any. This method must be calls after to use [appendControlButtonsWidget\(\)](#).

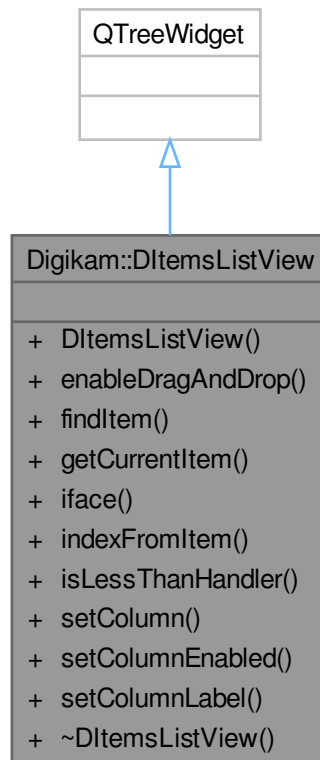
### 6.416.1.5 `setIsLessThanHandler()`

```
void Digikam::DItemsList::setIsLessThanHandler (
    DItemsListIsLessThanHandler fncptr )
```

Methods to handle function pointer used to customize sort items in list. See `DItemsListIsLessThanHandler` type for details.

## 6.417 Digikam::DItemsListView Class Reference

Inheritance diagram for Digikam::DItemsListView:



### Public Types

- enum **ColumnType** {  
**Thumbnail** = 0 , **Filename** , **User1** , **User2** ,  
**User3** , **User4** , **User5** , **User6** }

### Signals

- void **signalAddedDroppedItems** (const QList< QUrl > &)
- void **signalContextMenuRequested** ()
- void **signalItemClicked** (QTreeWidgetItem \*)

### Public Member Functions

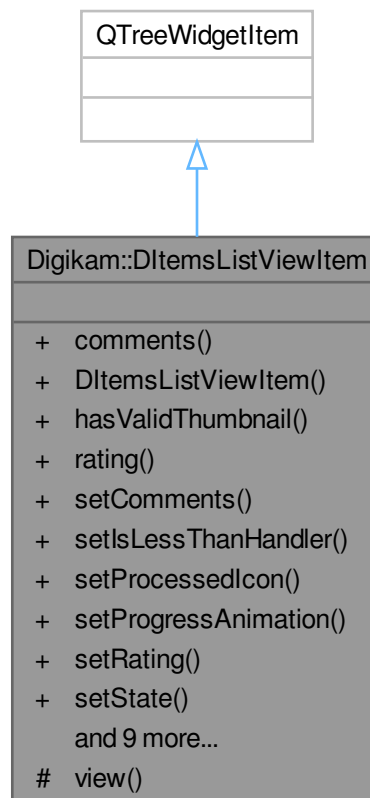
- **DItemsListView** (**DItemsList** \*const parent)
- void **enableDragAndDrop** (const bool enable=true)
- **DItemsListViewItem** \* **findItem** (const QUrl &url)



- [DItemsListViewItem](#) \* **getCurrentItem** () const
- [DInfoInterface](#) \* **iface** () const
- QModelIndex **indexFromItem** ([DItemsListViewItem](#) \*item, int column=0) const
- [DItemsListIsLessThanHandler](#) **isLessThanHandler** () const
- void **setColumn** (ColumnType column, const QString &label, bool enable)
- void **setColumnEnabled** (ColumnType column, bool enable)
- void **setColumnLabel** (ColumnType column, const QString &label)

## 6.418 Digikam::DItemsListViewItem Class Reference

Inheritance diagram for Digikam::DItemsListViewItem:



### Public Types

- enum **State** { **Waiting** , **Success** , **Failed** }

## Public Member Functions

- QString **comments** () const
- DItemsListViewItem (DItemsListView \*const view, const QUrl &url)
- bool **hasValidThumbnail** () const
- int **rating** () const
- void **setComments** (const QString &comments)
- void **setIsLessThanHandler** (DItemsListIsLessThanHandler fncptr)
- void **setProcessedIcon** (const QIcon &icon)
- void **setProgressAnimation** (const QPixmap &pix)
- void **setRating** (int rating)
- void **setState** (State state)
- void **setTags** (const QStringList &tags)
- void **setThumb** (const QPixmap &pix, bool hasThumb=true)
- void **setUrl** (const QUrl &url)
- State **state** () const
- QStringList **tags** () const
- void **updateInformation** ()
- virtual void **updateItemWidgets** ()
- QUrl **url** () const

## Protected Member Functions

- DItemsListView \* **view** () const

## 6.418.1 Member Function Documentation

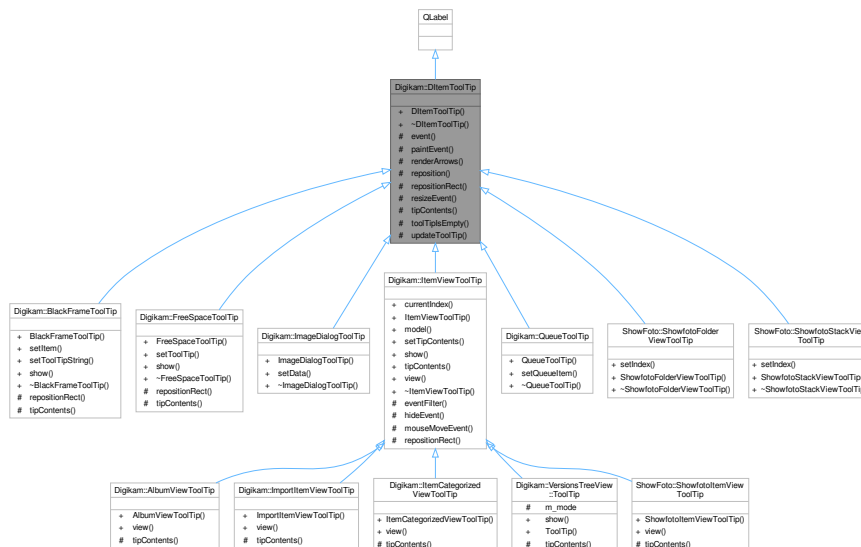
### 6.418.1.1 updateItemWidgets()

```
virtual void Digikam::DItemsListViewItem::updateItemWidgets ( ) [inline], [virtual]
```

Implement this, if you have special item widgets, e.g. an edit line they will be set automatically when adding items, changing order, etc.

## 6.419 Digikam::DItemToolTip Class Reference

Inheritance diagram for Digikam::DItemToolTip:



## Public Member Functions

- **DItemToolTip** (QWidget \*const parent=nullptr)

## Protected Member Functions

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- virtual QRect **repositionRect** ()=0
- void **resizeEvent** (QResizeEvent \*) override
- virtual QString **tipContents** ()=0
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()

## 6.419.1 Member Function Documentation

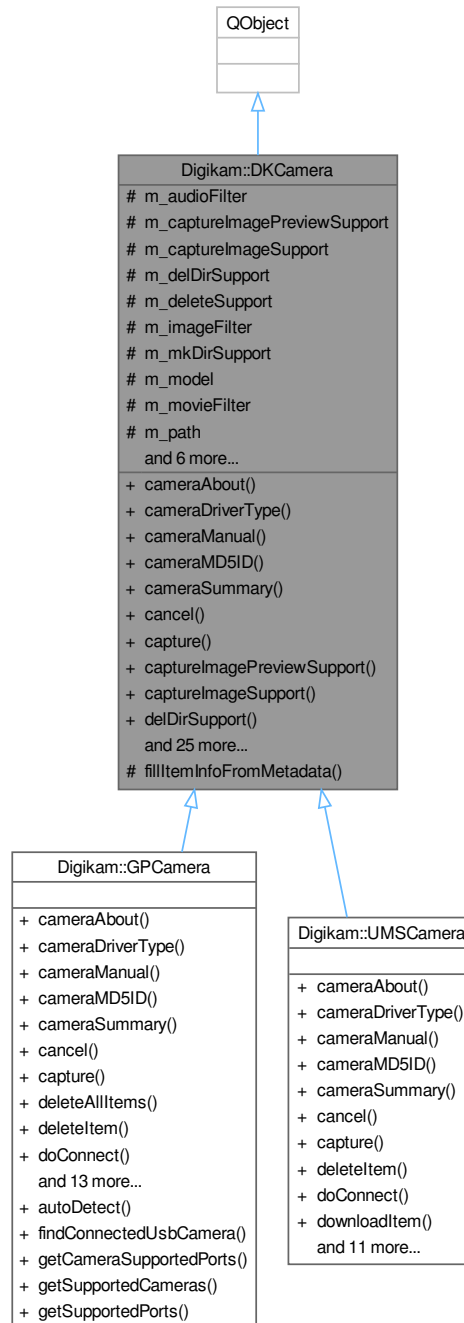
### 6.419.1.1 tipContents()

```
virtual QString Digikam::DItemToolTip::tipContents ( ) [protected], [pure virtual]
```

Implemented in [Digikam::ItemViewToolTip](#).

## 6.420 Digikam::DKCamera Class Reference

Inheritance diagram for Digikam::DKCamera:



### Public Types

- enum **CameraDriverType** { **GPhotoDriver** = 0 , **UMSDriver** }

## Signals

- void **signalFolderList** (const QStringList &)

## Public Member Functions

- virtual bool **cameraAbout** (QString &about)=0
- virtual DKCamera::CameraDriverType **cameraDriverType** ()=0
- virtual bool **cameraManual** (QString &manual)=0
- virtual QByteArray **cameraMD5ID** ()=0
- virtual bool **cameraSummary** (QString &summary)=0
- virtual void **cancel** ()=0
- virtual bool **capture** (CamItemInfo &itemInfo)=0
- bool **captureImagePreviewSupport** () const
- bool **captureImageSupport** () const
- bool **delDirSupport** () const
- virtual bool **deleteltem** (const QString &folder, const QString &itemName)=0
- bool **deleteSupport** () const
- **DKCamera** (const QString &title, const QString &model, const QString &port, const QString &path)
- virtual bool **doConnect** ()=0
- virtual bool **downloadItem** (const QString &folder, const QString &itemName, const QString &saveFile)=0
- virtual bool **getFolders** (const QString &folder)=0
- virtual bool **getFreeSpace** (qint64 &bytesSize, qint64 &bytesAvail)=0
- virtual void **getItemInfo** (const QString &folder, const QString &itemName, CamItemInfo &info, bool use← Metadata)=0
- virtual bool **getItemsInfoList** (const QString &folder, bool useMetadata, CamItemInfoList &infoList)=0
  - If getImageDimensions is false, the camera shall set width and height to -1 if the values are not immediately available.*
- virtual bool **getMetadata** (const QString &folder, const QString &itemName, DMetadata &meta)=0
- virtual bool **getPreview** (QImage &preview)=0
- virtual bool **getThumbnail** (const QString &folder, const QString &itemName, QImage &thumbnail)=0
- QString **mimeType** (const QString &fileext) const
- bool **mkDirSupport** () const
- QString **model** () const
- QString **path** () const
- QString **port** () const
- void **printSupportedFeatures** ()
- virtual bool **setLockItem** (const QString &folder, const QString &itemName, bool lock)=0
- bool **thumbnailSupport** () const
- QString **title** () const
- virtual bool **uploadItem** (const QString &folder, const QString &itemName, const QString &localFile, CamItemInfo &itemInfo)=0
- bool **uploadSupport** () const
- QString **uuid** () const

## Protected Member Functions

- void **fillItemInfoFromMetadata** (CamItemInfo &item, const DMetadata &meta) const

## Protected Attributes

- QString `m_audioFilter`
- bool `m_captureImagePreviewSupport` = false
- bool `m_captureImageSupport` = false
- bool `m_delDirSupport` = false
- bool `m_deleteSupport` = false
- QString `m_imageFilter`
- bool `m_mkdirSupport` = false
- QString `m_model`
- QString `m_movieFilter`
- QString `m_path`
- QString `m_port`
- QString `m_rawFilter`
- bool `m_thumbnailSupport` = false
- QString `m_title`
- bool `m_uploadSupport` = false
- QString `m_uuid`

## 6.420.1 Member Function Documentation

### 6.420.1.1 capture()

```
virtual bool Digikam::DKCamera::capture (
    CamItemInfo & itemInfo ) [pure virtual]
```

Implemented in [Digikam::UMSCamera](#).

### 6.420.1.2 getFreeSpace()

```
virtual bool Digikam::DKCamera::getFreeSpace (
    qint64 & bytesSize,
    qint64 & bytesAvail ) [pure virtual]
```

Implemented in [Digikam::UMSCamera](#).

### 6.420.1.3 getItemsInfoList()

```
virtual bool Digikam::DKCamera::getItemsInfoList (
    const QString & folder,
    bool useMetadata,
    CamItemInfoList & infoList ) [pure virtual]
```

Implemented in [Digikam::UMSCamera](#), and [Digikam::GPCamera](#).

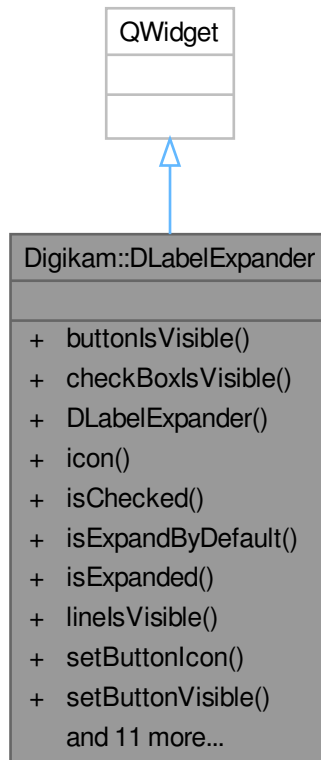
### 6.420.1.4 getPreview()

```
virtual bool Digikam::DKCamera::getPreview (
    QImage & preview ) [pure virtual]
```

Implemented in [Digikam::UMSCamera](#).

## 6.421 Digikam::DLabelExpander Class Reference

Inheritance diagram for Digikam::DLabelExpander:



### Signals

- void **signalButtonPressed** ()
- void **signalExpanded** (bool)
- void **signalToggled** (bool)

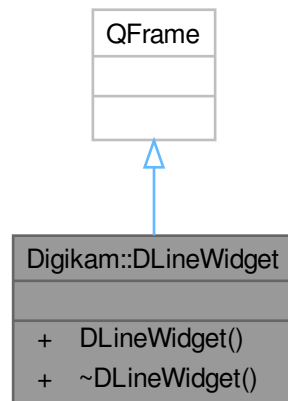
### Public Member Functions

- bool **buttonIsVisible** () const
- bool **checkboxIsVisible** () const
- **DLabelExpander** (QWidget \*const parent=nullptr)
- QIcon **icon** () const
- bool **isChecked** () const
- bool **isExpandByDefault** () const
- bool **isExpanded** () const
- bool **lineIsVisible** () const
- void **setButtonIcon** (const QIcon &icon)
- void **setButtonVisible** (bool b)

- void **setCheckBoxVisible** (bool b)
- void **setChecked** (bool b)
- void **setExpandByDefault** (bool b)
- void **setExpanded** (bool b)
- void **setIcon** (const QIcon &icon)
- void **setLineVisible** (bool b)
- void **setText** (const QString &txt)
- void **setWidget** (QWidget \*const widget)
- QString **text** () const
- QWidget \* **widget** () const

## 6.422 Digikam::DLineWidget Class Reference

Inheritance diagram for Digikam::DLineWidget:



### Public Member Functions

- **DLineWidget** (Qt::Orientation orientation, QWidget \*const parent=nullptr)

### 6.422.1 Detailed Description

A widget to show an horizontal or vertical line separator



## 6.423 Digikam::DLogoAction Class Reference

Inheritance diagram for Digikam::DLogoAction:



### Public Member Functions

- **DLogoAction** (QObject \*const parent, bool alignOnright=true)
- bool **running** () const
- void **start** ()
- void **stop** ()

### Protected Member Functions

- QWidget \* **createWidget** (QWidget \*parent) override
- void **deleteWidget** (QWidget \*widget) override

## 6.424 Digikam::DMessageBox Class Reference

### Static Public Member Functions

- static bool [readMsgBoxShouldBeShown](#) (const QString &dontShowAgainName)
- static void [saveMsgBoxShouldBeShown](#) (const QString &dontShowAgainName, bool value)
- static int [showContinueCancel](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, const QString &dontAskAgainName=QString())

- static int [showContinueCancelList](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, const QStringList &items, const QString &dontAskAgainName=QString())
- static int [showContinueCancelWidget](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, QWidget \*const listWidget, const QString &dontAskAgainName)
- static void [showInformationList](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, const QStringList &items, const QString &dontShowAgainName=QString())
- static void [showInformationWidget](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, QWidget \*const listWidget, const QString &dontShowAgainName)
- static int [showYesNo](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, const QString &dontAskAgainName=QString())
- static int [showYesNoList](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, const QStringList &items, const QString &dontAskAgainName=QString())
- static int [showYesNoWidget](#) (QMessageBox::Icon icon, QWidget \*const parent, const QString &caption, const QString &text, QWidget \*const listWidget, const QString &dontAskAgainName=QString())

## 6.424.1 Member Function Documentation

### 6.424.1.1 readMsgBoxShouldBeShown()

```
bool Digikam::DMessageBox::readMsgBoxShouldBeShown (
    const QString & dontShowAgainName ) [static]
```

#### Returns

true if the corresponding message box should be shown.

#### Parameters

<i>dontShowAgainName</i>	the name that identify the message box. If empty, this method return false.
--------------------------	---

### 6.424.1.2 saveMsgBoxShouldBeShown()

```
void Digikam::DMessageBox::saveMsgBoxShouldBeShown (
    const QString & dontShowAgainName,
    bool value ) [static]
```

Save the fact that the message box should not be shown again.

#### Parameters

<i>dontShowAgainName</i>	the name that identify the message box. If empty, this method does nothing.
<i>value</i>	the value chosen in the message box to show it again next time.

### 6.424.1.3 showContinueCancel()

```
int Digikam::DMessageBox::showContinueCancel (
    QMessageBox::Icon icon,
```

```
QWidget *const parent,  
const QString & caption,  
const QString & text,  
const QString & dontAskAgainName = QString() ) [static]
```

Show a message box with Continue and Cancel buttons, and wait user feedback. Return QMessageBox::Yes or QMessageBox::Cancel.

#### 6.424.1.4 showContinueCancelList()

```
int Digikam::DMessageBox::showContinueCancelList (  
    QMessageBox::Icon icon,  
    QWidget *const parent,  
    const QString & caption,  
    const QString & text,  
    const QStringList & items,  
    const QString & dontAskAgainName = QString() ) [static]
```

Show List of items to process into a message box with Continue and Cancel buttons, and wait user feedback. Return QMessageBox::Yes or QMessageBox::Cancel.

#### 6.424.1.5 showContinueCancelWidget()

```
int Digikam::DMessageBox::showContinueCancelWidget (  
    QMessageBox::Icon icon,  
    QWidget *const parent,  
    const QString & caption,  
    const QString & text,  
    QWidget *const listWidget,  
    const QString & dontAskAgainName ) [static]
```

Show widget into a message box with Continue and Cancel buttons, and wait user feedback. Return QMessageBox::Yes or QMessageBox::Cancel.

#### 6.424.1.6 showInformationList()

```
void Digikam::DMessageBox::showInformationList (  
    QMessageBox::Icon icon,  
    QWidget *const parent,  
    const QString & caption,  
    const QString & text,  
    const QStringList & items,  
    const QString & dontShowAgainName = QString() ) [static]
```

Show List of items into an informative message box.

#### 6.424.1.7 showInformationWidget()

```
void Digikam::DMessageBox::showInformationWidget (  
    QMessageBox::Icon icon,  
    QWidget *const parent,  
    const QString & caption,  
    const QString & text,  
    QWidget *const listWidget,  
    const QString & dontShowAgainName ) [static]
```

Show widget into an informative message box.

#### 6.424.1.8 showYesNo()

```
int Digikam::DMessageBox::showYesNo (
    QMessageBox::Icon icon,
    QWidget *const parent,
    const QString & caption,
    const QString & text,
    const QString & dontAskAgainName = QString() ) [static]
```

Show a message box with Yes and No buttons, and wait user feedback. Return QMessageBox::Yes or QMessageBox::No.

#### 6.424.1.9 showYesNoList()

```
int Digikam::DMessageBox::showYesNoList (
    QMessageBox::Icon icon,
    QWidget *const parent,
    const QString & caption,
    const QString & text,
    const QStringList & items,
    const QString & dontAskAgainName = QString() ) [static]
```

Show List of items to process into a message box with Yes and No buttons, and wait user feedback. Return QMessageBox::Yes or QMessageBox::No.

#### 6.424.1.10 showYesNoWidget()

```
int Digikam::DMessageBox::showYesNoWidget (
    QMessageBox::Icon icon,
    QWidget *const parent,
    const QString & caption,
    const QString & text,
    QWidget *const listWidget,
    const QString & dontAskAgainName = QString() ) [static]
```

Show widget into a message box with Yes and No buttons, and wait user feedback. Return QMessageBox::Yes or QMessageBox::No.



- enum [VIDEOCOLORMODEL](#) {  
**VIDEOCOLORMODEL\_UNKNOWN** = 1000 , **VIDEOCOLORMODEL\_OTHER** , **VIDEOCOLORMODEL\_SRGB** , **VIDEOCOLORMODEL\_BT709** ,  
**VIDEOCOLORMODEL\_BT601** }

## Public Types inherited from [Digikam::MetaEngine](#)

- typedef QMap< QString, QString > [AltLangMap](#)
- enum [Backend](#) {  
**Exiv2Backend** = 0 , **LibRawBackend** , **LibHeifBackend** , **ImageMagickBackend** ,  
**FFmpegBackend** , **ExifToolBackend** , **VideoMergeBackend** , **NoBackend** }
- enum [ImageColorWorkSpace](#) { **WORKSPACE\_UNSPECIFIED** = 0 , **WORKSPACE\_SRGB** = 1 ,  
**WORKSPACE\_ADOBERGB** = 2 , **WORKSPACE\_UNCALIBRATED** = 65535 }
- enum [ImageOrientation](#) {  
**ORIENTATION\_UNSPECIFIED** = 0 , **ORIENTATION\_NORMAL** = 1 , **ORIENTATION\_HFLIP** = 2 ,  
**ORIENTATION\_ROT\_180** = 3 ,  
**ORIENTATION\_VFLIP** = 4 , **ORIENTATION\_ROT\_90\_HFLIP** = 5 , **ORIENTATION\_ROT\_90** = 6 ,  
**ORIENTATION\_ROT\_90\_VFLIP** = 7 ,  
**ORIENTATION\_ROT\_270** = 8 }
- typedef QMap< QString, QString > [MetadataMap](#)
- enum [MetadataWritingMode](#) { **WRITE\_TO\_FILE\_ONLY** = 0 , **WRITE\_TO\_SIDECAR\_ONLY** = 1 ,  
**WRITE\_TO\_SIDECAR\_AND\_FILE** = 2 , **WRITE\_TO\_SIDECAR\_ONLY\_FOR\_READ\_ONLY\_FILES** = 3  
}
- typedef QMap< QString, QStringList > [TagsMap](#)
- enum [XmpTagType](#) {  
**NormalTag** = 0 , **ArrayBagTag** = 1 , **StructureTag** = 2 , **ArrayLangTag** = 3 ,  
**ArraySeqTag** = 4 }

## Public Member Functions

- bool [addToXmpTagStringBag](#) (const char \*const xmpTagName, const QStringList &entriesToAdd) const
- bool [applyChanges](#) (bool setVersion=false) const
- **DMetadata** (const [MetaEngineData](#) &data)
- **DMetadata** (const QString &filePath)
- bool [getACDSeeTagsPath](#) (QStringList &tagsPath) const
- QString [getCameraSerialNumber](#) () const
- bool [getCopyrightInformation](#) ([Template](#) &t) const
- [IptcCoreContactInfo](#) [getCreatorContactInfo](#) () const
- [IccProfile](#) [getIccProfile](#) () const
- [IptcCoreLocationInfo](#) [getIptcCoreLocation](#) () const
- QStringList [getIptcCoreSubjects](#) () const
- int [getItemColorLabel](#) (const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance](#)() ->settings()) const
- [CaptionsMap](#) [getItemComments](#) (const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance](#)() ->settings()) const
- bool [getItemFacesMap](#) (QMultiMap< QString, QVariant > &facesPath) const
- QString [getItemHistory](#) () const
- int [getItemPickLabel](#) (const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance](#)() ->settings()) const
- int [getItemRating](#) (const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance](#)() ->settings()) const
- bool [getItemTagsPath](#) (QStringList &tagsPath, const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance](#)() ->settings()) const

- [CaptionsMap](#) **getItemTitles** (const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance\(\)](#) ->settings()) const
- [QString](#) **getItemUniqueId** () const
- [QString](#) **getLensDescription** () const
- [QVariant](#) **getMetadataField** ([MetadataInfo::Field](#) field) const
- [QVariantList](#) **getMetadataFields** (const [MetadataFields](#) &fields) const
- [Template](#) **getMetadataTemplate** () const
- [int](#) **getMsecsInfo** () const
- [PhotoInfoContainer](#) **getPhotographInformation** () const
- [VideoInfoContainer](#) **getVideoInformation** () const
- [QStringList](#) **getXmpKeywords** () const
- [QStringList](#) **getXmpSubCategories** () const
- [QStringList](#) **getXmpSubjects** () const
- [bool](#) **hasItemHistoryTag** () const
- [bool](#) **load** (const [QString](#) &filePath, [bool](#) videoAll=false, [Backend](#) \*backend=nullptr)
- [bool](#) **loadUsingFFmpeg** (const [QString](#) &filePath)
- [bool](#) **loadUsingRawEngine** (const [QString](#) &filePath)
- [bool](#) **mSecTimeStamp** (const [char](#) \*const exifTagName, [int](#) &ms) const
- [void](#) **registerMetadataSettings** ()
- [bool](#) **removeExifColorSpace** () const
- [bool](#) **removeExifTags** (const [QStringList](#) &tagFilters)
- [bool](#) **removeFromXmpTagStringBag** (const [char](#) \*const xmpTagName, const [QStringList](#) &entriesToRemove) const
- [bool](#) **removelptcTags** (const [QStringList](#) &tagFilters)
- [bool](#) **removeItemFacesMap** () const
- [bool](#) **removeMetadataTemplate** () const
- [bool](#) **removeXmpKeywords** (const [QStringList](#) &keywordsToRemove)
- [bool](#) **removeXmpSubCategories** (const [QStringList](#) &categoriesToRemove)
- [bool](#) **removeXmpSubjects** (const [QStringList](#) &subjectsToRemove)
- [bool](#) **removeXmpTags** (const [QStringList](#) &tagFilters)
- [bool](#) **save** (const [QString](#) &filePath, [bool](#) setVersion=false) const
- [bool](#) **setACDSeeTagsPath** (const [QStringList](#) &tagsPath) const
- [bool](#) **setCreatorContactInfo** (const [IptcCoreContactInfo](#) &info) const
- [bool](#) **setIccProfile** (const [IccProfile](#) &profile)
- [bool](#) **setIptcCoreLocation** (const [IptcCoreLocationInfo](#) &location) const
- [bool](#) **setItemColorLabel** ([int](#) colorId, const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance\(\)](#) ->settings()) const
- [bool](#) **setItemComments** (const [CaptionsMap](#) &comments, const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance\(\)](#) ->settings()) const
- [bool](#) **setItemFacesMap** (const [QMultiMap](#)< [QString](#), [QVariant](#) > &facesPath, [bool](#) write, const [QSize](#) &size=[QSize\(\)](#)) const
- [bool](#) **setItemHistory** (const [QString](#) &imageHistoryXml) const
- [bool](#) **setItemPickLabel** ([int](#) pickId, const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance\(\)](#) ->settings()) const
- [bool](#) **setItemRating** ([int](#) rating, const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance\(\)](#) ->settings()) const
- [bool](#) **setItemTagsPath** (const [QStringList](#) &tagsPath, const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance\(\)](#) ->settings()) const
- [bool](#) **setItemTitles** (const [CaptionsMap](#) &title, const [DMetadataSettingsContainer](#) &settings=[DMetadataSettings::instance\(\)](#) ->settings()) const
- [bool](#) **setItemUniqueId** (const [QString](#) &uuid) const
- [bool](#) **setMetadataTemplate** (const [Template](#) &t) const
- [void](#) **setSettings** (const [MetaEngineSettingsContainer](#) &settings)
- [bool](#) **setXmpKeywords** (const [QStringList](#) &newKeywords) const
- [bool](#) **setXmpSubCategories** (const [QStringList](#) &newSubCategories) const
- [bool](#) **setXmpSubjects** (const [QStringList](#) &newSubjects) const

## Public Member Functions inherited from [Digikam::MetaEngine](#)

- [MetaEngine](#) ()
  - [MetaEngine](#) (const [MetaEngineData](#) &data)
  - [MetaEngine](#) (const QString &filePath)
  - virtual [~MetaEngine](#) ()
- 
- [MetaEngineData](#) **data** () const
  - void **setData** (const [MetaEngineData](#) &data)
  - bool **loadFromData** (const QByteArray &imgData)
  - bool **loadFromDataAndMerge** (const QByteArray &imgData, const QStringList &exclude=QStringList())
  - bool **isEmpty** () const
  - QSize **getPixelSize** () const
  - QString **getMimeType** () const
  - void **setReadWithExifTool** (const bool on)
  - bool **readWithExifTool** () const
  - void **setWriteWithExifTool** (const bool on)
  - bool **writeWithExifTool** () const
  - void **setWriteRawFiles** (const bool on)
  - bool **writeRawFiles** () const
  - void **setWriteDngFiles** (const bool on)
  - bool **writeDngFiles** () const
  - void **setUseXMPSidecar4Reading** (const bool on)
  - bool **useXMPSidecar4Reading** () const
  - void **setUseCompatibleFileName** (const bool on)
  - bool **useCompatibleFileName** () const
  - void **setMetadataWritingMode** (const int mode)
  - int **metadataWritingMode** () const
  - void **setUpdateFileTimeStamp** (bool on)
  - bool **updateFileTimeStamp** () const
- 
- bool **setItemProgramId** (const QString &program, const QString &version) const
  - QSize **getItemDimensions** () const
  - bool **setItemDimensions** (const QSize &size) const
  - [MetaEngine::ImageOrientation](#) **getItemOrientation** () const
  - bool **setItemOrientation** ([ImageOrientation](#) orientation) const
  - [MetaEngine::ImageColorWorkSpace](#) **getItemColorWorkSpace** () const
  - bool **setItemColorWorkSpace** ([ImageColorWorkSpace](#) workspace) const
  - QDateTime **getItemDateTime** () const
  - bool **setItemDateTime** (const QDateTime &dateTime, bool setDateDigitized=false) const
  - QDateTime **getDigitizationDateTime** (bool fallbackToCreationTime=false) const
  - bool **getItemPreview** (QImage &preview) const
  - bool **setItemPreview** (const QImage &preview) const
  - QByteArray **getItemIccProfile** () const
  - bool **setItemIccProfile** (const QByteArray &iccData) const
- 
- bool **initializeGPSInfo** ()
- 
- bool **getGPSInfo** (double &altitude, double &latitude, double &longitude) const
- 
- QString **getGPSLatitudeString** () const



- QString **getGPSLongitudeString** () const
- bool **getGPSLatitudeNumber** (double \*const latitude) const
- bool **getGPSLongitudeNumber** (double \*const longitude) const
- bool **getGPSAltitude** (double \*const altitude) const
- bool **setGPSInfo** (const double altitude, const double latitude, const double longitude)
- bool **setGPSInfo** (const double \*const altitude, const double latitude, const double longitude)
- bool **setGPSInfo** (const double altitude, const QString &latitude, const QString &longitude)
- bool **removeGPSInfo** ()
- void **setFilePath** (const QString &path)
- QString **getFilePath** () const
- bool **load** (const QString &filePath, **Backend** \*backend=nullptr)
- bool **loadFromSidecarAndMerge** (const QString &filePath)
- bool **save** (const QString &filePath, bool setVersion=false) const
- bool **applyChanges** (bool setVersion=false) const
- bool **exportChanges** (const QString &exvTmpFile) const
- bool **hasComments** () const
- bool **clearComments** () const
- QByteArray **getComments** () const
- QString **getCommentsDecoded** () const
- bool **setComments** (const QByteArray &data) const
- **TagsMap** **getStdExifTagsList** () const
- **TagsMap** **getMakernoteTagsList** () const

- bool [hasExif](#) () const
- bool [clearExif](#) () const
- QByteArray [getExifEncoded](#) (bool addExifHeader=false) const
- bool [setExif](#) (const QByteArray &data) const
- QImage [getExifThumbnail](#) (bool fixOrientation) const
- bool [rotateExifQImage](#) (QImage &image, [ImageOrientation](#) orientation) const
- bool [setExifThumbnail](#) (const QImage &thumb) const
- bool [removeExifThumbnail](#) () const
- bool [setTiffThumbnail](#) (const QImage &thumb) const
- QString [getExifComment](#) (bool readDescription=true) const
- QString [getExifTagComment](#) (const char \*exifTagName) const
- bool [setExifComment](#) (const QString &comment, bool writeDescription=true) const
- QString [getExifTagString](#) (const char \*exifTagName, bool escapeCR=true) const
- bool [setExifTagString](#) (const char \*exifTagName, const QString &value) const
- bool [getExifTagLong](#) (const char \*exifTagName, long &val) const
- bool [getExifTagLong](#) (const char \*exifTagName, long &val, int component) const
- bool [setExifTagLong](#) (const char \*exifTagName, long val) const
- bool [setExifTagUShort](#) (const char \*exifTagName, unsigned int val) const
- bool [getExifTagRational](#) (const char \*exifTagName, long int &num, long int &den, int component=0) const
- bool [setExifTagRational](#) (const char \*exifTagName, long int num, long int den) const
- bool [setExifTagURational](#) (const char \*exifTagName, unsigned long int num, unsigned long int den) const
- QByteArray [getExifTagData](#) (const char \*exifTagName) const

- bool [setExifTagData](#) (const char \*exifTagName, const QByteArray &data) const
- QVariant [getExifTagVariant](#) (const char \*exifTagName, bool rationalAsListOfInts=true, bool escapeCR=true, int component=0) const
- bool [setExifTagVariant](#) (const char \*exifTagName, const QVariant &data, bool rationalWantSmall↔Denominator=true) const
- bool [removeExifTag](#) (const char \*exifTagName) const
- QString [getExifTagTitle](#) (const char \*exifTagName)
- QString [getExifTagDescription](#) (const char \*exifTagName)
- QString [createExifUserStringFromValue](#) (const char \*exifTagName, const QVariant &val, bool escape↔CR=true)
- [MetaEngine::MetaDataMap](#) [getExifTagsDataList](#) (const QStringList &exifKeysFilter=QStringList(), bool invertSelection=false, bool extractBinary=true) const
- [MetaEngine::TagsMap](#) [getIptcTagsList](#) () const
- bool [hasIptc](#) () const
- bool [clearIptc](#) () const
- QByteArray [getIptc](#) (bool addIrbHeader=false) const
- bool [setIptc](#) (const QByteArray &data) const
- QString [getIptcTagString](#) (const char \*iptcTagName, bool escapeCR=true) const
- bool [setIptcTagString](#) (const char \*iptcTagName, const QString &value) const
- QStringList [getIptcTagsStringList](#) (const char \*iptcTagName, bool escapeCR=true) const
- bool [setIptcTagsStringList](#) (const char \*iptcTagName, int maxSize, const QStringList &oldValues, const QStringList &newValues) const
- QByteArray [getIptcTagData](#) (const char \*iptcTagName) const
- bool [setIptcTagData](#) (const char \*iptcTagName, const QByteArray &data) const
- bool [removeIptcTag](#) (const char \*iptcTagName) const

- QString [getIptcTagTitle](#) (const char \*iptcTagName)
- QString [getIptcTagDescription](#) (const char \*iptcTagName)
- [MetaEngine::MetaDataMap](#) [getIptcTagsDataList](#) (const QStringList &iptcKeysFilter=QStringList(), bool invertSelection=false) const
- QStringList [getIptcKeywords](#) () const
- bool [setIptcKeywords](#) (const QStringList &oldKeywords, const QStringList &newKeywords) const
- QStringList [getIptcSubjects](#) () const
- bool [setIptcSubjects](#) (const QStringList &oldSubjects, const QStringList &newSubjects) const
- QStringList [getIptcSubCategories](#) () const
- bool [setIptcSubCategories](#) (const QStringList &oldSubCategories, const QStringList &newSubCategories) const
- [MetaEngine::TagsMap](#) [getXmpTagsList](#) () const
- bool [hasXmp](#) () const
- bool [clearXmp](#) () const
- QByteArray [getXmp](#) () const
- bool [setXmp](#) (const QByteArray &data) const
- QString [getXmpTagString](#) (const char \*xmpTagName, bool escapeCR=true) const
- bool [setXmpTagString](#) (const char \*xmpTagName, const QString &value) const
- bool [setXmpTagString](#) (const char \*xmpTagName, const QString &value, [XmpTagType](#) type) const
- QString [getXmpTagTitle](#) (const char \*xmpTagName)
- QString [getXmpTagDescription](#) (const char \*xmpTagName)
- [MetaEngine::MetaDataMap](#) [getXmpTagsDataList](#) (const QStringList &xmpKeysFilter=QStringList(), bool invertSelection=false) const

- [MetaEngine::AltLangMap getXmpTagStringListLangAlt](#) (const char \*xmpTagName, bool escapeCR=true) const
- bool [setXmpTagStringListLangAlt](#) (const char \*xmpTagName, const [MetaEngine::AltLangMap](#) &values) const
- QString [getXmpTagStringLangAlt](#) (const char \*xmpTagName, const QString &langAlt, bool escapeCR) const
- bool [setXmpTagStringLangAlt](#) (const char \*xmpTagName, const QString &value, const QString &langAlt) const
- QStringList [getXmpTagStringSeq](#) (const char \*xmpTagName, bool escapeCR=true) const
- bool [setXmpTagStringSeq](#) (const char \*xmpTagName, const QStringList &seq) const
- QStringList [getXmpTagStringBag](#) (const char \*xmpTagName, bool escapeCR) const
- bool [setXmpTagStringBag](#) (const char \*xmpTagName, const QStringList &bag) const
- bool [addToXmpTagStringBag](#) (const char \*xmpTagName, const QStringList &entriesToAdd) const
- bool [removeFromXmpTagStringBag](#) (const char \*xmpTagName, const QStringList &entriesToRemove) const
- QVariant [getXmpTagVariant](#) (const char \*xmpTagName, bool rationalAsListofInts=true, bool stringEscape↔CR=true) const
- QStringList [getXmpKeywords](#) () const
- bool [setXmpKeywords](#) (const QStringList &newKeywords) const
- bool [removeXmpKeywords](#) (const QStringList &keywordsToRemove)
- QStringList [getXmpSubjects](#) () const
- bool [setXmpSubjects](#) (const QStringList &newSubjects) const
- bool [removeXmpSubjects](#) (const QStringList &subjectsToRemove)
- QStringList [getXmpSubCategories](#) () const
- bool [setXmpSubCategories](#) (const QStringList &newSubCategories) const
- bool [removeXmpSubCategories](#) (const QStringList &categoriesToRemove)
- bool [removeXmpTag](#) (const char \*xmpTagName, bool family=false) const

### Static Public Member Functions

- static double **apexApertureToFNumber** (double aperture)
- static double **apexShutterSpeedToExposureTime** (double shutterSpeed)
- static CountryCodeMap **countryCodeMap** ()
- static CountryCodeMap **countryCodeMap2** ()
- static QMap< int, QString > **possibleValuesForEnumField** (MetadataInfo::Field field)
- static **MetaEngine::AltLangMap toAltLangMap** (const QVariant &var)
- static QStringList **valuesToString** (const QVariantList &list, const MetadataFields &fields)
- static QString **valueToString** (const QVariant &value, MetadataInfo::Field field)
- static QString **videoColorModelToString** (VIDEOCOLORMODEL videoColorModel)

### Static Public Member Functions inherited from **Digikam::MetaEngine**

- static bool **initializeExiv2** ()
  - static bool **supportXmp** ()
  - static bool **supportJpegXL** ()
  - static bool **supportBmff** ()
  - static bool **supportMetadataWriting** (const QString &typeMime)
  - static QString **Exiv2Version** ()
- 
- static void **convertToRational** (const double number, long int \*const numerator, long int \*const denominator, const int rounding)
- 
- static void **convertToRationalSmallDenominator** (const double number, long int \*const numerator, long int \*const denominator)
- 
- static double **convertDegreeAngleToDouble** (double degrees, double minutes, double seconds)
- 
- static QString **convertToGPSCoordinateString** (const long int numeratorDegrees, const long int denominatorDegrees, const long int numeratorMinutes, const long int denominatorMinutes, const long int numeratorSeconds, const long int denominatorSeconds, const char directionReference)
- 
- static QString **convertToGPSCoordinateString** (const bool isLatitude, double coordinate)
- 
- static bool **convertFromGPSCoordinateString** (const QString &coordinate, long int \*const numeratorDegrees, long int \*const denominatorDegrees, long int \*const numeratorMinutes, long int \*const denominatorMinutes, long int \*const numeratorSeconds, long int \*const denominatorSeconds, char \*const directionReference)
- 
- static bool **convertFromGPSCoordinateString** (const QString &gpsString, double \*const coordinate)
- 
- static bool **convertToUserPresentableNumbers** (const QString &coordinate, int \*const degrees, int \*const minutes, double \*const seconds, char \*const directionReference)
- 
- static void **convertToUserPresentableNumbers** (const bool isLatitude, double coordinate, int \*const degrees, int \*const minutes, double \*const seconds, char \*const directionReference)
- 
- static QString **sidecarFilePathForFile** (const QString &path)

- static QString [sidecarPath](#) (const QString &path)
- static QUrl [sidecarUrl](#) (const QUrl &url)
- static QUrl [sidecarUrl](#) (const QString &path)
- static bool [hasSidecar](#) (const QString &path)
- static QString [backendName](#) ([Backend](#) t)
- static bool [canWriteComment](#) (const QString &filePath)
- static QString [detectLanguageAlt](#) (const QString &value, QString &lang)
- static bool [canWriteExif](#) (const QString &filePath)
- static bool [canWriteIptc](#) (const QString &filePath)
- static bool [canWriteXmp](#) (const QString &filePath)
- static bool [registerXmpNameSpace](#) (const QString &uri, const QString &prefix)
- static bool [unregisterXmpNameSpace](#) (const QString &uri)

#### Additional Inherited Members

#### Protected Member Functions inherited from [Digikam::MetaEngine](#)

- bool [setProgramId](#) () const

### 6.425.1 Member Enumeration Documentation

#### 6.425.1.1 VIDEOCOLORMODEL

enum [Digikam::DMetadata::VIDEOCOLORMODEL](#)

Video color model reported by FFmpeg following XMP DM Spec from Adobe. These values are stored in DB as Image color model properties (extension of [DImg::ColorModel](#))

## 6.425.2 Member Function Documentation

### 6.425.2.1 addToXmpTagStringBag()

```
bool Digikam::DMetadata::addToXmpTagStringBag (
    const char *const xmpTagName,
    const QStringList & entriesToAdd ) const
```

Set an Xmp tag content using a list of strings defined by the 'entriesToAdd' parameter. The existing entries are preserved. The method will compare all new with all already existing entries to prevent duplicates in the image. Return true if the entries have been added to metadata.

### 6.425.2.2 countryCodeMap()

```
DMetadata::CountryCodeMap Digikam::DMetadata::countryCodeMap ( ) [static]
```

Return a map of ISO-639-1 2 letters country codes with country names.

### 6.425.2.3 countryCodeMap2()

```
DMetadata::CountryCodeMap Digikam::DMetadata::countryCodeMap2 ( ) [static]
```

Return a map of ISO-639-2 3 letters country codes with country names.

### 6.425.2.4 getCameraSerialNumber()

```
QString Digikam::DMetadata::getCameraSerialNumber ( ) const
```

Return a string with Camera serial number.

### 6.425.2.5 getCopyrightInformation()

```
bool Digikam::DMetadata::getCopyrightInformation (
    Template & t ) const
```

Fills only the copyright values in the template. Use getMetadataTemplate() usually. Returns true if valid fields were read.

### 6.425.2.6 getIccProfile()

```
IccProfile Digikam::DMetadata::getIccProfile ( ) const
```

Reads an [IccProfile](#) that is described or embedded in the metadata. This method does not retrieve profiles embedded in the image but from the Exif metadata, e.g. embedded profiles in JPEG images. Returns a null profile if no profile is found.



### 6.425.2.7 getItemFacesMap()

```
bool Digikam::DMetadata::getItemFacesMap (
    QMap<QString, QVariant> & facesPath ) const
```

Get Images Face Map based on tags stored in Picassa/Metadatagroup format. Read face tags only if Exiv2 can write them, otherwise garbage tags will be generated on image transformation

### 6.425.2.8 getLensDescription()

```
QString Digikam::DMetadata::getLensDescription ( ) const
```

Return a string with Lens mounted on the front of camera. There no standard Exif tag for Lens information. Camera makernotes and Xmp tags are parsed. Take a care : lens information are not standardized and string content is not homogeneous between camera model/maker. < Canon Cameras Makernote.

< Canon Cameras Makernote.

< Alternative Canon Cameras Makernote.

< Nikon Cameras Makernote.

< Nikon Cameras Makernote.

< Nikon Cameras Makernote.

< Minolta Cameras Makernote.

< Sony Cameras Makernote.

< Sony Cameras Makernote.

< Sony Cameras Makernote.

< Pentax Cameras Makernote.

< Pentax Cameras Makernote.

< Panasonic Cameras Makernote.

< Panasonic Cameras Makernote.

< Sigma Cameras Makernote.

< Samsung Cameras Makernote.

< Non-standard Exif tag set by Camera Raw.

< Olympus Cameras Makernote.

< Olympus Cameras Makernote.

### 6.425.2.9 getMetadataField()

```
QVariant Digikam::DMetadata::getMetadataField (
    MetadataInfo::Field field ) const
```

Returns the requested metadata field as a QVariant. See `metadatainfo.h` for a specification of the format of the QVariant.

### 6.425.2.10 getMSecsInfo()

```
int Digikam::DMetadata::getMSecsInfo ( ) const
```

Returns millisecond time-stamp from Exif tags or 0 if not found.

### 6.425.2.11 getVideoInformation()

```
VideoInfoContainer Digikam::DMetadata::getVideoInformation ( ) const
```

Returns video metadata from Xmp tags.

### 6.425.2.12 getXmpKeywords()

```
QStringList Digikam::DMetadata::getXmpKeywords ( ) const
```

Return a strings list of Xmp keywords from image. Return an empty list if no keyword are set.

### 6.425.2.13 getXmpSubCategories()

```
QStringList Digikam::DMetadata::getXmpSubCategories ( ) const
```

Return a strings list of Xmp sub-categories from image. Return an empty list if no sub-category are set.

### 6.425.2.14 getXmpSubjects()

```
QStringList Digikam::DMetadata::getXmpSubjects ( ) const
```

Return a strings list of Xmp subjects from image. Return an empty list if no subject are set.

### 6.425.2.15 load()

```
bool Digikam::DMetadata::load (
    const QString & filePath,
    bool videoAll = false,
    Backend * backend = nullptr )
```

Re-implemented from [MetaEngine](#) to use `libraw` identify, `libheif`, `ffmpeg` probe, and `ImageMagick` identify methods if `Exiv2` failed. If `backend` is non null, return the backend used to populate metadata (`Exiv2`). See [MetaEngine::Backend](#) enum for details.

#### 6.425.2.16 loadUsingFFmpeg()

```
bool Digikam::DMetadata::loadUsingFFmpeg (
    const QString & filePath )
```

Try to extract metadata using FFMpeg probe method (libav).

#### 6.425.2.17 loadUsingRawEngine()

```
bool Digikam::DMetadata::loadUsingRawEngine (
    const QString & filePath )
```

Try to extract metadata using Raw Engine identify method (libraw).

#### 6.425.2.18 mSecTimeStamp()

```
bool Digikam::DMetadata::mSecTimeStamp (
    const char *const exifTagName,
    int & ms ) const
```

Extract milliseconds time-stamp of photo from an Exif tag and store it to 'ms'. Returns true if data are extracted.

#### 6.425.2.19 possibleValuesForEnumField()

```
QMap< int, QString > Digikam::DMetadata::possibleValuesForEnumField (
    MetadataInfo::Field field ) [static]
```

Returns a map of possible enum values and their user-presentable, i18n'ed representation. Valid fields are those which are described as "enum from" or "bit mask from" in metadatainfo.h. Int, enum from libMetaEngine

Int, enum from Exif

Int, enum from Exif

Int, enum from Exif

Int, enum from Exif

int, enum from Exif

Int, bit mask from Exif

#### 6.425.2.20 removeExifColorSpace()

```
bool Digikam::DMetadata::removeExifColorSpace ( ) const
```

Remove the Exif color space identification from the image.

#### 6.425.2.21 removeFromXmpTagStringBag()

```
bool Digikam::DMetadata::removeFromXmpTagStringBag (
    const char *const xmpTagName,
    const QStringList & entriesToRemove ) const
```

Remove those Xmp tag entries that are listed in entriesToRemove from the entries in metadata. Return true if tag entries are no longer contained in metadata. All other entries are preserved.

#### 6.425.2.22 removeItemFacesMap()

```
bool Digikam::DMetadata::removeItemFacesMap ( ) const
```

Remove Images Face Map tags from Picassa/Metadatagroup format.

#### 6.425.2.23 removeXmpKeywords()

```
bool Digikam::DMetadata::removeXmpKeywords (
    const QStringList & keywordsToRemove )
```

Remove those Xmp keywords that are listed in keywordsToRemove from the keywords in metadata. Return true if keywords are no longer contained in metadata.

#### 6.425.2.24 removeXmpSubCategories()

```
bool Digikam::DMetadata::removeXmpSubCategories (
    const QStringList & categoriesToRemove )
```

Remove those Xmp sub-categories that are listed in categoriesToRemove from the sub-categories in metadata. Return true if subjects are no longer contained in metadata.

#### 6.425.2.25 removeXmpSubjects()

```
bool Digikam::DMetadata::removeXmpSubjects (
    const QStringList & subjectsToRemove )
```

Remove those Xmp subjects that are listed in subjectsToRemove from the subjects in metadata. Return true if subjects are no longer contained in metadata.

#### 6.425.2.26 setIccProfile()

```
bool Digikam::DMetadata::setIccProfile (
    const IccProfile & profile )
```

Sets the [IccProfile](#) embedded in the Exif metadata.

#### 6.425.2.27 setItemFacesMap()

```
bool Digikam::DMetadata::setItemFacesMap (
    const QMap< QString, QVariant > & facesPath,
    bool write,
    const QSize & size = QSize() ) const
```

Set Images Face Map tags in Picassa/Metadatagroup format.

## Parameters

<i>facesPath</i>	The face map to register in metadata based on tags stored in Picassa/Metadatagroup
<i>write</i>	If true all faces will be written, else update mode search if at least a face tag exist and write if true.
<i>size</i>	The size of the area grouping all faces in image.

**6.425.2.28 setXmpKeywords()**

```
bool Digikam::DMetadata::setXmpKeywords (
    const QStringList & newKeywords ) const
```

Set Xmp keywords using a list of strings defined by 'newKeywords' parameter. The existing keywords from image are preserved. The method will compare all new keywords with all already existing keywords to prevent duplicate entries in image. Return true if keywords have been changed in metadata.

**6.425.2.29 setXmpSubCategories()**

```
bool Digikam::DMetadata::setXmpSubCategories (
    const QStringList & newSubCategories ) const
```

Set Xmp sub-categories using a list of strings defined by 'newSubCategories' parameter. The existing sub-categories from image are preserved. The method will compare all new sub-categories with all already existing sub-categories to prevent duplicate entries in image. Return true if sub-categories have been changed in metadata.

**6.425.2.30 setXmpSubjects()**

```
bool Digikam::DMetadata::setXmpSubjects (
    const QStringList & newSubjects ) const
```

Set Xmp subjects using a list of strings defined by 'newSubjects' parameter. The existing subjects from image are preserved. The method will compare all new subject with all already existing subject to prevent duplicate entries in image. Return true if subjects have been changed in metadata.

**6.425.2.31 valueToString()**

```
QString Digikam::DMetadata::valueToString (
    const QVariant & value,
    MetadataInfo::Field field ) [static]
```

Convert a QVariant value of the specified field to a user-presentable, i18n'ed string. The QVariant must be of the type as specified in metadatainfo.h and as obtained by getMetadataField.

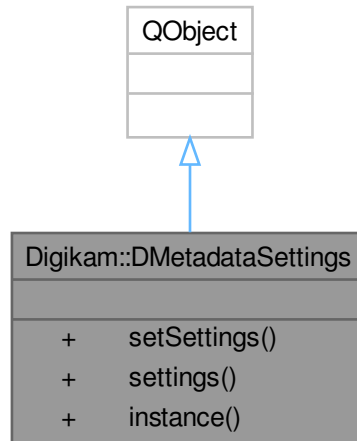
**6.425.2.32 videoColorModelToString()**

```
QString Digikam::DMetadata::videoColorModelToString (
    VIDEOCOLORMODEL videoColorModel ) [static]
```

Helper method to translate enum values to user presentable strings

## 6.426 Digikam::DMetadataSettings Class Reference

Inheritance diagram for Digikam::DMetadataSettings:



### Signals

- void **signalDMetadataSettingsChanged** (const [DMetadataSettingsContainer](#) &current, const [DMetadataSettingsContainer](#) &previous)
- void **signalSettingsChanged** ()

### Public Member Functions

- void [setSettings](#) (const [DMetadataSettingsContainer](#) &settings)
- [DMetadataSettingsContainer](#) [settings](#) () const

### Static Public Member Functions

- static [DMetadataSettings](#) \* [instance](#) ()

### Friends

- class [DMetadataSettingsCreator](#)

## 6.426.1 Member Function Documentation

### 6.426.1.1 instance()

```
DMetadataSettings * Digikam::DMetadataSettings::instance ( ) [static]
```

Global container for Metadata settings. All accessor methods are thread-safe.

### 6.426.1.2 setSettings()

```
void Digikam::DMetadadataSettings::setSettings (
    const DMetadadataSettingsContainer & settings )
```

Sets the current Metadata settings and writes them to config.

### 6.426.1.3 settings()

```
DMetadadataSettingsContainer Digikam::DMetadadataSettings::settings ( ) const
```

Returns the current Metadata settings.

## 6.427 Digikam::DMetadadataSettingsContainer Class Reference

### Public Member Functions

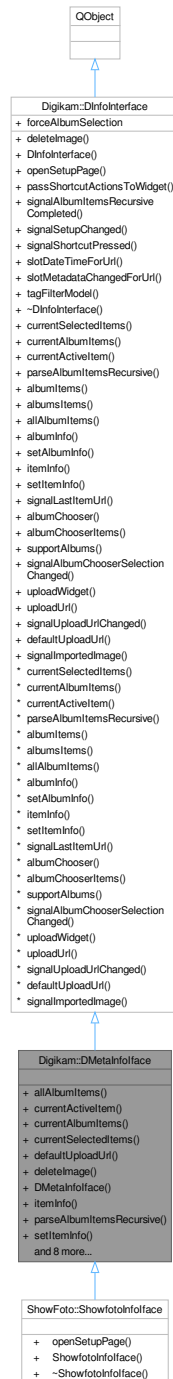
- void **addMapping** (const QString &key)
- void **defaultValues** ()  
*defaultValues - default namespaces used by digiKam*
- **DMetadadataSettingsContainer** (const [DMetadadataSettingsContainer](#) &other)
- QList< [NamespaceEntry](#) > & **getReadMapping** (const QString &key) const
- QList< [NamespaceEntry](#) > & **getWriteMapping** (const QString &key) const
- QList< QString > **mappingKeys** () const
- **DMetadadataSettingsContainer** & **operator=** (const [DMetadadataSettingsContainer](#) &other)
- void **readFromConfig** (KConfigGroup &group)
- bool **readingAllTags** () const
- void **setReadingAllTags** (bool b)
- void **setUnifyReadWrite** (bool b)
- QString **translateMappingKey** (const QString &key) const
- bool **unifyReadWrite** () const
- void **writeToConfig** (KConfigGroup &group) const

### 6.427.1 Detailed Description

The class [DMetadadataSettingsContainer](#) is designed to dynamically add namespaces.

## 6.428 Digikam::DMetaInfoface Class Reference

Inheritance diagram for Digikam::DMetaInfoface:



### Public Member Functions

- `QList< QUrl > allAlbumItems ()` const override
- `QUrl currentActiveItem ()` const override



- QList< QUrl > [currentAlbumItems](#) () const override
- QList< QUrl > [currentSelectedItems](#) () const override
  - Low level items and albums methods.*
- QUrl [defaultUploadUrl](#) () const override
  - Url to upload new items without to use album selector.*
- void [deleteImage](#) (const QUrl &url) override
  - Manipulate with item.*
- **DMetalnfolface** (QObject \*const parent, const QList< QUrl > &lst, const QUrl &currentActive)
- [DInfoMap itemInfo](#) (const QUrl &) const override
- void [parseAlbumItemsRecursive](#) () override
- void [setItemInfo](#) (const QUrl &, const [DInfoMap](#) &) override
- Q\_SIGNAL void [signalItemChanged](#) (const QUrl &url)
- Q\_SIGNAL void [signalRemoveImageFromAlbum](#) (const QUrl &)
- Q\_SLOT void [slotDateTimeForUrl](#) (const QUrl &url, const QDateTime &dt, bool updModDate) override
  - Slot to call when date time stamp from item is changed.*
- Q\_SLOT void [slotMetadataChangedForUrl](#) (const QUrl &url) override
  - Slot to call when something in metadata from item is changed.*
- bool [supportAlbums](#) () const override
- QUrl [uploadUrl](#) () const override
- QWidget \* [uploadWidget](#) (QWidget \*const parent) const override
  - Album selector view methods (to upload items from an external place).*

## Public Member Functions inherited from [Digikam::DInfoInterface](#)

- **DInfoInterface** (QObject \*const parent)
- virtual void [openSetupPage](#) (SetupPage page)
  - Open configuration dialog page.*
- virtual QMap< QString, QString > [passShortcutActionsToWidget](#) (QWidget \*const) const
  - Pass extra shortcut actions to widget and return prefixes of shortcuts.*
- Q\_SIGNAL void [signalAlbumItemsRecursiveCompleted](#) (const QList< QUrl > &imageList)
- Q\_SIGNAL void [signalSetupChanged](#) ()
- Q\_SIGNAL void [signalShortcutPressed](#) (const QString &shortcut, int val)
- virtual QAbstractItemModel \* [tagFilterModel](#) ()
  - Return an instance of tag filter model if host application support this feature, else null pointer.*
- virtual QList< QUrl > [albumItems](#) (int) const
- virtual QList< QUrl > [albumsItems](#) (const [DAlbumIDs](#) &) const
- virtual [DInfoMap](#) [albumInfo](#) (int) const
- virtual void [setAlbumInfo](#) (int, const [DInfoMap](#) &) const
- Q\_SIGNAL void [signalLastItemUrl](#) (const QUrl &)
- virtual QWidget \* [albumChooser](#) (QWidget \*const parent) const
  - Albums chooser view methods (to use items from albums before to process).*
- virtual [DAlbumIDs](#) [albumChooserItems](#) () const
- Q\_SIGNAL void [signalAlbumChooserSelectionChanged](#) ()
- Q\_SIGNAL void [signalUploadUrlChanged](#) ()
- Q\_SIGNAL void [signalImportedImage](#) (const QUrl &)

## Additional Inherited Members

### Public Types inherited from [Digikam::DInfoInterface](#)

- typedef QList< int > **DAlbumIDs**  
*List of [Album](#) ids.*
- typedef QMap< QString, QVariant > **DInfoMap**  
*Map of properties name and value.*
- enum **SetupPage** { **ExifToolPage** = 0 , **ImageQualityPage** }

### Public Attributes inherited from [Digikam::DInfoInterface](#)

- bool **forceAlbumSelection** = false

## 6.428.1 Member Function Documentation

### 6.428.1.1 allAlbumItems()

```
QList< QUrl > Digikam::DMetaInfoIface::allAlbumItems ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.2 currentActiveItem()

```
QUrl Digikam::DMetaInfoIface::currentActiveItem ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.3 currentAlbumItems()

```
QList< QUrl > Digikam::DMetaInfoIface::currentAlbumItems ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.4 currentSelectedItems()

```
QList< QUrl > Digikam::DMetaInfoIface::currentSelectedItems ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.5 defaultUploadUrl()

```
QUrl Digikam::DMetaInfoIface::defaultUploadUrl ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.6 deleteImage()

```
void Digikam::DMetaInfoIface::deleteImage (
    const QUrl & url ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.7 itemInfo()

```
DMetaInfoIface::DInfoMap Digikam::DMetaInfoIface::itemInfo (
    const QUrl & url ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.8 parseAlbumItemsRecursive()

```
void Digikam::DMetaInfoIface::parseAlbumItemsRecursive ( ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.9 setItemInfo()

```
void Digikam::DMetaInfoIface::setItemInfo (
    const QUrl & url,
    const DInfoMap & map ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.10 slotDateTimeForUrl()

```
void Digikam::DMetaInfoIface::slotDateTimeForUrl (
    const QUrl & url,
    const QDateTime & dt,
    bool updModDate ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.11 slotMetadataChangedForUrl()

```
void Digikam::DMetaInfoIface::slotMetadataChangedForUrl (
    const QUrl & url ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

### 6.428.1.12 supportAlbums()

```
bool Digikam::DMetaInfoIface::supportAlbums ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

**6.428.1.13 uploadUrl()**

```
QUrl Digikam::DMetaInfoIface::uploadUrl ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

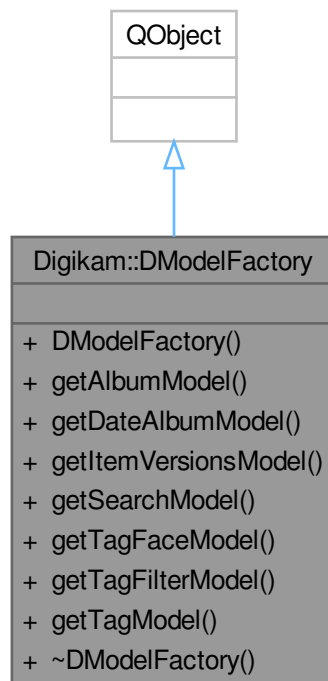
**6.428.1.14 uploadWidget()**

```
QWidget * Digikam::DMetaInfoIface::uploadWidget (
    QWidget *const parent ) const [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

**6.429 Digikam::DModelFactory Class Reference**

Inheritance diagram for Digikam::DModelFactory:

**Public Member Functions**

- [AlbumModel](#) \* `getAlbumModel ()` const
- [DateAlbumModel](#) \* `getDateAlbumModel ()` const
- [ItemVersionsModel](#) \* `getItemVersionsModel ()` const
- [SearchModel](#) \* `getSearchModel ()` const
- [TagModel](#) \* `getTagFaceModel ()` const
- [TagModel](#) \* `getTagFilterModel ()` const
- [TagModel](#) \* `getTagModel ()` const

### 6.429.1 Detailed Description

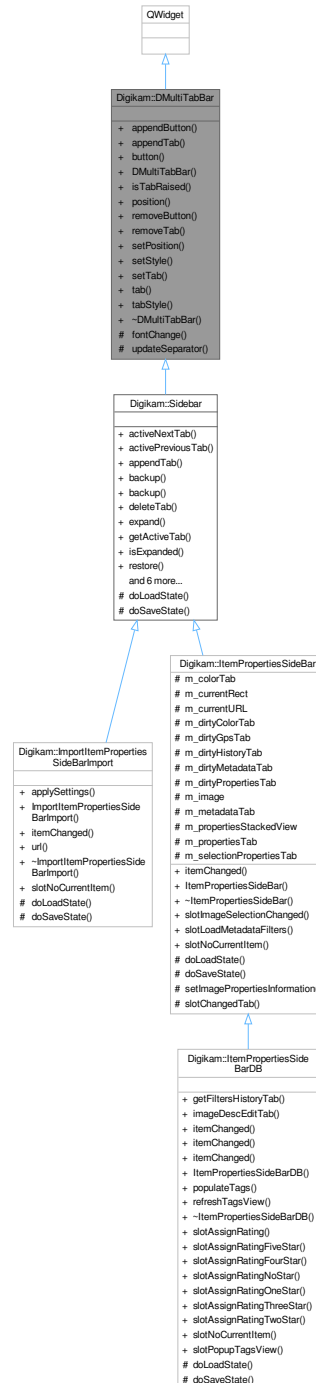
This class is simply a factory of all models that build the core of the digikam application.

Author

jwienke

## 6.430 Digikam::DMultiTabBar Class Reference

Inheritance diagram for Digikam::DMultiTabBar:



### Classes

- class [Private](#)

## Public Types

- enum `TextStyle` { `ActiveIconText` = 0 , `AllIconsText` = 2 }

## Public Member Functions

- void `appendButton` (const `QIcon` &pic, int id=-1, `QMenu` \*const popup=nullptr, const `QString` &not\_used\_↔ yet=`QString`())
- void `appendTab` (const `QIcon` &pic, int id=-1, const `QString` &text=`QString`())
- `DMultiTabBarButton` \* `button` (int id) const
- `DMultiTabBar` (Qt::Edge pos, `QWidget` \*const parent=nullptr)
- bool `isTabRaised` (int id) const
- Qt::Edge `position` () const
- void `removeButton` (int id)
- void `removeTab` (int id)
- void `setPosition` (Qt::Edge pos)
- void `setStyle` (`TextStyle` style)
- void `setTab` (int id, bool state)
- `DMultiTabBarTab` \* `tab` (int id) const
- `TextStyle` `tabStyle` () const

## Protected Member Functions

- virtual void `fontChange` (const `QFont` &)
- void `updateSeparator` ()

## Friends

- class `DMultiTabBarButton`

## 6.430.1 Detailed Description

A Widget for horizontal and vertical tabs.

## 6.430.2 Member Enumeration Documentation

### 6.430.2.1 TextStyle

```
enum Digikam::DMultiTabBar::TextStyle
```

The list of available styles for `DMultiTabBar`

#### Enumerator

<code>ActiveIconText</code>	Always shows icon, only show the text of active tabs.
<code>AllIconsText</code>	Always shows the text and icons.

### 6.430.3 Member Function Documentation

#### 6.430.3.1 appendButton()

```
void Digikam::DMultiTabBar::appendButton (
    const QIcon & pic,
    int id = -1,
    QMenu *const popup = nullptr,
    const QString & not_used_yet = QString() )
```

append a new button to the button area. The button can later on be accessed with button(ID) eg for connecting signals to it

##### Parameters

<i>pic</i>	a icon for the button
<i>id</i>	an arbitrary ID value. It will be emitted in the clicked signal for identifying the button if more than one button is connected to a signals.
<i>popup</i>	A popup menu which should be displayed if the button is clicked
<i>not_used_yet</i>	will be used for a popup text in the future

#### 6.430.3.2 appendTab()

```
void Digikam::DMultiTabBar::appendTab (
    const QIcon & pic,
    int id = -1,
    const QString & text = QString() )
```

append a new tab to the tab area. It can be accessed later on with tabb(id);

##### Parameters

<i>pic</i>	a icon for the tab
<i>id</i>	an arbitrary ID which can be used later on to identify the tab
<i>text</i>	if a mode with text is used it will be the tab text, otherwise a mouse over hint

#### 6.430.3.3 button()

```
DMultiTabBarButton * Digikam::DMultiTabBar::button (
    int id ) const
```

get a pointer to a button within the button area identified by its ID

#### 6.430.3.4 isTabRaised()

```
bool Digikam::DMultiTabBar::isTabRaised (
    int id ) const
```

return the state of a tab, identified by its ID



### 6.430.3.5 position()

```
Qt::Edge Digikam::DMultiTabBar::position ( ) const
```

get the tabbar position.

#### Returns

position

### 6.430.3.6 removeButton()

```
void Digikam::DMultiTabBar::removeButton (
    int id )
```

remove a button with the given ID

### 6.430.3.7 removeTab()

```
void Digikam::DMultiTabBar::removeTab (
    int id )
```

remove a tab with a given ID

### 6.430.3.8 setPosition()

```
void Digikam::DMultiTabBar::setPosition (
    Qt::Edge pos )
```

set the real position of the widget.

#### Parameters

<i>pos</i>	if the mode is horizontal, only use top, bottom, if it is vertical use left or right
------------	--

### 6.430.3.9 setStyle()

```
void Digikam::DMultiTabBar::setStyle (
    TextStyle style )
```

set the display style of the tabs

### 6.430.3.10 setTab()

```
void Digikam::DMultiTabBar::setTab (
    int id,
    bool state )
```

set a tab to "raised"

## Parameters

<i>id</i>	The ID of the tab to manipulate
<i>state</i>	true == activated/raised, false == not active

**6.430.3.11 tab()**

```
DMultiTabBarTab * Digikam::DMultiTabBar::tab (
    int id ) const
```

get a pointer to a tab within the tab area, identified by its ID

**6.430.3.12 tabStyle()**

```
DMultiTabBar::TextStyle Digikam::DMultiTabBar::tabStyle ( ) const
```

get the display style of the tabs

## Returns

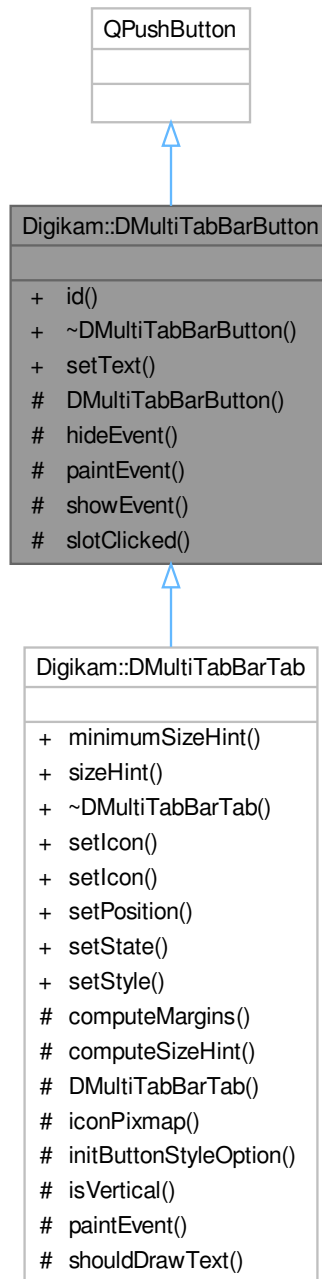
display style

**6.431 Digikam::DMultiTabBar::Private Class Reference****Public Attributes**

- QFrame \* **btnTabSep** = nullptr
- QList< [DMultiTabBarButton](#) \* > **buttons**
- [DMultiTabBarFrame](#) \* **internal** = nullptr
- QBoxLayout \* **layout** = nullptr
- Qt::Edge **position** = Qt::LeftEdge

## 6.432 Digikam::DMultiTabBarButton Class Reference

Inheritance diagram for Digikam::DMultiTabBarButton:



### Public Slots

- void **setText** (const QString &text)

## Signals

- void [signalClicked](#) (int id)

## Public Member Functions

- int **id** () const

## Protected Slots

- virtual void **slotClicked** ()

## Protected Member Functions

- **DMultiTabBarButton** (const QIcon &pic, const QString &, int id, QWidget \*const parent)
- void **hideEvent** (QHideEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **showEvent** (QShowEvent \*) override

## Friends

- class **DMultiTabBar**

## 6.432.1 Member Function Documentation

### 6.432.1.1 signalClicked

```
void Digikam::DMultiTabBarButton::signalClicked (  
    int id ) [signal]
```

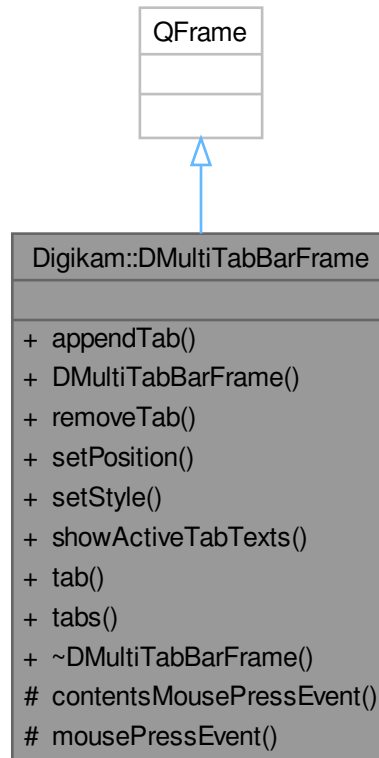
this is emitted if the button is clicked

#### Parameters

<i>id</i>	the ID identifying the button
-----------	-------------------------------

## 6.433 Digikam::DMultiTabBarFrame Class Reference

Inheritance diagram for Digikam::DMultiTabBarFrame:



### Classes

- class [Private](#)

### Public Member Functions

- void **appendTab** (const QIcon &, int=-1, const QString &=QString())
- **DMultiTabBarFrame** (QWidget \*const parent, Qt::Edge pos)
- void **removeTab** (int)
- void **setPosition** (Qt::Edge pos)
- void **setStyle** ([DMultiTabBar::TextStyle](#) style)
- void **showActiveTabTexts** (bool show)
- [DMultiTabBarTab](#) \* **tab** (int) const
- QList< [DMultiTabBarTab](#) \* > \* **tabs** ()

### Protected Member Functions

- virtual void [contentsMouseEvent](#) (QMouseEvent \*)
- void **mousePressEvent** (QMouseEvent \*) override

## Friends

- class **DMultiTabBar**

## 6.433.1 Member Function Documentation

### 6.433.1.1 contentsMouseEvent()

```
void Digikam::DMultiTabBarFrame::contentsMouseEvent (
    QMouseEvent * e ) [protected], [virtual]
```

Reimplemented from QScrollView in order to ignore all mouseEvents on the viewport, so that the parent can handle them.

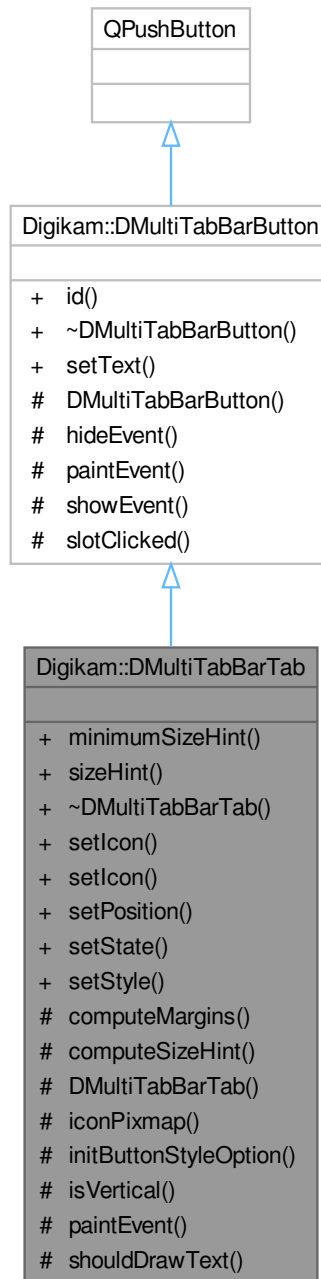
## 6.434 Digikam::DMultiTabBarFrame::Private Class Reference

### Public Attributes

- QBoxLayout \* **mainLayout** = nullptr
- Qt::Edge **position** = Qt::LeftEdge
- [DMultiTabBar::TextStyle](#) **style** = [DMultiTabBar::AllIconsText](#)
- QList< [DMultiTabBarTab](#) \* > **tabs**

## 6.435 Digikam::DMultiTabBarTab Class Reference

Inheritance diagram for Digikam::DMultiTabBarTab:



### Classes

- class [Private](#)

### Public Slots

- void **setIcon** (const QIcon &)
- void **setIcon** (const QString &)
- void **setPosition** (Qt::Edge)
- void **setState** (bool state)
- void **setStyle** (DMultiTabBar::TextStyle)

### Public Slots inherited from [Digikam::DMultiTabBarButton](#)

- void **setText** (const QString &text)

### Public Member Functions

- QSize **minimumSizeHint** () const override
- QSize **sizeHint** () const override

### Public Member Functions inherited from [Digikam::DMultiTabBarButton](#)

- int **id** () const

### Protected Member Functions

- void **computeMargins** (int \*hMargin, int \*vMargin) const
- QSize **computeSizeHint** (bool withText) const
- [DMultiTabBarTab](#) (const QIcon &pic, const QString &, int id, QWidget \*const parent, Qt::Edge pos, [DMultiTabBar::TextStyle](#) style)
- QPixmap **iconPixmap** () const
- void **initButtonStyleOption** (QStyleOptionToolButton \*opt) const
- bool **isVertical** () const
- void **paintEvent** (QPaintEvent \*) override
- bool **shouldDrawText** () const

### Protected Member Functions inherited from [Digikam::DMultiTabBarButton](#)

- [DMultiTabBarButton](#) (const QIcon &pic, const QString &, int id, QWidget \*const parent)
- void **hideEvent** (QHideEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **showEvent** (QShowEvent \*) override

### Friends

- class [DMultiTabBarFrame](#)

### Additional Inherited Members

### Signals inherited from [Digikam::DMultiTabBarButton](#)

- void [signalClicked](#) (int id)



## Protected Slots inherited from [Digikam::DMultiTabBarButton](#)

- virtual void `slotClicked ()`

### 6.435.1 Constructor & Destructor Documentation

#### 6.435.1.1 `DMultiTabBarTab()`

```
Digikam::DMultiTabBarTab::DMultiTabBarTab (
    const QIcon & pic,
    const QString & text,
    int id,
    QWidget *const parent,
    Qt::Edge pos,
    DMultiTabBar::TextStyle style ) [protected]
```

This class should never be created except with the `appendTab` call of [DMultiTabBar](#)

### 6.435.2 Member Function Documentation

#### 6.435.2.1 `setPosition`

```
void Digikam::DMultiTabBarTab::setPosition (
    Qt::Edge pos ) [slot]
```

this is used internally, but can be used by the user. It the according call of [DMultiTabBar](#) is invoked though this modifications will be overwritten

#### 6.435.2.2 `setState`

```
void Digikam::DMultiTabBarTab::setState (
    bool state ) [slot]
```

set the active state of the tab

##### Parameters

<i>state</i>	true==active false==not active
--------------	--------------------------------

#### 6.435.2.3 `setStyle`

```
void Digikam::DMultiTabBarTab::setStyle (
    DMultiTabBar::TextStyle style ) [slot]
```

this is used internally, but can be used by the user. It the according call of [DMultiTabBar](#) is invoked though this modifications will be overwritten

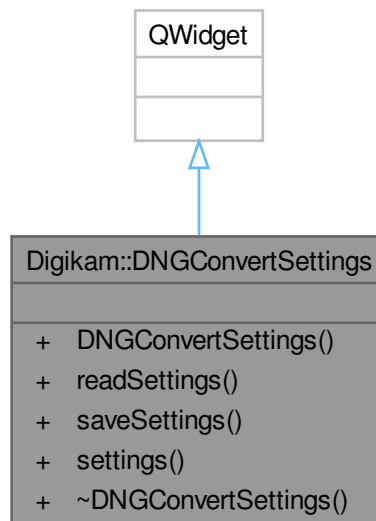
## 6.436 Digikam::DMultiTabBarTab::Private Class Reference

### Public Attributes

- Qt::Edge **position** = Qt::LeftEdge
- [DMultiTabBar::TextStyle](#) **style** = [DMultiTabBar::AllIconsText](#)

## 6.437 Digikam::DNGConvertSettings Class Reference

Inheritance diagram for Digikam::DNGConvertSettings:



### Signals

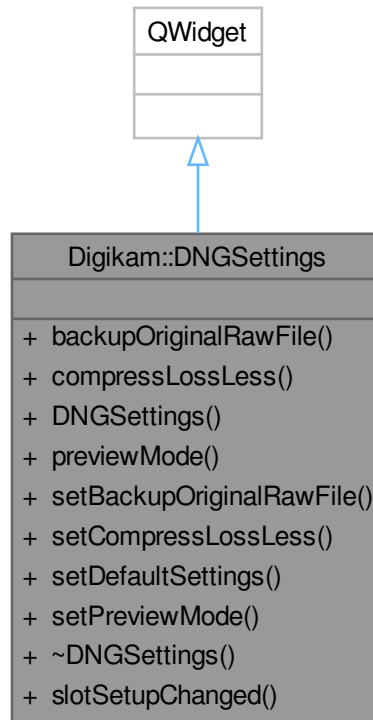
- void **signalDownloadNameChanged** ()

### Public Member Functions

- **DNGConvertSettings** (`QWidget *const parent=nullptr`)
- void **readSettings** (`const KConfigGroup &group`)
- void **saveSettings** (`KConfigGroup &group`)
- void **settings** ([DownloadSettings](#) `*const settings`)

## 6.438 Digikam::DNGSettings Class Reference

Inheritance diagram for Digikam::DNGSettings:



### Public Slots

- void **slotSetupChanged** ()  
*To handle changes from host application [Setup](#) dialog.*

### Signals

- void **signalSettingsChanged** ()
- void **signalSetupExifTool** ()

### Public Member Functions

- bool **backupOriginalRawFile** () const
- bool **compressLossLess** () const
- **DNGSettings** (QWidget \*const parent=nullptr)
- int **previewMode** () const
- void **setBackupOriginalRawFile** (bool b)
- void **setCompressLossLess** (bool b)
- void **setDefaultSettings** ()
- void **setPreviewMode** (int mode)

## 6.439 Digikam::DNGWriter Class Reference

### Classes

- class [Private](#)

### Public Types

- enum [ConvertError](#) { [PROCESS\\_CONTINUE](#) = 1, [PROCESS\\_COMPLETE](#) = 0, [PROCESS\\_FAILED](#) = -1, [PROCESS\\_CANCELED](#) = -2, [FILE\\_NOT\\_SUPPORTED](#) = -3, [DNG\\_SDK\\_INTERNAL\\_ERROR](#) = -4 }
- enum [JPEGPreview](#) { [NONE](#) = 0, [MEDIUM](#), [FULL\\_SIZE](#) }

### Public Member Functions

- bool **backupOriginalRawFile** () const
- void **cancel** ()
- bool **compressLossLess** () const
- int **convert** ()
- QString **inputFile** () const
- QString **outputFile** () const
- int **previewMode** () const
- void **reset** ()
- void **setBackupOriginalRawFile** (bool b)
- void **setCompressLossLess** (bool b)
- void **setInputFile** (const QString &filePath)
- void **setOutputFile** (const QString &filePath)
- void **setPreviewMode** (int mode)
- void **setUpdateFileDate** (bool b)
- bool **updateFileDate** () const

### Static Public Member Functions

- static QString **dngSdkVersion** ()
- static QString **xmpSdkVersion** ()

## 6.439.1 Member Enumeration Documentation

### 6.439.1.1 ConvertError

enum [Digikam::DNGWriter::ConvertError](#)

#### Enumerator

PROCESS_CONTINUE	Current stages is done.
PROCESS_COMPLETE	All stages done.
PROCESS_FAILED	A failure happen while processing.
PROCESS_CANCELED	User has canceled processing.
FILE_NOT_SUPPORTED	Raw file format is not supported by converter.
DNG_SDK_INTERNAL_ERROR	Adobe DNG SDK has generated an error while processing.

### 6.439.1.2 JPEGPreview

enum [Digikam::DNGWriter::JPEGPreview](#)

#### Enumerator

NONE	No preview will be generated.
MEDIUM	A medium size preview will be generated.
FULL_SIZE	A full size preview will be generated.

## 6.440 Digikam::DNGWriter::Private Class Reference

### Public Types

- enum **DNGBayerPattern** {  
**Unknown** = 1 , **LinearRaw** , **Standard** , **Fuji** ,  
**Fuji6x6** , **FourColor** }

### Public Member Functions

- int **backupRaw** ([DNGWriterHost](#) &host, [AutoPtr](#)< [dng\\_negative](#) > &negative)
- void **cleanup** ()
- int **createNegative** ([AutoPtr](#)< [dng\\_negative](#) > &negative, [DRawInfo](#) \*const identify)
- int **debugExtractedRAWData** (const [QByteArray](#) &rawData)
- [QString](#) **dngBayerPatternToString** (int pattern) const
- [dng\\_date\\_time](#) **dngDateTime** (const [QDateTime](#) &qDT) const
- [QString](#) **dngErrorCodeToString** (int errorCode) const
- int **exifToolPostProcess** ()
- int **exportTarget** ([DNGWriterHost](#) &host, [AutoPtr](#)< [dng\\_negative](#) > &negative, [AutoPtr](#)< [dng\\_image](#) > &image)
- bool **fujiRotate** ([QByteArray](#) &rawData, [DRawInfo](#) &identify) const
- int **identMosaic** ([DRawInfo](#) \*const identify, [DRawInfo](#) \*const identifyMake)
- int **importRaw** ([DRawInfo](#) \*const identify, [DRawInfo](#) \*const identifyMake)
- Private** ([DNGWriter](#) \*const dd)
- void **reset** ()
- int **storeExif** ([DNGWriterHost](#) &host, [AutoPtr](#)< [dng\\_negative](#) > &negative, [DRawInfo](#) \*const identify, [DRawInfo](#) \*const identifyMake, [DMetadata](#) \*const meta)
- int **storeMakernote** ([DNGWriterHost](#) &host, [AutoPtr](#)< [dng\\_negative](#) > &negative, [DRawInfo](#) \*const identify, [DRawInfo](#) \*const identifyMake, [DMetadata](#) \*const meta)
- int **storeXmp** ([DNGWriterHost](#) &host, [AutoPtr](#)< [dng\\_negative](#) > &negative, [DRawInfo](#) \*const identify, [DRawInfo](#) \*const identifyMake, [DMetadata](#) \*const meta)

## Public Attributes

- dng\_rect **activeArea**
- int **activeHeight** = 0
- int **activeWidth** = 0
- bool **backupOriginalRawFile** = false
- DNGBayerPattern **bayerPattern** = Unknown
- bool **cancel** = false
- QString **dngFilePath**
- dng\_exif \* **exif** = nullptr  
*Instance to Exif DNG SDK container.*
- QDateTime **fileDate**
- uint32 **filter** = 0
- int **height** = 0
- QString **inputFile**
- QFileInfo **inputInfo**
- bool **jpegLossLessCompression** = true
- bool **metaLoaded** = false  
*Set to true if metadata are properly loaded at Exif stage.*
- dng\_date\_time\_info **orgDateTimeInfo**
- QString **outputFile**
- int **outputHeight** = 0
- QFileInfo **outputInfo**
- int **outputWidth** = 0
- DNGWriter \* **parent** = nullptr  
*Parent class instance.*
- int **previewMode** = DNGWriter::FULL\_SIZE
- QByteArray **rawData**
- bool **updateFileDate** = false
- int **width** = 0

## 6.440.1 Member Function Documentation

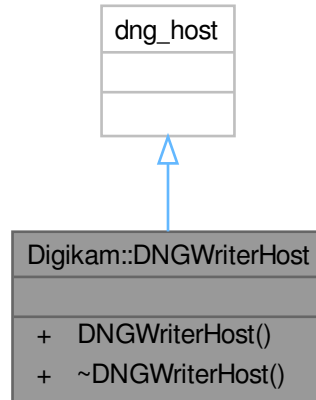
### 6.440.1.1 debugExtractedRAWData()

```
int Digikam::DNGWriter::Private::debugExtractedRAWData (
    const QByteArray & rawData )
```

Code to hack RAW data extraction.

## 6.441 Digikam::DNGWriterHost Class Reference

Inheritance diagram for Digikam::DNGWriterHost:

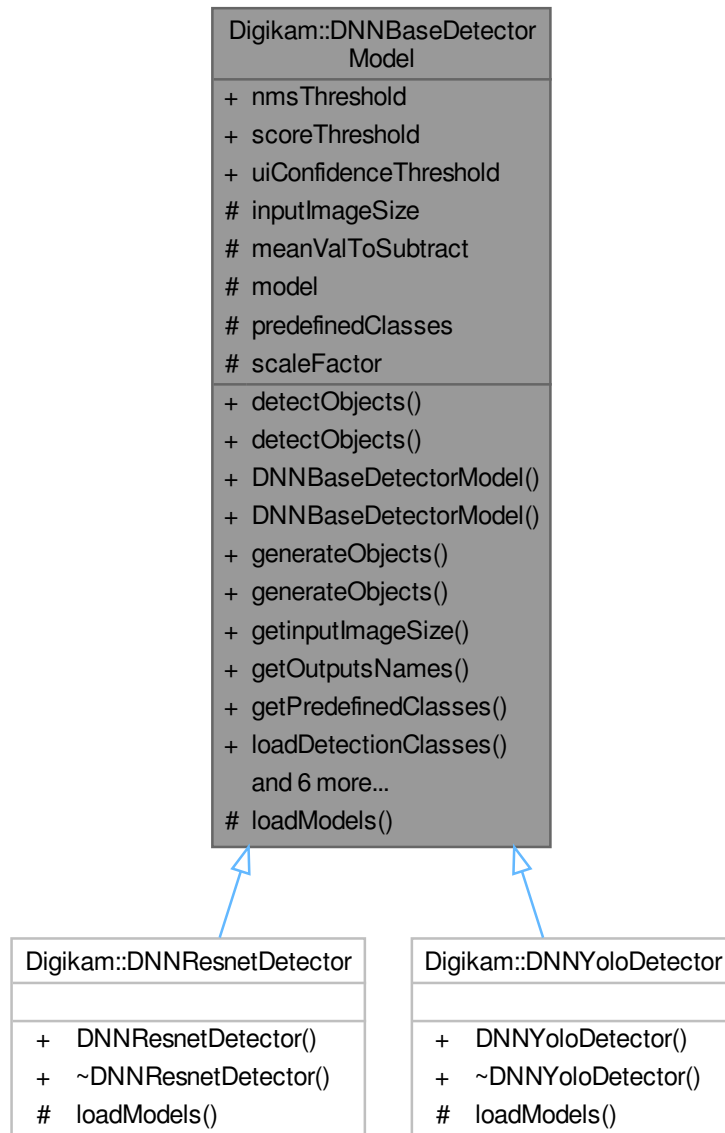


### Public Member Functions

- **DNGWriterHost** ([DNGWriter::Private](#) \*const priv, dng\_memory\_allocator \*const allocator=nullptr)

## 6.442 Digikam::DNNBaseDetectorModel Class Reference

Inheritance diagram for Digikam::DNNBaseDetectorModel:



### Public Member Functions

- virtual `QHash< QString, QVector< QRect > >` [detectObjects](#) (const cv::Mat &inputImage)
- virtual `QList< QHash< QString, QVector< QRect > > >` [detectObjects](#) (const std::vector< cv::Mat > &inputBatchImages)
- **DNNBaseDetectorModel** (float scale, const cv::Scalar &val, const cv::Size &inputImgSize)
- `QList< QString >` [generateObjects](#) (const cv::Mat &inputImage)
- `QList< QList< QString > >` [generateObjects](#) (const std::vector< cv::Mat > &inputImage)



- cv::Size [getImageSize](#) () const
- std::vector< cv::String > **getOutputsNames** () const
- virtual QList< QString > [getPredefinedClasses](#) () const
- QList< QString > **loadDetectionClasses** ()
- virtual QMap< QString, QVector< QRect > > **postprocess** (const cv::Mat &inputImage, const cv::Mat &out) const =0
- QList< QMap< QString, QVector< QRect > > > **postprocess** (const std::vector< cv::Mat > &inputBatchImages, const std::vector< cv::Mat > &outs) const
- std::vector< cv::Mat > **preprocess** (const cv::Mat &inputImage)
- std::vector< cv::Mat > **preprocess** (const std::vector< cv::Mat > &inputBatchImages)
- double **showInferenceTime** ()

### Static Public Attributes

- static float **nmsThreshold** = 0.4F  
*Threshold for nms suppression.*
- static float **scoreThreshold** = 0.45F  
*Threshold for class detection score.*
- static int [uiConfidenceThreshold](#) = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Threshold for bbox detection. It can be init and changed in the GUI.*

### Protected Member Functions

- virtual bool **loadModels** ()=0

### Protected Attributes

- cv::Size **inputImageSize**
- cv::Scalar **meanValToSubtract**
- [DNNModelBase](#) \* **model** = nullptr
- QList< QString > **predefinedClasses**
- float **scaleFactor** = 1.0F

## 6.442.1 Member Function Documentation

### 6.442.1.1 detectObjects() [1/2]

```
virtual QMap< QString, QVector< QRect > > Digikam::DNNBaseDetectorModel::detectObjects (
    const cv::Mat & inputImage ) [virtual]
```

detectObjects return the predicted objects and localization as well (if we use deeplearning for object detection like YOLO, etc) otherwise the map whose the key is the objects name and their values are empty.

### 6.442.1.2 detectObjects() [2/2]

```
QList< QMap< QString, QVector< QRect > > > Digikam::DNNBaseDetectorModel::detectObjects (
    const std::vector< cv::Mat > & inputBatchImages ) [virtual]
```

detectObjects in batch images (fixed batch size).

### 6.442.1.3 generateObjects() [1/2]

```
QList< QString > Digikam::DNNBaseDetectorModel::generateObjects (
    const cv::Mat & inputImage )
```

generateObjects in one image return just the predicted objects without locations of objects using for the assignment tagging names.

### 6.442.1.4 generateObjects() [2/2]

```
QList< QList< QString > > Digikam::DNNBaseDetectorModel::generateObjects (
    const std::vector< cv::Mat > & inputImage )
```

generateObjects in batch images return just the predicted objects without locations of objects using for the assignment tagging names.

### 6.442.1.5 getInputImageSize()

```
cv::Size Digikam::DNNBaseDetectorModel::getInputImageSize ( ) const
```

Return the input Image Size from Deep NN model.

### 6.442.1.6 getPredefinedClasses()

```
QList< QString > Digikam::DNNBaseDetectorModel::getPredefinedClasses ( ) const [virtual]
```

Get predefined objects according to selected model.

## 6.442.2 Member Data Documentation

### 6.442.2.1 uiConfidenceThreshold

```
int Digikam::DNNBaseDetectorModel::uiConfidenceThreshold = DNN_MODEL_THRESHOLD_NOT_SET [static]
```

setting 1000 will use the value from dnnmodels.conf if passed in

## 6.443 Digikam::DNNFaceDetectorBase Class Reference

Inheritance diagram for Digikam::DNNFaceDetectorBase:



### Public Member Functions

- virtual void **detectFaces** (const cv::Mat &inputImage, const cv::Size &paddedSize, std::vector< cv::Rect > &detectedBboxes)=0
- **DNNFaceDetectorBase** (float scale, const cv::Scalar &val, const cv::Size &inputImgSize)
- cv::Size **nnInputSizeRequired** () const
- virtual void **setFaceDetectionSize** ([FaceScanSettings::FaceDetectionSize](#) faceSize)

### Static Public Attributes

- static float **nmsThreshold** = 0.4F  
*Threshold for nms suppression.*
- static int **uiConfidenceThreshold** = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Threshold for bbox detection. It can be init and changed in the GUI.*

### Protected Member Functions

- void **correctBbox** (cv::Rect &bbox, const cv::Size &paddedSize) const
- void **selectBbox** (const cv::Size &paddedSize, float confidence, int left, int right, int top, int bottom, std::vector< float > &goodConfidences, std::vector< cv::Rect > &goodBoxes, std::vector< float > &doubtConfidences, std::vector< cv::Rect > &doubtBoxes) const

## Protected Attributes

- `cv::Size inputImageSize = cv::Size(300, 300)`
- `cv::Scalar meanValToSubtract = cv::Scalar(0.0, 0.0, 0.0)`
- `DNNModelBase * model = nullptr`
- `float scaleFactor = 1.0F`

## 6.443.1 Member Function Documentation

### 6.443.1.1 selectBbox()

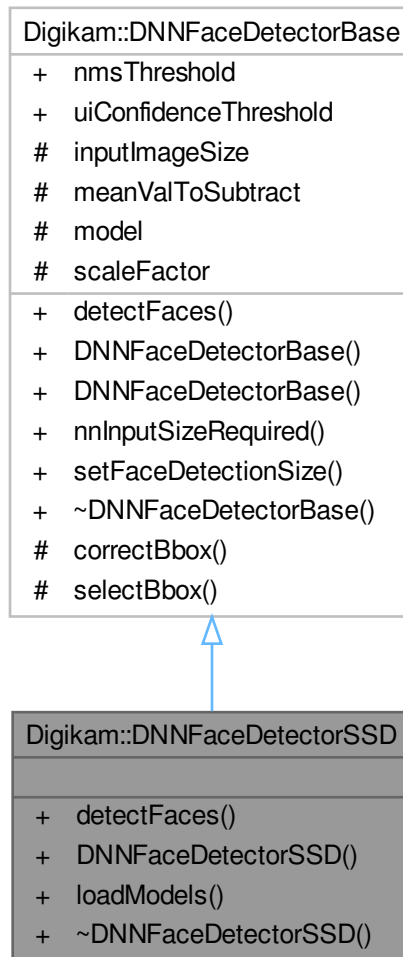
```
void Digikam::DNNFaceDetectorBase::selectBbox (
    const cv::Size & paddedSize,
    float confidence,
    int left,
    int right,
    int top,
    int bottom,
    std::vector< float > & goodConfidences,
    std::vector< cv::Rect > & goodBoxes,
    std::vector< float > & doubtConfidences,
    std::vector< cv::Rect > & doubtBoxes ) const [protected]
```

Classify bounding boxes detected. Good bounding boxes are defined as boxes that reside within the non-padded zone or those that are out only for min of (10% of padded range, 10% of bbox dim).

Bad bounding boxes are defined as boxes that have at maximum 25% of each dimension out of non-padded zone.

## 6.444 Digikam::DNNFaceDetectorSSD Class Reference

Inheritance diagram for Digikam::DNNFaceDetectorSSD:



### Public Member Functions

- void `detectFaces` (const cv::Mat &inputImage, const cv::Size &paddedSize, std::vector< cv::Rect > &detectedBboxes) override
- bool `loadModels` ()

### Public Member Functions inherited from [Digikam::DNNFaceDetectorBase](#)

- `DNNFaceDetectorBase` (float scale, const cv::Scalar &val, const cv::Size &inputImgSize)
- cv::Size `nnInputSizeRequired` () const
- virtual void `setFaceDetectionSize` ([FaceScanSettings::FaceDetectionSize](#) faceSize)

## Additional Inherited Members

### Static Public Attributes inherited from [Digikam::DNNFaceDetectorBase](#)

- static float **nmsThreshold** = 0.4F  
*Threshold for nms suppression.*
- static int **uiConfidenceThreshold** = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Threshold for bbox detection. It can be init and changed in the GUI.*

### Protected Member Functions inherited from [Digikam::DNNFaceDetectorBase](#)

- void **correctBbox** (cv::Rect &bbox, const cv::Size &paddedSize) const
- void **selectBbox** (const cv::Size &paddedSize, float confidence, int left, int right, int top, int bottom, std::vector< float > &goodConfidences, std::vector< cv::Rect > &goodBoxes, std::vector< float > &doubtConfidences, std::vector< cv::Rect > &doubtBoxes) const

### Protected Attributes inherited from [Digikam::DNNFaceDetectorBase](#)

- cv::Size **inputImageSize** = cv::Size(300, 300)
- cv::Scalar **meanValToSubtract** = cv::Scalar(0.0, 0.0, 0.0)
- [DNNModelBase](#) \* **model** = nullptr
- float **scaleFactor** = 1.0F

## 6.444.1 Member Function Documentation

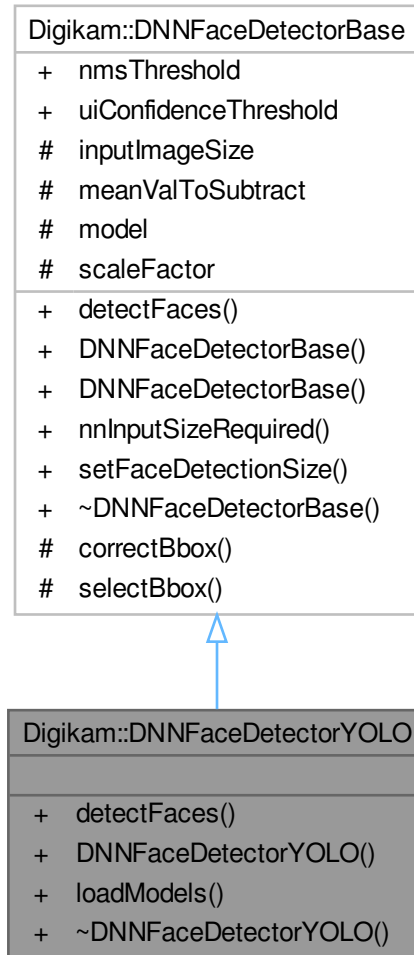
### 6.444.1.1 detectFaces()

```
void Digikam::DNNFaceDetectorSSD::detectFaces (
    const cv::Mat & inputImage,
    const cv::Size & paddedSize,
    std::vector< cv::Rect > & detectedBboxes ) [override], [virtual]
```

Implements [Digikam::DNNFaceDetectorBase](#).

## 6.445 Digikam::DNNFaceDetectorYOLO Class Reference

Inheritance diagram for Digikam::DNNFaceDetectorYOLO:



### Public Member Functions

- void `detectFaces` (const cv::Mat &inputImage, const cv::Size &paddedSize, std::vector< cv::Rect > &detectedBboxes) override
- bool `loadModels` ()

### Public Member Functions inherited from `Digikam::DNNFaceDetectorBase`

- `DNNFaceDetectorBase` (float scale, const cv::Scalar &val, const cv::Size &inputImgSize)
- cv::Size `nnInputSizeRequired` () const
- virtual void `setFaceDetectionSize` (`FaceScanSettings::FaceDetectionSize` faceSize)

## Additional Inherited Members

### Static Public Attributes inherited from [Digikam::DNNFaceDetectorBase](#)

- static float **nmsThreshold** = 0.4F  
*Threshold for nms suppression.*
- static int **uiConfidenceThreshold** = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Threshold for bbox detection. It can be init and changed in the GUI.*

### Protected Member Functions inherited from [Digikam::DNNFaceDetectorBase](#)

- void **correctBbox** (cv::Rect &bbox, const cv::Size &paddedSize) const
- void **selectBbox** (const cv::Size &paddedSize, float confidence, int left, int right, int top, int bottom, std::vector< float > &goodConfidences, std::vector< cv::Rect > &goodBoxes, std::vector< float > &doubtConfidences, std::vector< cv::Rect > &doubtBoxes) const

### Protected Attributes inherited from [Digikam::DNNFaceDetectorBase](#)

- cv::Size **inputImageSize** = cv::Size(300, 300)
- cv::Scalar **meanValToSubtract** = cv::Scalar(0.0, 0.0, 0.0)
- [DNNModelBase](#) \* **model** = nullptr
- float **scaleFactor** = 1.0F

## 6.445.1 Member Function Documentation

### 6.445.1.1 detectFaces()

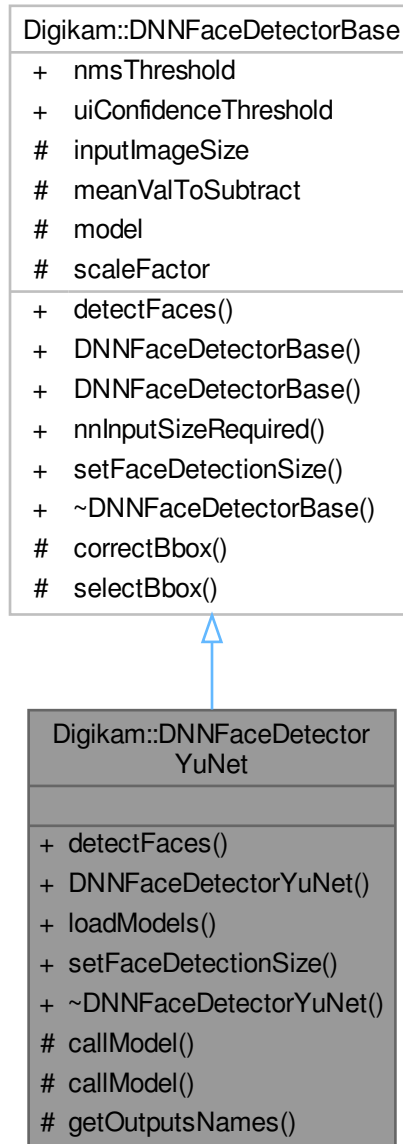
```
void Digikam::DNNFaceDetectorYOLO::detectFaces (
    const cv::Mat & inputImage,
    const cv::Size & paddedSize,
    std::vector< cv::Rect > & detectedBboxes ) [override], [virtual]
```

Implements [Digikam::DNNFaceDetectorBase](#).



## 6.446 Digikam::DNNFaceDetectorYuNet Class Reference

Inheritance diagram for Digikam::DNNFaceDetectorYuNet:



### Public Member Functions

- void [detectFaces](#) (const cv::Mat &inputImage, const cv::Size &paddedSize, std::vector< cv::Rect > &detectedBboxes) override
- bool **loadModels** ()
- virtual void [setFaceDetectionSize](#) ([FaceScanSettings::FaceDetectionSize](#) faceSize) override

## Public Member Functions inherited from [Digikam::DNNFaceDetectorBase](#)

- **DNNFaceDetectorBase** (float scale, const cv::Scalar &val, const cv::Size &inputImgSize)
- cv::Size **nnInputSizeRequired** () const

## Protected Member Functions

- cv::Mat **callModel** (const cv::Mat &inputImage)
- cv::UMat **callModel** (const cv::UMat &inputImage)
- std::vector< cv::String > **getOutputsNames** () const

## Protected Member Functions inherited from [Digikam::DNNFaceDetectorBase](#)

- void **correctBbox** (cv::Rect &bbox, const cv::Size &paddedSize) const
- void **selectBbox** (const cv::Size &paddedSize, float confidence, int left, int right, int top, int bottom, std::vector< float > &goodConfidences, std::vector< cv::Rect > &goodBoxes, std::vector< float > &doubtConfidences, std::vector< cv::Rect > &doubtBoxes) const

## Friends

- class **FacePipelineDetectRecognize**

## Additional Inherited Members

## Static Public Attributes inherited from [Digikam::DNNFaceDetectorBase](#)

- static float **nmsThreshold** = 0.4F  
*Threshold for nms suppression.*
- static int **uiConfidenceThreshold** = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Threshold for bbox detection. It can be init and changed in the GUI.*

## Protected Attributes inherited from [Digikam::DNNFaceDetectorBase](#)

- cv::Size **inputImageSize** = cv::Size(300, 300)
- cv::Scalar **meanValToSubtract** = cv::Scalar(0.0, 0.0, 0.0)
- [DNNModelBase](#) \* **model** = nullptr
- float **scaleFactor** = 1.0F

## 6.446.1 Member Function Documentation

### 6.446.1.1 detectFaces()

```
void Digikam::DNNFaceDetectorYuNet::detectFaces (
    const cv::Mat & inputImage,
    const cv::Size & paddedSize,
    std::vector< cv::Rect > & detectedBboxes ) [override], [virtual]
```

Implements [Digikam::DNNFaceDetectorBase](#).

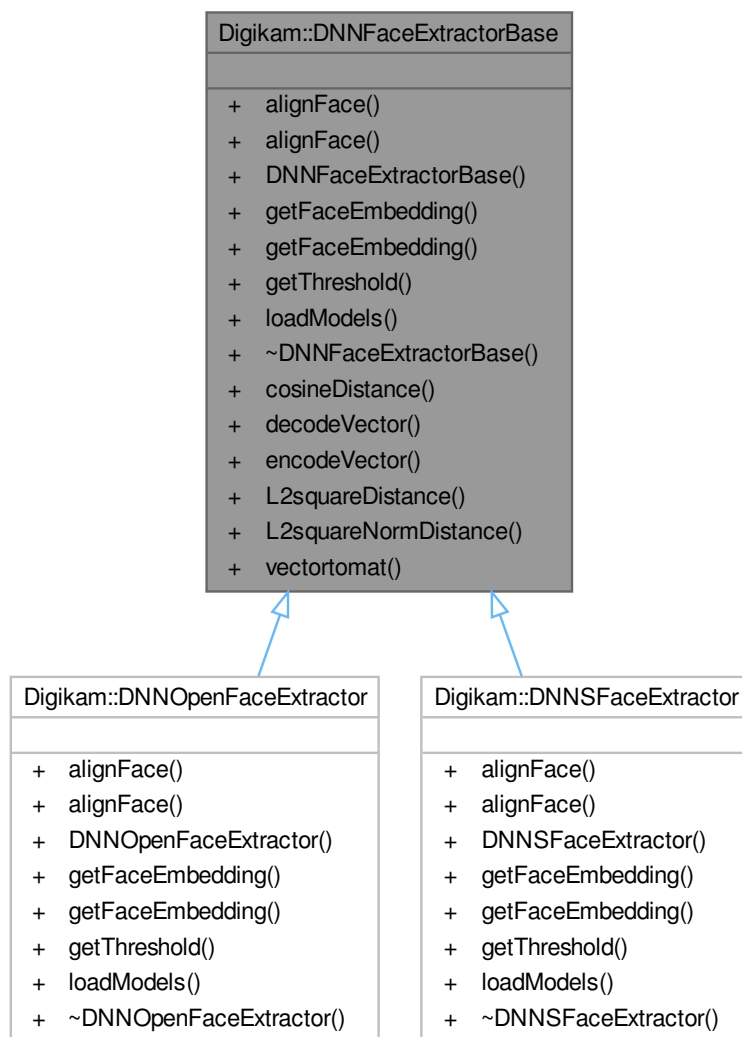
## 6.446.1.2 setFaceDetectionSize()

```
void Digikam::DNNFaceDetectorYuNet::setFaceDetectionSize (
    FaceScanSettings::FaceDetectionSize faceSize ) [override], [virtual]
```

Reimplemented from [Digikam::DNNFaceDetectorBase](#).

## 6.447 Digikam::DNNFaceExtractorBase Class Reference

Inheritance diagram for Digikam::DNNFaceExtractorBase:



## Public Member Functions

- virtual cv::Mat **alignFace** (const cv::Mat &inputImage) const =0
- virtual cv::UMat **alignFace** (const cv::UMat &inputImage) const =0
- virtual cv::Mat **getFaceEmbedding** (const cv::Mat &faceImage)=0
- virtual cv::Mat **getFaceEmbedding** (const cv::UMat &faceImage)=0
- virtual float **getThreshold** (int uiThreshold=DNN\_MODEL\_THRESHOLD\_NOT\_SET) const =0
- virtual bool **loadModels** ()=0

## Static Public Member Functions

- static double **cosineDistance** (const std::vector< float > &v1, const std::vector< float > &v2)
- static std::vector< float > **decodeVector** (const QJsonArray &json)
- static QJsonArray **encodeVector** (const std::vector< float > &vector)
- static double **L2squareDistance** (const std::vector< float > &v1, const std::vector< float > &v2)
- static double **L2squareNormDistance** (const std::vector< float > &v1, const std::vector< float > &v2)
- static cv::Mat **vectortomat** (const std::vector< float > &vector)

## 6.447.1 Member Function Documentation

### 6.447.1.1 cosineDistance()

```
double Digikam::DNNFaceExtractorBase::cosineDistance (
    const std::vector< float > & v1,
    const std::vector< float > & v2 ) [static]
```

Calculate different between 2 vectors.

### 6.447.1.2 getThreshold()

```
virtual float Digikam::DNNFaceExtractorBase::getThreshold (
    int uiThreshold = DNN_MODEL_THRESHOLD_NOT_SET ) const [pure virtual]
```

cover the UI threshold to a float using the conversion factor built into the model

Implemented in [Digikam::DNNOpenFaceExtractor](#), and [Digikam::DNNSFaceExtractor](#).

### 6.447.1.3 loadModels()

```
virtual bool Digikam::DNNFaceExtractorBase::loadModels ( ) [pure virtual]
```

Read pretrained neural network for face recognition.

Implemented in [Digikam::DNNOpenFaceExtractor](#), and [Digikam::DNNSFaceExtractor](#).

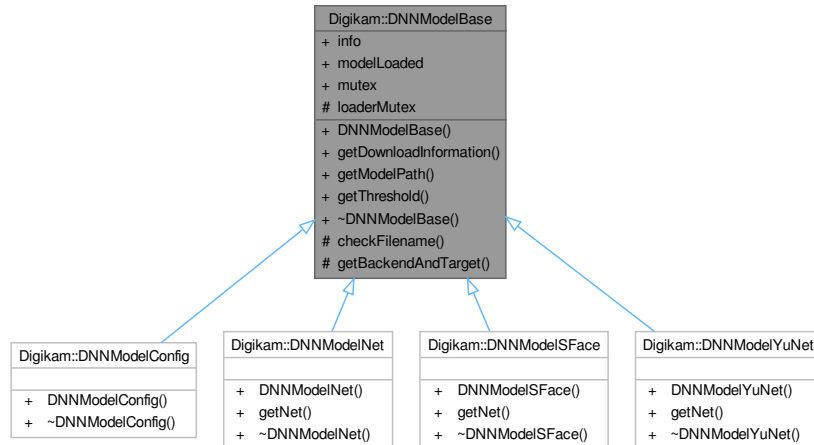
### 6.447.1.4 vectortomat()

```
cv::Mat Digikam::DNNFaceExtractorBase::vectortomat (
    const std::vector< float > & vector ) [static]
```

Convert face embedding between different formats.

## 6.448 Digikam::DNNModelBase Class Reference

Inheritance diagram for Digikam::DNNModelBase:



### Public Member Functions

- **DNNModelBase** (const [DNNModelInfoContainer](#) &\_info)
- **DownloadInfo** **getDownloadInformation** () const
- const QString **getModelPath** () const
- float **getThreshold** (int uiThreshold=DNN\_MODEL\_THRESHOLD\_NOT\_SET) const

### Public Attributes

- const [DNNModelInfoContainer](#) **info**  
*information about the model.*
- bool **modelLoaded** = false  
*check if the model has been loaded.*
- QMutex **mutex**  
*mutex to sigle-thread model during critical processing functions.*

### Protected Member Functions

- bool **checkFilename** () const
- const QPair< int, int > **getBackendAndTarget** () const

### Protected Attributes

- QMutex **loaderMutex**

## 6.448.1 Member Function Documentation

### 6.448.1.1 getModelPath()

```
const QString Digikam::DNNModelBase::getModelPath ( ) const
```

Return path to the model, or null string if path cannot be found.

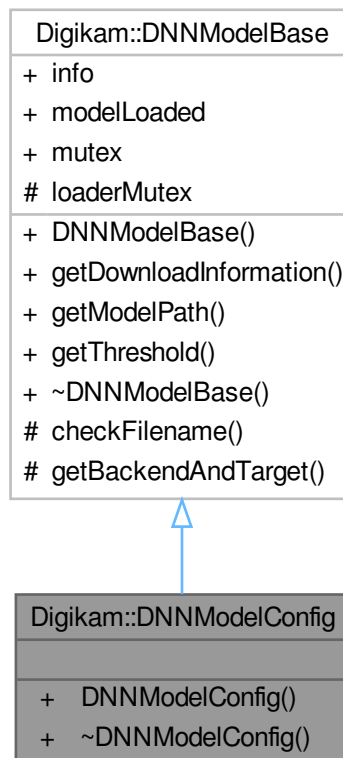
### 6.448.1.2 getThreshold()

```
float Digikam::DNNModelBase::getThreshold (
    int uiThreshold = DNN_MODEL_THRESHOLD_NOT_SET ) const
```

input: uiThreshold is the slider value from the UI. return: float threshold to be used by processing ([FaceDetector](#), [FaceRecognizer](#), etc..).

## 6.449 Digikam::DNNModelConfig Class Reference

Inheritance diagram for Digikam::DNNModelConfig:



**Public Member Functions**

- **DNNModelConfig** (const [DNNModelInfoContainer](#) &\_info)

**Public Member Functions inherited from [Digikam::DNNModelBase](#)**

- **DNNModelBase** (const [DNNModelInfoContainer](#) &\_info)
- **DownloadInfo** **getDownloadInformation** () const
- const QString **getModelPath** () const
- float **getThreshold** (int uiThreshold=DNN\_MODEL\_THRESHOLD\_NOT\_SET) const

**Additional Inherited Members****Public Attributes inherited from [Digikam::DNNModelBase](#)**

- const [DNNModelInfoContainer](#) **info**  
*information about the model.*
- bool **modelLoaded** = false  
*check if the model has been loaded.*
- QMutex **mutex**  
*mutex to sigle-thread model during critical processing functions.*

**Protected Member Functions inherited from [Digikam::DNNModelBase](#)**

- bool **checkFilename** () const
- const QPair< int, int > **getBackendAndTarget** () const

**Protected Attributes inherited from [Digikam::DNNModelBase](#)**

- QMutex **loaderMutex**

**6.450 Digikam::DNNModelInfoContainer Class Reference****Public Member Functions**

- **DNNModelInfoContainer** (const [DNNModelInfoContainer](#) &)
- **DNNModelInfoContainer** (const QString &\_displayName, const QString &\_fileName, const DNNModelUsageList &\_usage, const QVersionNumber &\_minVersion, const QString &\_downloadPath, const QString &\_sha256, const qint64 &\_fileSize, int \_defaultThreshold, int \_minUsableThreshold, int \_maxUsableThreshold, DNNLoaderType \_loaderType, const QString &\_classList, const QString &\_configName, const cv::Scalar &\_meanValToSubtract, int \_imageSize)
- **DNNModelInfoContainer** & **operator=** (const [DNNModelInfoContainer](#) &)
- **DNNModelInfoContainer** & **operator=** ([DNNModelInfoContainer](#) &&)
- bool **operator==** (const [DNNModelInfoContainer](#) &t) const

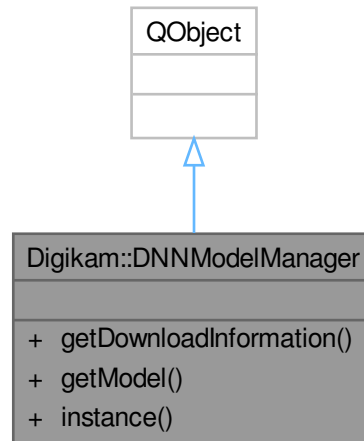
## Public Attributes

- QString **classList**  
*Name of model containing list of class names for classification.*
- QString **configName**
- int **defaultThreshold** = 0  
*Threshold used for models that aren't configured by the UI.*
- QString **displayName**  
*Name used for display in UI (QComboBox).*
- QString **downloadPath**  
*Used by the downloader for the download path.*
- QString **fileName**  
*Used by the downloader and model loader.*
- qint64 **fileSize** = 0  
*Used by the downloader to verify size.*
- int **imageSize** = 0  
*Max dimension of a side of an image.*
- DNNLoaderType **loaderType** = DNNLoaderNet  
*Model loader type custom (YuNet/SFace), Caffe, Darknet, Torch, Tensorflow.*
- int **maxUsableThreshold** = 0  
*Used to convert UI 1-10 slider to float for processing.*
- cv::Scalar **meanValToSubtract** = cv::Scalar(0.0, 0.0, 0.0)
- int **minUsableThreshold** = 0  
*Used to convert UI 1-10 slider to float for processing.*
- QVersionNumber **minVersion**  
*Minimum version of digiKam needed to use this model.*
- QString **sha256**  
*SHA265 hash of the file for download.*
- DNNModelUsageList **usage**  
*How the model can be used. | for more than one use. face\_detection, face\_recognition, weight, object\_detection, etc...*



## 6.451 Digikam::DNNModelManager Class Reference

Inheritance diagram for Digikam::DNNModelManager:



### Public Member Functions

- `const QList< DownloadInfo > & getDownloadInformation (DNNModelUsage usage)`
- `DNNModelBase * getModel (const QString &modelName, DNNModelUsage usage) const`

### Static Public Member Functions

- `static DNNModelManager * instance ()`

### Friends

- class `DNNModelManagerCreator`

## 6.451.1 Member Function Documentation

### 6.451.1.1 getDownloadInformation()

```
const QList< DownloadInfo > & Digikam::DNNModelManager::getDownloadInformation (
    DNNModelUsage usage )
```

Used by the filesdownload to get a stream containing the files and information to download.

### 6.451.1.2 getModel()

```
DNNModelBase * Digikam::DNNModelManager::getModel (
    const QString & modelName,
    DNNModelUsage usage ) const
```

Retrieve a [DNNModelBase](#) pointer by name. This will load and create the model on first use. It will also find the best OpenCV Target and Backend for the model based on computer capabilities. Returns nullptr if 'modelName' cannot be found.

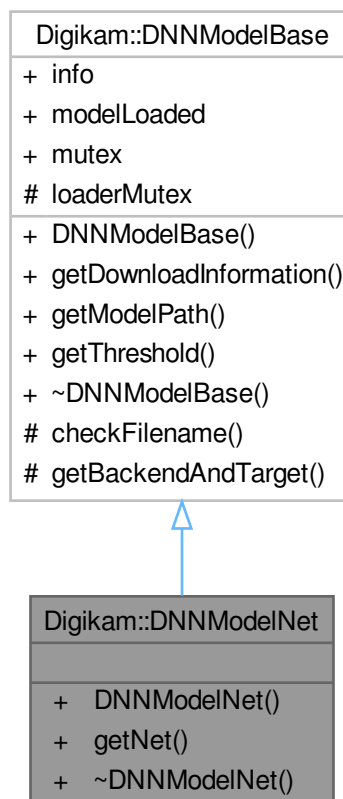
### 6.451.1.3 instance()

```
DNNModelManager * Digikam::DNNModelManager::instance ( ) [static]
```

Global instance of internal model manager. All accessor methods are thread-safe.

## 6.452 Digikam::DNNModelNet Class Reference

Inheritance diagram for Digikam::DNNModelNet:



### Public Member Functions

- **DNNModelNet** (const [DNNModelInfoContainer](#) &\_info)
- cv::dnn::Net & **getNet** ()

### Public Member Functions inherited from [Digikam::DNNModelBase](#)

- **DNNModelBase** (const [DNNModelInfoContainer](#) &\_info)
- [DownloadInfo](#) **getDownloadInformation** () const
- const QString **getModelPath** () const
- float **getThreshold** (int uiThreshold=DNN\_MODEL\_THRESHOLD\_NOT\_SET) const

### Additional Inherited Members

### Public Attributes inherited from [Digikam::DNNModelBase](#)

- const [DNNModelInfoContainer](#) **info**  
*information about the model.*
- bool **modelLoaded** = false  
*check if the model has been loaded.*
- QMutex **mutex**  
*mutex to sigle-thread model during critical processing functions.*

### Protected Member Functions inherited from [Digikam::DNNModelBase](#)

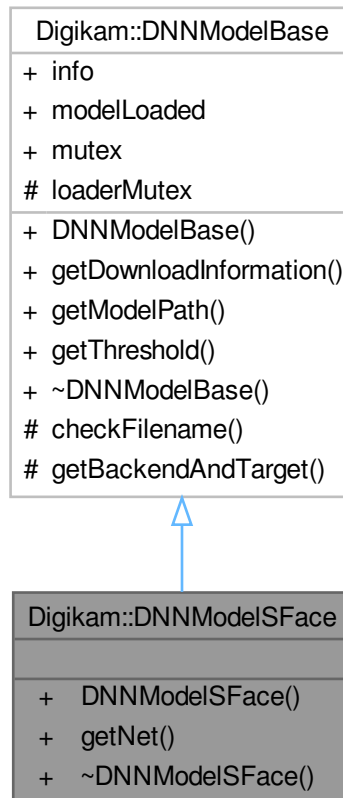
- bool **checkFilename** () const
- const QPair< int, int > **getBackendAndTarget** () const

### Protected Attributes inherited from [Digikam::DNNModelBase](#)

- QMutex **loaderMutex**

## 6.453 Digikam::DNNModelSFace Class Reference

Inheritance diagram for Digikam::DNNModelSFace:



### Public Member Functions

- **DNNModelSFace** (const [DNNModelInfoContainer](#) &\_info)
- `cv::Ptr< cv::FaceRecognizerSF >` & **getNet** ()

### Public Member Functions inherited from [Digikam::DNNModelBase](#)

- **DNNModelBase** (const [DNNModelInfoContainer](#) &\_info)
- [DownloadInfo](#) **getDownloadInformation** () const
- const QString **getModelPath** () const
- float **getThreshold** (int uiThreshold=DNN\_MODEL\_THRESHOLD\_NOT\_SET) const

**Additional Inherited Members****Public Attributes inherited from [Digikam::DNNModelBase](#)**

- const [DNNModelInfoContainer](#) **info**  
*information about the model.*
- bool **modelLoaded** = false  
*check if the model has been loaded.*
- QMutex **mutex**  
*mutex to sigle-thread model during critical processing functions.*

**Protected Member Functions inherited from [Digikam::DNNModelBase](#)**

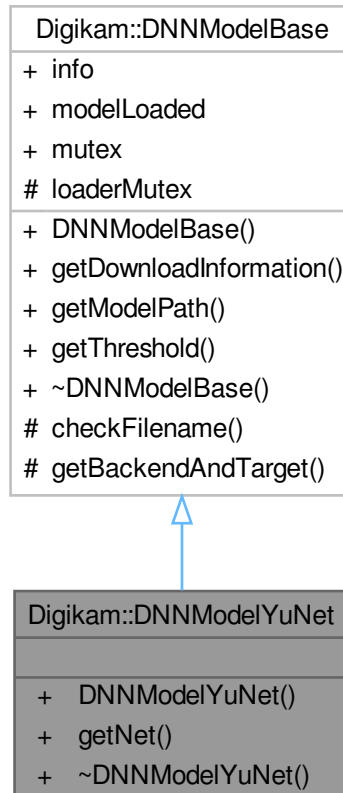
- bool **checkFilename** () const
- const QPair< int, int > **getBackendAndTarget** () const

**Protected Attributes inherited from [Digikam::DNNModelBase](#)**

- QMutex **loaderMutex**

**6.454 Digikam::DNNModelYuNet Class Reference**

Inheritance diagram for Digikam::DNNModelYuNet:



### Public Member Functions

- **DNNModelYuNet** (const [DNNModelInfoContainer](#) &\_info)
- cv::Ptr< cv::FaceDetectorYN > & **getNet** ()

### Public Member Functions inherited from [Digikam::DNNModelBase](#)

- **DNNModelBase** (const [DNNModelInfoContainer](#) &\_info)
- [DownloadInfo](#) **getDownloadInformation** () const
- const QString **getModelPath** () const
- float **getThreshold** (int uiThreshold=DNN\_MODEL\_THRESHOLD\_NOT\_SET) const

### Additional Inherited Members

### Public Attributes inherited from [Digikam::DNNModelBase](#)

- const [DNNModelInfoContainer](#) **info**  
*information about the model.*
- bool **modelLoaded** = false  
*check if the model has been loaded.*
- QMutex **mutex**  
*mutex to sigle-thread model during critical processing functions.*

### Protected Member Functions inherited from [Digikam::DNNModelBase](#)

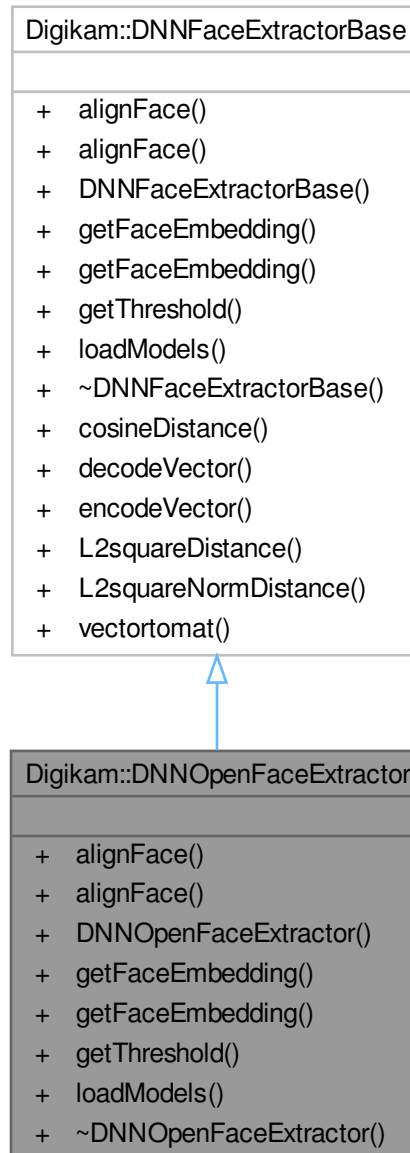
- bool **checkFilename** () const
- const QPair< int, int > **getBackendAndTarget** () const

### Protected Attributes inherited from [Digikam::DNNModelBase](#)

- QMutex **loaderMutex**

## 6.455 Digikam::DNNOpenFaceExtractor Class Reference

Inheritance diagram for Digikam::DNNOpenFaceExtractor:



### Public Member Functions

- virtual `cv::Mat alignFace` (`const cv::Mat &inputImage`) `const` override
- virtual `cv::UMat alignFace` (`const cv::UMat &inputImage`) `const` override
- virtual `cv::Mat getFaceEmbedding` (`const cv::Mat &faceImage`) `override`
- virtual `cv::Mat getFaceEmbedding` (`const cv::UMat &faceImage`) `override`
- float `getThreshold` (`int uiThreshold=DNN_MODEL_THRESHOLD_NOT_SET`) `const` override
- bool `loadModels` (`()`) `override`

## Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::DNNFaceExtractorBase](#)

- static double [cosineDistance](#) (const std::vector< float > &v1, const std::vector< float > &v2)
- static std::vector< float > [decodeVector](#) (const QJsonArray &json)
- static QJsonArray [encodeVector](#) (const std::vector< float > &vector)
- static double [L2squareDistance](#) (const std::vector< float > &v1, const std::vector< float > &v2)
- static double [L2squareNormDistance](#) (const std::vector< float > &v1, const std::vector< float > &v2)
- static cv::Mat [vectortomat](#) (const std::vector< float > &vector)

## 6.455.1 Member Function Documentation

### 6.455.1.1 [alignFace\(\)](#) [1/2]

```
cv::Mat Digikam::DNNOpenFaceExtractor::alignFace (
    const cv::Mat & inputImage ) const [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.455.1.2 [alignFace\(\)](#) [2/2]

```
virtual cv::UMat Digikam::DNNOpenFaceExtractor::alignFace (
    const cv::UMat & inputImage ) const [inline], [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.455.1.3 [getFaceEmbedding\(\)](#) [1/2]

```
cv::Mat Digikam::DNNOpenFaceExtractor::getFaceEmbedding (
    const cv::Mat & faceImage ) [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.455.1.4 [getFaceEmbedding\(\)](#) [2/2]

```
virtual cv::Mat Digikam::DNNOpenFaceExtractor::getFaceEmbedding (
    const cv::UMat & faceImage ) [inline], [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.455.1.5 [getThreshold\(\)](#)

```
float Digikam::DNNOpenFaceExtractor::getThreshold (
    int uiThreshold = DNN_MODEL_THRESHOLD_NOT_SET ) const [override], [virtual]
```

cover the UI threshold to a float using the conversion factor built into the model

Implements [Digikam::DNNFaceExtractorBase](#).



## 6.455.1.6 loadModels()

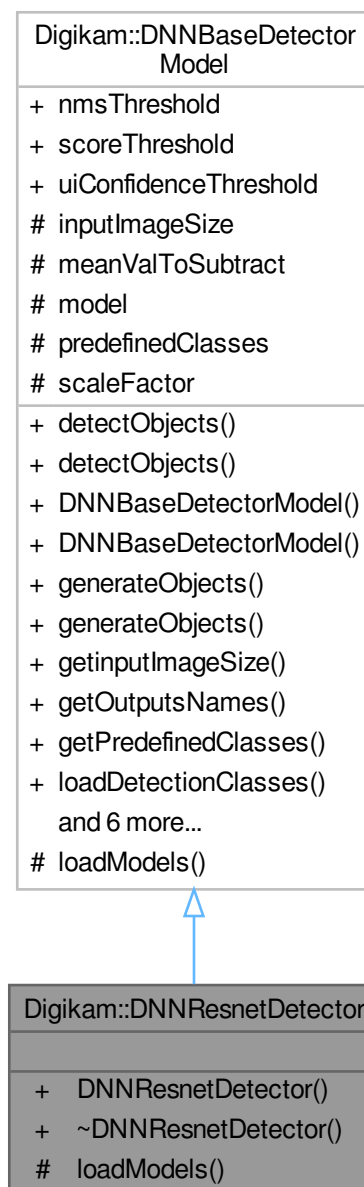
```
bool Digikam::DNNOpenFaceExtractor::loadModels ( ) [override], [virtual]
```

Read pretrained neural network for face recognition.

Implements [Digikam::DNNFaceExtractorBase](#).

## 6.456 Digikam::DNNResnetDetector Class Reference

Inheritance diagram for Digikam::DNNResnetDetector:



## Protected Member Functions

- bool [loadModels](#) () override

## Additional Inherited Members

### Public Member Functions inherited from [Digikam::DNNBaseDetectorModel](#)

- virtual QHash< QString, QVector< QRect > > [detectObjects](#) (const cv::Mat &inputImage)
- virtual QList< QHash< QString, QVector< QRect > > > [detectObjects](#) (const std::vector< cv::Mat > &inputBatchImages)
- **DNNBaseDetectorModel** (float scale, const cv::Scalar &val, const cv::Size &inputImgSize)
- QList< QString > [generateObjects](#) (const cv::Mat &inputImage)
- QList< QList< QString > > [generateObjects](#) (const std::vector< cv::Mat > &inputImage)
- cv::Size [getInputImageSize](#) () const
- std::vector< cv::String > [getOutputsNames](#) () const
- virtual QList< QString > [getPredefinedClasses](#) () const
- QList< QString > [loadDetectionClasses](#) ()
- QList< QHash< QString, QVector< QRect > > > [postprocess](#) (const std::vector< cv::Mat > &inputBatchImages, const std::vector< cv::Mat > &outs) const
- std::vector< cv::Mat > [preprocess](#) (const cv::Mat &inputImage)
- std::vector< cv::Mat > [preprocess](#) (const std::vector< cv::Mat > &inputBatchImages)
- double [showInferenceTime](#) ()

### Static Public Attributes inherited from [Digikam::DNNBaseDetectorModel](#)

- static float **nmsThreshold** = 0.4F  
*Threshold for nms suppression.*
- static float **scoreThreshold** = 0.45F  
*Threshold for class detection score.*
- static int **uiConfidenceThreshold** = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Threshold for bbox detection. It can be init and changed in the GUI.*

### Protected Attributes inherited from [Digikam::DNNBaseDetectorModel](#)

- cv::Size **inputImageSize**
- cv::Scalar **meanValToSubtract**
- [DNNModelBase](#) \* **model** = nullptr
- QList< QString > **predefinedClasses**
- float **scaleFactor** = 1.0F

## 6.456.1 Member Function Documentation

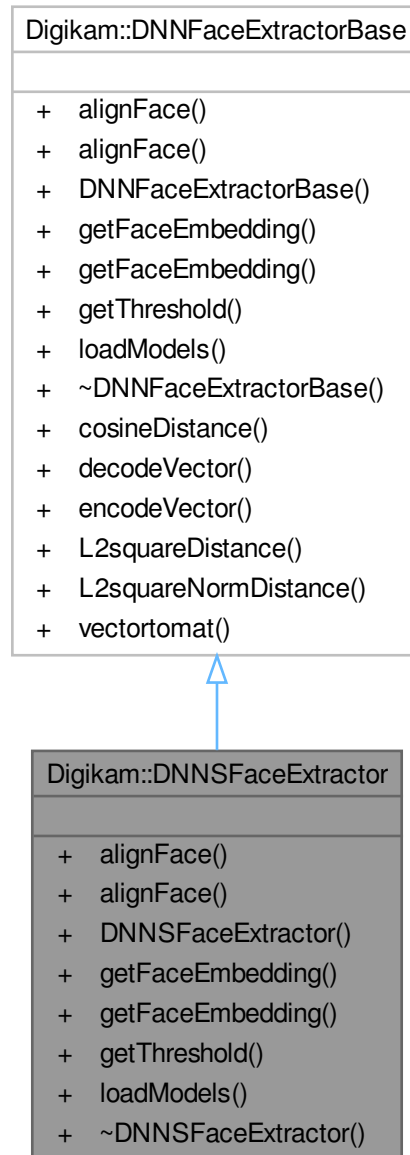
### 6.456.1.1 loadModels()

```
bool Digikam::DNNResnetDetector::loadModels ( ) [override], [protected], [virtual]
```

Implements [Digikam::DNNBaseDetectorModel](#).

## 6.457 Digikam::DNNSFaceExtractor Class Reference

Inheritance diagram for Digikam::DNNSFaceExtractor:



### Public Member Functions

- virtual `cv::Mat alignFace` (`const cv::Mat &inputImage`) `const` override
- virtual `cv::UMat alignFace` (`const cv::UMat &inputImage`) `const` override
- virtual `cv::Mat getFaceEmbedding` (`const cv::Mat &facelImage`) `override`
- virtual `cv::Mat getFaceEmbedding` (`const cv::UMat &facelImage`) `override`
- float `getThreshold` (`int uiThreshold=DNN_MODEL_THRESHOLD_NOT_SET`) `const` override
- bool `loadModels` () `override`

## Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::DNNFaceExtractorBase](#)

- static double [cosineDistance](#) (const std::vector< float > &v1, const std::vector< float > &v2)
- static std::vector< float > [decodeVector](#) (const QJsonArray &json)
- static QJsonArray [encodeVector](#) (const std::vector< float > &vector)
- static double [L2squareDistance](#) (const std::vector< float > &v1, const std::vector< float > &v2)
- static double [L2squareNormDistance](#) (const std::vector< float > &v1, const std::vector< float > &v2)
- static cv::Mat [vectortomat](#) (const std::vector< float > &vector)

## 6.457.1 Member Function Documentation

### 6.457.1.1 [alignFace\(\)](#) [1/2]

```
cv::Mat Digikam::DNNSFaceExtractor::alignFace (
    const cv::Mat & inputImage ) const [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.457.1.2 [alignFace\(\)](#) [2/2]

```
cv::UMat Digikam::DNNSFaceExtractor::alignFace (
    const cv::UMat & inputImage ) const [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.457.1.3 [getFaceEmbedding\(\)](#) [1/2]

```
cv::Mat Digikam::DNNSFaceExtractor::getFaceEmbedding (
    const cv::Mat & faceImage ) [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.457.1.4 [getFaceEmbedding\(\)](#) [2/2]

```
cv::Mat Digikam::DNNSFaceExtractor::getFaceEmbedding (
    const cv::UMat & faceImage ) [override], [virtual]
```

Implements [Digikam::DNNFaceExtractorBase](#).

### 6.457.1.5 [getThreshold\(\)](#)

```
float Digikam::DNNSFaceExtractor::getThreshold (
    int uiThreshold = DNN_MODEL_THRESHOLD_NOT_SET ) const [override], [virtual]
```

cover the UI threshold to a float using the conversion factor built into the model

Implements [Digikam::DNNFaceExtractorBase](#).

## 6.457.1.6 loadModels()

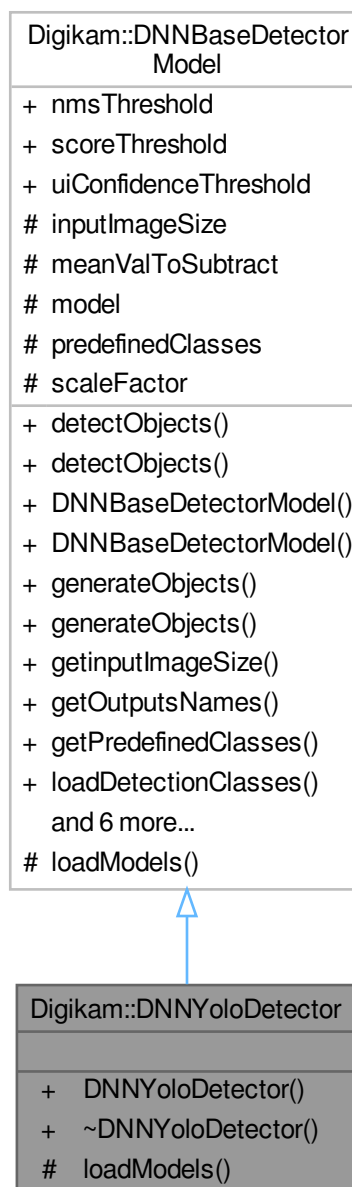
```
bool Digikam::DNNFaceExtractor::loadModels ( ) [override], [virtual]
```

Read pretrained neural network for face recognition.

Implements [Digikam::DNNFaceExtractorBase](#).

## 6.458 Digikam::DNNYoloDetector Class Reference

Inheritance diagram for Digikam::DNNYoloDetector:



## Public Member Functions

- **DNNYoloDetector** ([YoloVersions](#) modelVersion=[YoloVersions::YOLOV5NANO](#))

## Public Member Functions inherited from [Digikam::DNNBaseDetectorModel](#)

- virtual [QHash](#)< [QString](#), [QVector](#)< [QRect](#) > > [detectObjects](#) (const cv::Mat &inputImage)
- virtual [QList](#)< [QHash](#)< [QString](#), [QVector](#)< [QRect](#) > > > [detectObjects](#) (const std::vector< cv::Mat > &inputBatchImages)
- **DNNBaseDetectorModel** (float scale, const cv::Scalar &val, const cv::Size &inputImgSize)
- [QList](#)< [QString](#) > [generateObjects](#) (const cv::Mat &inputImage)
- [QList](#)< [QList](#)< [QString](#) > > [generateObjects](#) (const std::vector< cv::Mat > &inputImage)
- cv::Size [getInputImageSize](#) () const
- std::vector< cv::String > [getOutputsNames](#) () const
- virtual [QList](#)< [QString](#) > [getPredefinedClasses](#) () const
- [QList](#)< [QString](#) > [loadDetectionClasses](#) ()
- [QList](#)< [QHash](#)< [QString](#), [QVector](#)< [QRect](#) > > > [postprocess](#) (const std::vector< cv::Mat > &inputBatchImages, const std::vector< cv::Mat > &outs) const
- std::vector< cv::Mat > [preprocess](#) (const cv::Mat &inputImage)
- std::vector< cv::Mat > [preprocess](#) (const std::vector< cv::Mat > &inputBatchImages)
- double [showInferenceTime](#) ()

## Protected Member Functions

- bool [loadModels](#) () override

## Additional Inherited Members

## Static Public Attributes inherited from [Digikam::DNNBaseDetectorModel](#)

- static float **nmsThreshold** = 0.4F  
*Threshold for nms suppression.*
- static float **scoreThreshold** = 0.45F  
*Threshold for class detection score.*
- static int [uiConfidenceThreshold](#) = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Threshold for bbox detection. It can be init and changed in the GUI.*

## Protected Attributes inherited from [Digikam::DNNBaseDetectorModel](#)

- cv::Size **inputImageSize**
- cv::Scalar **meanValToSubtract**
- [DNNModelBase](#) \* **model** = nullptr
- [QList](#)< [QString](#) > **predefinedClasses**
- float **scaleFactor** = 1.0F

## 6.458.1 Member Function Documentation

### 6.458.1.1 loadModels()

```
bool Digikam::DNNYoloDetector::loadModels ( ) [override], [protected], [virtual]
```

Implements [Digikam::DNNBaseDetectorModel](#).

## 6.459 Digikam::DNotificationPopup Class Reference

A dialog-like popup that displays messages without interrupting the user.

Inheritance diagram for Digikam::DNotificationPopup:



### Public Types

- enum `PopupStyle` { `Boxed` , `Balloon` }

### Public Slots

- void [setPopupStyle](#) (int popupstyle)
- void [setTimeout](#) (int delay)
- void **setVisible** (bool visible) override
- void [show](#) (const QPoint &p)

### Signals

- void [clicked](#) ()
- void [clicked](#) (const QPoint &pos)

### Public Member Functions

- QPoint [anchor](#) () const
- bool [autoDelete](#) () const
- [DNotificationPopup](#) (QWidget \*const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())
- [DNotificationPopup](#) (Wld parent)
- void [setAnchor](#) (const QPoint &anchor)
- virtual void [setAutoDelete](#) (bool autoDelete)
- virtual void [setView](#) (const QString &caption, const QString &text, const QPixmap &icon)
- void [setView](#) (const QString &caption, const QString &text=QString())
- void [setView](#) (QWidget \*child)
- QWidget \* [standardView](#) (const QString &caption, const QString &text, const QPixmap &icon, QWidget \*parent=nullptr)
- int [timeout](#) () const
- QWidget \* [view](#) () const
- [~DNotificationPopup](#) () override

### Static Public Member Functions

- static [DNotificationPopup](#) \* [message](#) (const QString &caption, const QString &text, const QPixmap &icon, QSystemTrayIcon \*parent, int timeout=-1)
- static [DNotificationPopup](#) \* [message](#) (const QString &caption, const QString &text, const QPixmap &icon, QWidget \*parent, int timeout=-1, const QPoint &p=QPoint())
- static [DNotificationPopup](#) \* [message](#) (const QString &caption, const QString &text, const QPixmap &icon, Wld parent, int timeout=-1, const QPoint &p=QPoint())
- static [DNotificationPopup](#) \* [message](#) (const QString &caption, const QString &text, QSystemTrayIcon \*parent)
- static [DNotificationPopup](#) \* [message](#) (const QString &caption, const QString &text, QWidget \*parent, const QPoint &p=QPoint())
- static [DNotificationPopup](#) \* [message](#) (const QString &text, QSystemTrayIcon \*parent)
- static [DNotificationPopup](#) \* [message](#) (const QString &text, QWidget \*parent, const QPoint &p=QPoint())
- static [DNotificationPopup](#) \* [message](#) (int popupStyle, const QString &caption, const QString &text, const QPixmap &icon, QSystemTrayIcon \*parent, int timeout=-1)
- static [DNotificationPopup](#) \* [message](#) (int popupStyle, const QString &caption, const QString &text, const QPixmap &icon, QWidget \*parent, int timeout=-1, const QPoint &p=QPoint())
- static [DNotificationPopup](#) \* [message](#) (int popupStyle, const QString &caption, const QString &text, const QPixmap &icon, Wld parent, int timeout=-1, const QPoint &p=QPoint())
- static [DNotificationPopup](#) \* [message](#) (int popupStyle, const QString &caption, const QString &text, QSystemTrayIcon \*parent)
- static [DNotificationPopup](#) \* [message](#) (int popupStyle, const QString &caption, const QString &text, QWidget \*parent, const QPoint &p=QPoint())
- static [DNotificationPopup](#) \* [message](#) (int popupStyle, const QString &text, QSystemTrayIcon \*parent)
- static [DNotificationPopup](#) \* [message](#) (int popupStyle, const QString &text, QWidget \*parent, const QPoint &p=QPoint())



## Protected Member Functions

- virtual QPoint [defaultLocation](#) () const
- void [hideEvent](#) (QHideEvent \*) override
- void [mouseReleaseEvent](#) (QMouseEvent \*) override
- void [moveNear](#) (const QRect &target)
- void [paintEvent](#) (QPaintEvent \*pe) override
- virtual void [positionSelf](#) ()

## Properties

- bool [autoDelete](#)
- int [timeout](#)

### 6.459.1 Detailed Description

The simplest uses of [DNotificationPopup](#) are by using the various [message\(\)](#) static methods. The position the popup appears at depends on the type of the parent window:

### 6.459.2 Member Enumeration Documentation

#### 6.459.2.1 PopupStyle

```
enum Digikam::DNotificationPopup::PopupStyle
```

Styles that a [DNotificationPopup](#) can have.

Enumerator

Boxed	Information will appear in a framed box (default)
Balloon	Information will appear in a comic-alike balloon.

### 6.459.3 Constructor & Destructor Documentation

#### 6.459.3.1 DNotificationPopup() [1/2]

```
Digikam::DNotificationPopup::DNotificationPopup (
    QWidget *const parent = nullptr,
    Qt::WindowFlags f = Qt::WindowFlags() ) [explicit]
```

Creates a popup for the specified widget.

#### 6.459.3.2 DNotificationPopup() [2/2]

```
Digikam::DNotificationPopup::DNotificationPopup (
    WId parent ) [explicit]
```

Creates a popup for the specified window.

### 6.459.3.3 ~DNotificationPopup()

```
Digikam::DNotificationPopup::~DNotificationPopup ( ) [override]
```

Cleans up.

## 6.459.4 Member Function Documentation

### 6.459.4.1 anchor()

```
QPoint Digikam::DNotificationPopup::anchor ( ) const
```

Returns the position to which this popup is anchored.

### 6.459.4.2 autoDelete()

```
bool Digikam::DNotificationPopup::autoDelete ( ) const
```

Returns whether the popup will be deleted when it is hidden.

See also

[setAutoDelete](#)

### 6.459.4.3 clicked [1/2]

```
void Digikam::DNotificationPopup::clicked ( ) [signal]
```

Emitted when the popup is clicked.

### 6.459.4.4 clicked [2/2]

```
void Digikam::DNotificationPopup::clicked (
    const QPoint & pos ) [signal]
```

Emitted when the popup is clicked.

### 6.459.4.5 defaultLocation()

```
QPoint Digikam::DNotificationPopup::defaultLocation ( ) const [protected], [virtual]
```

Returns a default location for popups when a better placement cannot be found.

The default implementation returns the top-left corner of the available work area of the desktop (ie: minus panels, etc).

#### 6.459.4.6 message() [1/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    const QString & caption,
    const QString & text,
    const QPixmap & icon,
    QSystemTrayIcon * parent,
    int timeout = -1 ) [static]
```

Convenience method that displays popup with the specified icon, caption and message beside the icon of the specified QSystemTrayIcon. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.7 message() [2/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    const QString & caption,
    const QString & text,
    const QPixmap & icon,
    QWidget * parent,
    int timeout = -1,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified icon, caption and message beside the icon of the specified widget. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.8 message() [3/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    const QString & caption,
    const QString & text,
    const QPixmap & icon,
    WId parent,
    int timeout = -1,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified icon, caption and message beside the icon of the specified window. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.9 message() [4/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    const QString & caption,
    const QString & text,
    QSystemTrayIcon * parent ) [static]
```

Convenience method that displays popup with the specified caption and message beside the icon of the specified QSystemTrayIcon. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.10 message() [5/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    const QString & caption,
    const QString & text,
    QWidget * parent,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified caption and message beside the icon of the specified widget. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.11 message() [6/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    const QString & text,
    QSystemTrayIcon * parent ) [static]
```

Convenience method that displays popup with the specified message beside the icon of the specified QSystemTrayIcon. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.12 message() [7/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    const QString & text,
    QWidget * parent,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified message beside the icon of the specified widget. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

**6.459.4.13 message()** [8/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    int popupStyle,
    const QString & caption,
    const QString & text,
    const QPixmap & icon,
    QSystemTrayIcon * parent,
    int timeout = -1 ) [static]
```

Convenience method that displays popup with the specified popup-style, icon, caption and message beside the icon of the specified QSystemTrayIcon. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

**6.459.4.14 message()** [9/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    int popupStyle,
    const QString & caption,
    const QString & text,
    const QPixmap & icon,
    QWidget * parent,
    int timeout = -1,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified popup-style, icon, caption and message beside the icon of the specified widget. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

**6.459.4.15 message()** [10/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    int popupStyle,
    const QString & caption,
    const QString & text,
    const QPixmap & icon,
    WId parent,
    int timeout = -1,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified popup-style, icon, caption and message beside the icon of the specified window. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

**6.459.4.16 message()** [11/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    int popupStyle,
    const QString & caption,
    const QString & text,
    QSystemTrayIcon * parent ) [static]
```

Convenience method that displays popup with the specified popup-style, caption and message beside the icon of the specified QSystemTrayIcon. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

**6.459.4.17 message()** [12/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    int popupStyle,
    const QString & caption,
    const QString & text,
    QWidget * parent,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified popup-style, caption and message beside the icon of the specified widget. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

**6.459.4.18 message()** [13/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    int popupStyle,
    const QString & text,
    QSystemTrayIcon * parent ) [static]
```

Convenience method that displays popup with the specified popup-style and message beside the icon of the specified QSystemTrayIcon. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.19 message() [14/14]

```
DNotificationPopup * Digikam::DNotificationPopup::message (
    int popupStyle,
    const QString & text,
    QWidget * parent,
    const QPoint & p = QPoint() ) [static]
```

Convenience method that displays popup with the specified popup-style and message beside the icon of the specified widget. Note that the returned object is destroyed when it is hidden.

See also

[setAutoDelete](#)

#### 6.459.4.20 moveNear()

```
void Digikam::DNotificationPopup::moveNear (
    const QRect & target ) [protected]
```

Moves the popup to be adjacent to `target`.

The popup will be placed adjacent to, but outside of, `target`, without going off the current desktop.

Reimplementations of [positionSelf\(\)](#) can use this to actually position the popup.

#### 6.459.4.21 positionSelf()

```
void Digikam::DNotificationPopup::positionSelf ( ) [protected], [virtual]
```

Positions the popup.

The default implementation attempts to place it by the taskbar entry; failing that it places it by the window of the associated widget; failing that it places it at the location given by [defaultLocation\(\)](#).

See also

[moveNear\(\)](#)

#### 6.459.4.22 setAnchor()

```
void Digikam::DNotificationPopup::setAnchor (
    const QPoint & anchor )
```

Sets the anchor of this popup.

The popup is placed near to the anchor.

#### 6.459.4.23 `setAutoDelete()`

```
void Digikam::DNotificationPopup::setAutoDelete (
    bool autoDelete ) [virtual]
```

Sets whether the popup will be deleted when it is hidden.

The default is false (unless created by one of the static [message\(\)](#) overloads).

#### 6.459.4.24 `setPopupStyle`

```
void Digikam::DNotificationPopup::setPopupStyle (
    int popupstyle ) [slot]
```

Sets the visual appearance of the popup.

See also

[PopupStyle](#)

#### 6.459.4.25 `setTimeout`

```
void Digikam::DNotificationPopup::setTimeout (
    int delay ) [slot]
```

Sets the delay for the popup is removed automatically. Setting the delay to 0 disables the timeout, if you're doing this, you may want to connect the [clicked\(\)](#) signal to the `hide()` slot. Setting the delay to -1 makes it use the default value.

See also

`timeout`

#### 6.459.4.26 `setView()` [1/3]

```
void Digikam::DNotificationPopup::setView (
    const QString & caption,
    const QString & text,
    const QPixmap & icon ) [virtual]
```

Creates a standard view then calls [setView\(QWidget\\*\)](#) .

#### 6.459.4.27 `setView()` [2/3]

```
void Digikam::DNotificationPopup::setView (
    const QString & caption,
    const QString & text = QString() )
```

Creates a standard view then calls [setView\(QWidget\\*\)](#) .



**6.459.4.28 setView()** [3/3]

```
void Digikam::DNotificationPopup::setView (
    QWidget * child )
```

Sets the main view to be the specified widget (which must be a child of the popup).

**6.459.4.29 show**

```
void Digikam::DNotificationPopup::show (
    const QPoint & p ) [slot]
```

Shows the popup in the given point

**6.459.4.30 standardView()**

```
QWidget * Digikam::DNotificationPopup::standardView (
    const QString & caption,
    const QString & text,
    const QPixmap & icon,
    QWidget * parent = nullptr )
```

Returns a widget that is used as standard view if one of the [setView\(\)](#) methods taking the QString arguments is used. You can use the returned widget to customize the passivepopup while keeping the look similar to the "standard" passivepopups.

After customizing the widget, pass it to [setView\( QWidget\\* \)](#)

**Parameters**

<i>caption</i>	The window caption (title) on the popup
<i>text</i>	The text for the popup
<i>icon</i>	The icon to use for the popup
<i>parent</i>	The parent widget used for the returned widget. If left 0, then "this", i.e. the passive popup object will be used.

**Returns**

a QWidget containing the given arguments, looking like the standard passivepopups. The returned widget contains a QVBoxLayout, which is accessible through layout().

**See also**

[setView\( QWidget \\* \)](#)  
[setView\( const QString&, const QString& \)](#)  
[setView\( const QString&, const QString&, const QPixmap& \)](#)

**6.459.4.31 timeout()**

```
int Digikam::DNotificationPopup::timeout ( ) const
```

Returns the delay before the popup is removed automatically.

**6.459.4.32 view()**

```
QWidget * Digikam::DNotificationPopup::view ( ) const
```

Returns the main view.

**6.460 Digikam::DNotificationWidget Class Reference**

Inheritance diagram for Digikam::DNotificationWidget:



## Classes

- class [Private](#)

## Public Types

- enum [MessageType](#) {  
    **Positive** , **Notification** , **Information** , **Warning** ,  
    **Error** }

## Public Slots

- void [animatedHide](#) ()
- void [animatedShow](#) ()
- void [setCloseButtonVisible](#) (bool visible)
- void [setIcon](#) (const QIcon &icon)
- void [setMessageType](#) (DNotificationWidget::MessageType type)
- void [setText](#) (const QString &text)
- void [setWordWrap](#) (bool wordWrap)

## Signals

- void [hideAnimationFinished](#) ()
- void [linkActivated](#) (const QString &contents)
- void [linkHovered](#) (const QString &contents)
- void [showAnimationFinished](#) ()

## Public Member Functions

- void [addAction](#) (QAction \*action)
- void [animatedShowTemporized](#) (int delay)
- void [clearAllActions](#) ()
- [DNotificationWidget](#) (const QString &text, QWidget \*const parent=nullptr)
- [DNotificationWidget](#) (QWidget \*const parent=nullptr)
- int [heightForWidth](#) (int width) const override
- QIcon [icon](#) () const
- bool [isCloseButtonVisible](#) () const
- bool [isHideAnimationRunning](#) () const
- bool [isShowAnimationRunning](#) () const
- [MessageType](#) [messageType](#) () const
- QSize [minimumSizeHint](#) () const override
- void [removeAction](#) (QAction \*action)
- QSize [sizeHint](#) () const override
- QString [text](#) () const
- bool [wordWrap](#) () const
- [~DNotificationWidget](#) () override

## Protected Member Functions

- bool [event](#) (QEvent \*event) override
- void [paintEvent](#) (QPaintEvent \*event) override
- void [resizeEvent](#) (QResizeEvent \*event) override

## Properties

- bool `closeButtonVisible`
- QIcon `icon`
- [MessageType](#) `messageType`
- QString `text`
- bool `wordWrap`

## Friends

- class `Private`

### 6.460.1 Detailed Description

This widget can be used to provide inline positive or negative feedback, or to implement opportunistic interactions.

### 6.460.2 Member Enumeration Documentation

#### 6.460.2.1 MessageType

```
enum Digikam::DNotificationWidget::MessageType
```

Available message types. The background colors are chosen depending on the message type.

### 6.460.3 Constructor & Destructor Documentation

#### 6.460.3.1 DNotificationWidget() [1/2]

```
Digikam::DNotificationWidget::DNotificationWidget (  
    QWidget *const parent = nullptr ) [explicit]
```

Constructs a [DNotificationWidget](#) with the specified `parent`.

#### 6.460.3.2 DNotificationWidget() [2/2]

```
Digikam::DNotificationWidget::DNotificationWidget (  
    const QString & text,  
    QWidget *const parent = nullptr ) [explicit]
```

Constructs a [DNotificationWidget](#) with the specified `parent` and contents `text`.

#### 6.460.3.3 ~DNotificationWidget()

```
Digikam::DNotificationWidget::~~DNotificationWidget ( ) [override]
```

Destructor.

### 6.460.4 Member Function Documentation

#### 6.460.4.1 addAction()

```
void Digikam::DNotificationWidget::addAction (  
    QAction * action )
```

Add `action` to the message widget. For each action a button is added to the message widget in the order the actions were added.

## Parameters

<i>action</i>	the action to add
---------------	-------------------

## See also

[removeAction\(\)](#), [QWidget::actions\(\)](#)

**6.460.4.2 animatedHide**

```
void Digikam::DNotificationWidget::animatedHide ( ) [slot]
```

Hide the widget using an animation.

**6.460.4.3 animatedShow**

```
void Digikam::DNotificationWidget::animatedShow ( ) [slot]
```

Show the widget using an animation.

**6.460.4.4 animatedShowTemporized()**

```
void Digikam::DNotificationWidget::animatedShowTemporized (
    int delay )
```

Show the widget using an animation. The widget is automatically hidden after the delay (in ms).

**6.460.4.5 clearAllActions()**

```
void Digikam::DNotificationWidget::clearAllActions ( )
```

clear all actions from the message widget.

## See also

[DNotificationWidget::MessageType](#), [addAction\(\)](#), [setMessageType\(\)](#)

**6.460.4.6 heightForWidth()**

```
int Digikam::DNotificationWidget::heightForWidth (
    int width ) const [override]
```

Returns the required height for width.

## Parameters

<i>width</i>	the width in pixels
--------------	---------------------

**6.460.4.7 hideAnimationFinished**

```
void Digikam::DNotificationWidget::hideAnimationFinished ( ) [signal]
```

This signal is emitted when the hide animation is finished, started by calling [animatedHide\(\)](#). If animations are disabled, this signal is emitted immediately after the message widget got hidden.

## Note

This signal is *not* emitted if the widget was hidden by calling `hide()`, so this signal is only useful in conjunction with [animatedHide\(\)](#).

## See also

[animatedHide\(\)](#)

**6.460.4.8 icon()**

```
QIcon Digikam::DNotificationWidget::icon ( ) const
```

The icon shown on the left of the text. By default, no icon is shown.

**6.460.4.9 isCloseButtonVisible()**

```
bool Digikam::DNotificationWidget::isCloseButtonVisible ( ) const
```

Check whether the close button is visible.

## See also

[setCloseButtonVisible\(\)](#)

**6.460.4.10 isHideAnimationRunning()**

```
bool Digikam::DNotificationWidget::isHideAnimationRunning ( ) const
```

Check whether the hide animation started by calling [animatedHide\(\)](#) is still running. If animations are disabled, this function always returns *false*.

## See also

[animatedHide\(\)](#), [hideAnimationFinished\(\)](#)

#### 6.460.4.11 isShowAnimationRunning()

```
bool Digikam::DNotificationWidget::isShowAnimationRunning ( ) const
```

Check whether the show animation started by calling [animatedShow\(\)](#) is still running. If animations are disabled, this function always returns *false*.

See also

[animatedShow\(\)](#), [showAnimationFinished\(\)](#)

#### 6.460.4.12 linkActivated

```
void Digikam::DNotificationWidget::linkActivated (
    const QString & contents ) [signal]
```

This signal is emitted when the user clicks a link in the text label. The URL referred to by the href anchor is passed in contents.

Parameters

<i>contents</i>	text of the href anchor
-----------------	-------------------------

See also

[QLabel::linkActivated\(\)](#)

#### 6.460.4.13 linkHovered

```
void Digikam::DNotificationWidget::linkHovered (
    const QString & contents ) [signal]
```

This signal is emitted when the user hovers over a link in the text label. The URL referred to by the href anchor is passed in contents.

Parameters

<i>contents</i>	text of the href anchor
-----------------	-------------------------

See also

[QLabel::linkHovered\(\)](#)

#### 6.460.4.14 messageType()

```
DNotificationWidget::MessageType Digikam::DNotificationWidget::messageType ( ) const
```

Get the type of this message. By default, the type is set to `DNotificationWidget::Information`.

See also

[DNotificationWidget::MessageType](#), [setMessageType\(\)](#)

#### 6.460.4.15 `minimumSizeHint()`

```
QSize Digikam::DNotificationWidget::minimumSizeHint ( ) const [override]
```

Returns the minimum size of the message widget.

#### 6.460.4.16 `removeAction()`

```
void Digikam::DNotificationWidget::removeAction (
    QAction * action )
```

Remove `action` from the message widget.

Parameters

<code>action</code>	the action to remove
---------------------	----------------------

See also

[DNotificationWidget::MessageType](#), [addAction\(\)](#), [setMessageType\(\)](#)

#### 6.460.4.17 `setCloseButtonVisible`

```
void Digikam::DNotificationWidget::setCloseButtonVisible (
    bool visible ) [slot]
```

Set the visibility of the close button. If `visible` is `true`, a close button is shown that calls [animatedHide\(\)](#) if clicked.

See also

[closeButtonVisible\(\)](#), [animatedHide\(\)](#)

#### 6.460.4.18 `setIcon`

```
void Digikam::DNotificationWidget::setIcon (
    const QIcon & icon ) [slot]
```

Define an icon to be shown on the left of the text



#### 6.460.4.19 setMessageType

```
void Digikam::DNotificationWidget::setMessageType (
    DNotificationWidget::MessageType type ) [slot]
```

Set the message type to `type`. By default, the message type is set to `DNotificationWidget::Information`.

See also

`messageType()`, [DNotificationWidget::MessageType](#)

#### 6.460.4.20 setText

```
void Digikam::DNotificationWidget::setText (
    const QString & text ) [slot]
```

Set the text of the message widget to `text`. If the message widget is already visible, the text changes on the fly.

Parameters

<code>text</code>	the text to display, rich text is allowed
-------------------	---

See also

`text()`

#### 6.460.4.21 setWordWrap

```
void Digikam::DNotificationWidget::setWordWrap (
    bool wordWrap ) [slot]
```

Set word wrap to `wordWrap`. If word wrap is enabled, the `text()` of the message widget is wrapped to fit the available width. If word wrap is disabled, the message widget's minimum size is such that the entire text fits.

Parameters

<code>wordWrap</code>	disable/enable word wrap
-----------------------	--------------------------

See also

`wordWrap()`

#### 6.460.4.22 showAnimationFinished

```
void Digikam::DNotificationWidget::showAnimationFinished ( ) [signal]
```

This signal is emitted when the show animation is finished, started by calling [animatedShow\(\)](#). If animations are disabled, this signal is emitted immediately after the message widget got shown.

**Note**

This signal is *not* emitted if the widget was shown by calling `show()`, so this signal is only useful in conjunction with `animatedShow()`.

**See also**

[animatedShow\(\)](#)

**6.460.4.23 sizeHint()**

```
QSize Digikam::DNotificationWidget::sizeHint ( ) const [override]
```

Returns the preferred size of the message widget.

**6.460.4.24 text()**

```
QString Digikam::DNotificationWidget::text ( ) const
```

Get the text of this message widget.

**See also**

[setText\(\)](#)

**6.460.4.25 wordWrap()**

```
bool Digikam::DNotificationWidget::wordWrap ( ) const
```

Check whether word wrap is enabled.

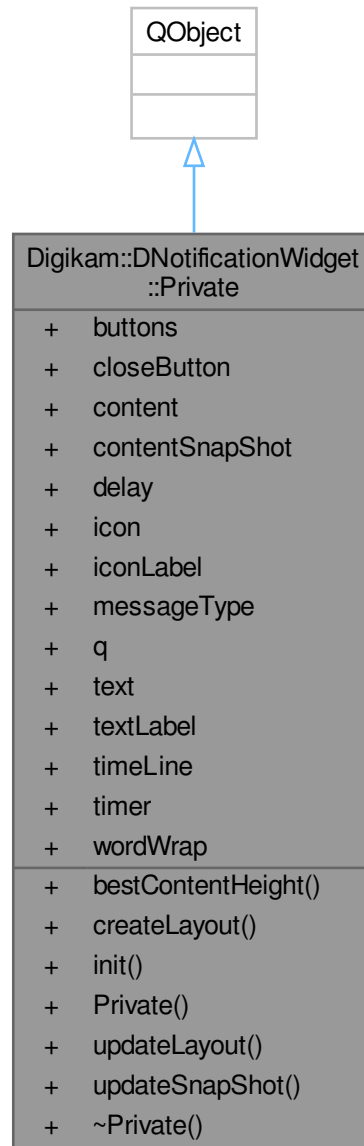
If word wrap is enabled, the message widget wraps the displayed text as required to the available width of the widget. This is useful to avoid breaking widget layouts.

**See also**

[setWordWrap\(\)](#)

## 6.461 Digikam::DNotificationWidget::Private Class Reference

Inheritance diagram for Digikam::DNotificationWidget::Private:



### Public Member Functions

- int **bestContentHeight** () const
- void **createLayout** ()
- void **init** ()
- **Private** ([DNotificationWidget](#) \*const)
- void **updateLayout** ()
- void **updateSnapShot** ()

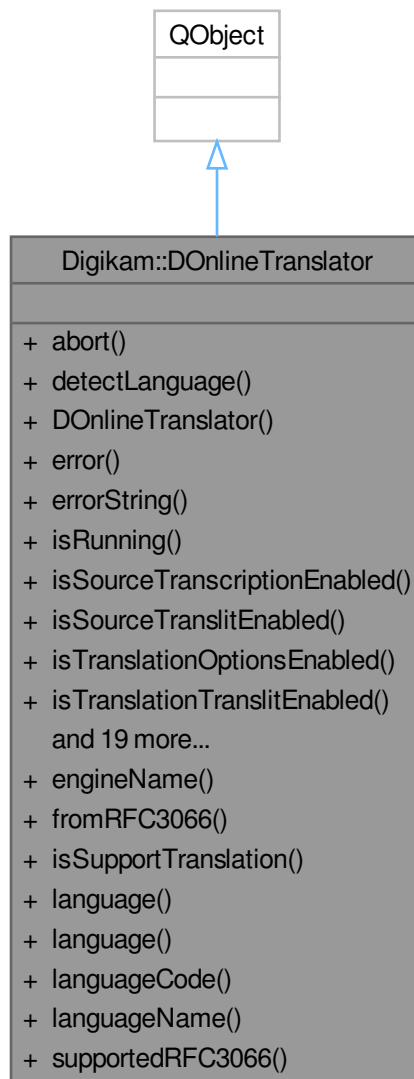
### Public Attributes

- `QList< QPushButton * >` **buttons**
- `QPushButton *` **closeButton** = `nullptr`
- `QFrame *` **content** = `nullptr`
- `QPixmap` **contentSnapshot**
- `int` **delay** = -1
- `QIcon` **icon**
- `QLabel *` **iconLabel** = `nullptr`
- `DNotificationWidget::MessageType` **messageType** = `DNotificationWidget::Information`
- `DNotificationWidget *` **q** = `nullptr`
- `QString` **text**
- `QLabel *` **textLabel** = `nullptr`
- `QTimeLine *` **timeLine** = `nullptr`
- `QTimer *` **timer** = `nullptr`
- `bool` **wordWrap** = `false`

## 6.462 Digikam::DOnlineTranslator Class Reference

Provides translation data.

Inheritance diagram for Digikam::DOnlineTranslator:



## Classes

- class [Private](#)

## Public Types

- enum [Engine](#) {  
**Google** , **Yandex** , **Bing** , **LibreTranslate** ,  
**Lingva** }

*Represents online engines.*

- enum [Language](#) {  
**NoLanguage** = -1 , **Auto** , **Afrikaans** , **Albanian** ,  
**Amharic** , **Arabic** , **Armenian** , **Azerbaijani** ,  
**Bashkir** , **Basque** , **Belarusian** , **Bengali** ,  
**Bosnian** , **Bulgarian** , **Cantonese** , **Catalan** ,  
**Cebuano** , **Chichewa** , **Corsican** , **Croatian** ,  
**Czech** , **Danish** , **Dutch** , **English** ,  
**Esperanto** , **Estonian** , **Fijian** , **Filipino** ,  
**Finnish** , **French** , **Frisian** , **Galician** ,  
**Georgian** , **German** , **Greek** , **Gujarati** ,  
**HaitianCreole** , **Hausa** , **Hawaiian** , **Hebrew** ,  
**HillMari** , **Hindi** , **Hmong** , **Hungarian** ,  
**Icelandic** , **Igbo** , **Indonesian** , **Irish** ,  
**Italian** , **Japanese** , **Javanese** , **Kannada** ,  
**Kazakh** , **Khmer** , **Kinyarwanda** , **Klingon** ,  
**KlingonPlqaD** , **Korean** , **Kurdish** , **Kyrgyz** ,  
**Lao** , **Latin** , **Latvian** , **LevantineArabic** ,  
**Lithuanian** , **Luxembourgish** , **Macedonian** , **Malagasy** ,  
**Malay** , **Malayalam** , **Maltese** , **Maori** ,  
**Marathi** , **Mari** , **Mongolian** , **Myanmar** ,  
**Nepali** , **Norwegian** , **Oriya** , **Papiamentu** ,  
**Pashto** , **Persian** , **Polish** , **Portuguese** ,  
**Punjabi** , **QueretaroOtomi** , **Romanian** , **Russian** ,  
**Samoan** , **ScotsGaelic** , **SerbianCyrillic** , **SerbianLatin** ,  
**Sesotho** , **Shona** , **SimplifiedChinese** , **Sindhi** ,  
**Sinhala** , **Slovak** , **Slovenian** , **Somali** ,  
**Spanish** , **Sundanese** , **Swahili** , **Swedish** ,  
**Tagalog** , **Tahitian** , **Tajik** , **Tamil** ,  
**Tatar** , **Telugu** , **Thai** , **Tongan** ,  
**TraditionalChinese** , **Turkish** , **Turkmen** , **Udmurt** ,  
**Uighur** , **Ukrainian** , **Urdu** , **Uzbek** ,  
**Vietnamese** , **Welsh** , **Xhosa** , **Yiddish** ,  
**Yoruba** , **YucatecMaya** , **Zulu** }

*Represents all languages for translation.*

- enum [TranslationError](#) {  
[NoError](#) , [ParametersError](#) , [NetworkError](#) , [ServiceError](#) ,  
[ParsingError](#) }

*Indicates all possible error conditions found during the processing of the translation.*

## Signals

- void [signalFinished](#) ()  
*Translation finished.*

## Public Member Functions

- void **abort** ()  
*Cancel translation operation (if any).*
- void [detectLanguage](#) (const QString &text, [Engine](#) engine=Google)  
*Detect language.*
- [DOnlineTranslator](#) (QObject \*const parent=nullptr)  
*Create object.*
- [TranslationError](#) **error** () const  
*Last error.*

- QString `errorString` () const  
*Last error string.*
- bool `isRunning` () const  
*Check translation progress.*
- bool `isSourceTranscriptionEnabled` () const  
*Check if source transcription is enabled.*
- bool `isSourceTranslitEnabled` () const  
*Check if source transliteration is enabled.*
- bool `isTranslationOptionsEnabled` () const  
*Check if translation options are enabled.*
- bool `isTranslationTranslitEnabled` () const  
*Check if translation transliteration is enabled.*
- void `setEngineApiKey` (Engine engine, const QByteArray &apiKey)  
*Set api key for engine.*
- void `setEngineUrl` (Engine engine, const QString &url)  
*Set the URL engine.*
- void `setSourceTranscriptionEnabled` (bool enable)  
*Enable or disable source transcription.*
- void `setSourceTranslitEnabled` (bool enable)  
*Enable or disable source transliteration.*
- void `setTranslationOptionsEnabled` (bool enable)  
*Enable or disable translation options.*
- void `setTranslationTranslitEnabled` (bool enable)  
*Enable or disable translation transliteration.*
- QString `source` () const  
*Source text.*
- Language `sourceLanguage` () const  
*Source language.*
- QString `sourceLanguageName` () const  
*Source language name.*
- QString `sourceTranscription` () const  
*Source transcription.*
- QString `sourceTranslit` () const  
*Source transliteration.*
- QJsonDocument `toJson` () const  
*Converts the object to JSON.*
- void `translate` (const QString &text, Engine engine=Google, Language translationLang=Auto, Language sourceLang=Auto, Language uiLang=Auto)  
*Translate text.*
- QString `translation` () const  
*Translated text.*
- Language `translationLanguage` () const  
*Translation language.*
- QString `translationLanguageName` () const  
*Translation language name.*
- QMap< QString, QVector< DOnlineTranslatorOption > > `translationOptions` () const  
*Translation options.*
- QString `translationTranslit` () const  
*Translation transliteration.*

## Static Public Member Functions

- static QString [engineName](#) ([Engine](#) engine)
- static QString [fromRFC3066](#) ([Engine](#) engine, const QString &langCodeRFC3066)
- static bool [isSupportTranslation](#) ([Engine](#) engine, [Language](#) lang)
 

*Check if transliteration is supported.*
- static [Language language](#) (const QLocale &locale)
 

*Language.*
- static [Language language](#) (const QString &langCode)
 

*Returns general language code.*
- static QString [languageCode](#) ([Language](#) lang)
 

*Language code.*
- static QString [languageName](#) ([Language](#) lang)
 

*Language name.*
- static QStringList [supportedRFC3066](#) ([Engine](#) engine)

## Friends

- class [DOnlineTts](#)

## 6.462.1 Member Enumeration Documentation

### 6.462.1.1 TranslationError

```
enum Digikam::DOnlineTranslator::TranslationError
```

#### Enumerator

NoError	No error condition
ParametersError	Unsupported combination of parameters
NetworkError	Network error
ServiceError	Service unavailable or maximum number of requests
ParsingError	The request could not be parsed (report a bug if you see this)

## 6.462.2 Constructor & Destructor Documentation

### 6.462.2.1 DOnlineTranslator()

```
Digikam::DOnlineTranslator::DOnlineTranslator (
    QObject *const parent = nullptr ) [explicit]
```

Constructs an object with empty data and with parent. You can use [translate\(\)](#) to send text to object.

#### Parameters

<i>parent</i>	the parent object
---------------	-------------------



## 6.462.3 Member Function Documentation

### 6.462.3.1 detectLanguage()

```
void Digikam::DOnlineTranslator::detectLanguage (
    const QString & text,
    Engine engine = Google )
```

#### Parameters

<i>text</i>	the text for language detection
<i>engine</i>	the engine to use

### 6.462.3.2 engineName()

```
QString Digikam::DOnlineTranslator::engineName (
    Engine engine ) [static]
```

Return the engine literal name.

### 6.462.3.3 error()

```
DOnlineTranslator::TranslationError Digikam::DOnlineTranslator::error ( ) const
```

Error that was found during the processing of the last translation. If no error was found, returns [DOnlineTranslator::NoError](#). The text of the error can be obtained by [errorString\(\)](#).

#### Returns

last error

### 6.462.3.4 errorString()

```
QString Digikam::DOnlineTranslator::errorString ( ) const
```

A human-readable description of the last translation error that occurred.

#### Returns

last error string

### 6.462.3.5 fromRFC3066()

```
QString Digikam::DOnlineTranslator::fromRFC3066 (
    Engine engine,
    const QString & langCodeRFC3066 ) [static]
```

Convert language RFC3066 to supported language code

### 6.462.3.6 isRunning()

```
bool Digikam::DOnlineTranslator::isRunning ( ) const
```

#### Returns

`true` when the translation is still processing and has not finished or was aborted yet.

### 6.462.3.7 isSourceTranscriptionEnabled()

```
bool Digikam::DOnlineTranslator::isSourceTranscriptionEnabled ( ) const
```

#### Returns

`true` if source transcription is enabled

### 6.462.3.8 isSourceTranslitEnabled()

```
bool Digikam::DOnlineTranslator::isSourceTranslitEnabled ( ) const
```

#### Returns

`true` if source transliteration is enabled

### 6.462.3.9 isSupportTranslation()

```
bool Digikam::DOnlineTranslator::isSupportTranslation (
    Engine engine,
    Language lang ) [static]
```

#### Parameters

<i>engine</i>	the engine to use
<i>lang</i>	language

#### Returns

`true` if the specified engine supports transliteration for specified language

### 6.462.3.10 isTranslationOptionsEnabled()

```
bool Digikam::DOnlineTranslator::isTranslationOptionsEnabled ( ) const
```

**Returns**

`true` if translation options are enabled

**See also**

[DOnlineTranslatorOption](#)

**6.462.3.11 isTranslationTranslitEnabled()**

```
bool Digikam::DOnlineTranslator::isTranslationTranslitEnabled ( ) const
```

**Returns**

`true` if translation transliteration is enabled

**6.462.3.12 language() [1/2]**

```
DOnlineTranslator::Language Digikam::DOnlineTranslator::language (
    const QLocale & locale ) [static]
```

**Parameters**

<i>locale</i>	the locale to use
---------------	-------------------

**Returns**

language

**6.462.3.13 language() [2/2]**

```
DOnlineTranslator::Language Digikam::DOnlineTranslator::language (
    const QString & langCode ) [static]
```

**Parameters**

<i>langCode</i>	code
-----------------	------

**Returns**

language

**6.462.3.14 languageCode()**

```
QString Digikam::DOnlineTranslator::languageCode (
    Language lang ) [static]
```

## Parameters

<i>lang</i>	language
-------------	----------

## Returns

language code

**6.462.3.15 languageName()**

```
QString Digikam::DOnlineTranslator::languageName (
    Language lang ) [static]
```

## Parameters

<i>lang</i>	language
-------------	----------

## Returns

language name

**6.462.3.16 setEngineApiKey()**

```
void Digikam::DOnlineTranslator::setEngineApiKey (
    Engine engine,
    const QByteArray & apiKey )
```

Affects only LibreTranslate.

## Parameters

<i>engine</i>	the engine to use
<i>apiKey</i>	your key for this particular instance

**6.462.3.17 setEngineUrl()**

```
void Digikam::DOnlineTranslator::setEngineUrl (
    Engine engine,
    const QString & url )
```

Only affects LibreTranslate and Lingva because these engines have multiple instances. You need to call this function to specify the URL of an instance for them.

## Parameters

<i>engine</i>	the engine to use
<i>url</i>	engine url

### 6.462.3.18 setSourceTranscriptionEnabled()

```
void Digikam::DOnlineTranslator::setSourceTranscriptionEnabled (
    bool enable )
```

#### Parameters

<i>enable</i>	whether to enable source transcription
---------------	--

### 6.462.3.19 setSourceTranslitEnabled()

```
void Digikam::DOnlineTranslator::setSourceTranslitEnabled (
    bool enable )
```

#### Parameters

<i>enable</i>	whether to enable source transliteration
---------------	--

### 6.462.3.20 setTranslationOptionsEnabled()

```
void Digikam::DOnlineTranslator::setTranslationOptionsEnabled (
    bool enable )
```

#### Parameters

<i>enable</i>	whether to enable translation options
---------------	---------------------------------------

#### See also

[DOnlineTranslatorOption](#)

### 6.462.3.21 setTranslationTranslitEnabled()

```
void Digikam::DOnlineTranslator::setTranslationTranslitEnabled (
    bool enable )
```

#### Parameters

<i>enable</i>	whether to enable translation transliteration
---------------	---

### 6.462.3.22 signalFinished

```
void Digikam::DOnlineTranslator::signalFinished ( ) [signal]
```

This signal is emitted when the translation is complete.

**6.462.3.23 source()**

```
QString Digikam::DOnlineTranslator::source ( ) const
```

**Returns**

source text

**6.462.3.24 sourceLanguage()**

```
DOnlineTranslator::Language Digikam::DOnlineTranslator::sourceLanguage ( ) const
```

**Returns**

language of the source text

**6.462.3.25 sourceLanguageName()**

```
QString Digikam::DOnlineTranslator::sourceLanguageName ( ) const
```

**Returns**

language name of the source text

**6.462.3.26 sourceTranscription()**

```
QString Digikam::DOnlineTranslator::sourceTranscription ( ) const
```

**Returns**

transcription of the source text

**6.462.3.27 sourceTranslit()**

```
QString Digikam::DOnlineTranslator::sourceTranslit ( ) const
```

**Returns**

transliteration of the source text

**6.462.3.28 supportedRFC3066()**

```
QStringList Digikam::DOnlineTranslator::supportedRFC3066 (
    Engine engine ) [static]
```

Return a list of all supported language in RFC3066.

### 6.462.3.29 toJson()

```
QJsonDocument Digikam::DOnlineTranslator::toJson ( ) const
```

#### Returns

JSON representation

### 6.462.3.30 translate()

```
void Digikam::DOnlineTranslator::translate (
    const QString & text,
    Engine engine = Google,
    Language translationLang = Auto,
    Language sourceLang = Auto,
    Language uiLang = Auto )
```

#### Parameters

<i>text</i>	the text to translate
<i>engine</i>	online engine to use
<i>translationLang</i>	language to translation
<i>sourceLang</i>	language of the passed text
<i>uiLang</i>	ui language to use for display

### 6.462.3.31 translation()

```
QString Digikam::DOnlineTranslator::translation ( ) const
```

#### Returns

translated text.

### 6.462.3.32 translationLanguage()

```
DOnlineTranslator::Language Digikam::DOnlineTranslator::translationLanguage ( ) const
```

#### Returns

language of the translated text

### 6.462.3.33 translationLanguageName()

```
QString Digikam::DOnlineTranslator::translationLanguageName ( ) const
```

#### Returns

language name of the translated text

### 6.462.3.34 translationOptions()

```
QMap< QString, QVector< DOnlineTranslatorOption > > Digikam::DOnlineTranslator::translationOptions ( ) const
```

#### Returns

QMap whose key represents the type of speech, and the value is a QVector of translation options

#### See also

[DOnlineTranslatorOption](#)

### 6.462.3.35 translationTranslit()

```
QString Digikam::DOnlineTranslator::translationTranslit ( ) const
```

#### Returns

transliteration of the translated text

## 6.463 Digikam::DOnlineTranslator::Private Class Reference

### Public Member Functions

- **Private** ([DOnlineTranslator](#) \*const parent)

### Public Attributes

- QPointer< QNetworkReply > **currentReply**
- [TranslationError](#) **error** = [NoError](#)
- QString **errorString**
- QByteArray **libreApiKey**
- QString **libreUrl**
- QString **lingvaUrl**
- QNetworkAccessManager \* **networkManager** = nullptr
- bool **onlyDetectLanguage** = false
- QString **source**
- [Language](#) **sourceLang** = [NoLanguage](#)
- QString **sourceTranscription**
- bool **sourceTranscriptionEnabled** = true
- QString **sourceTranslit**
- bool **sourceTranslitEnabled** = true
- QStateMachine \* **stateMachine** = nullptr
- QString **translation**
- [Language](#) **translationLang** = [NoLanguage](#)
- QMap< QString, QVector< [DOnlineTranslatorOption](#) > > **translationOptions**
- bool **translationOptionsEnabled** = true
- QString **translationTranslit**
- bool **translationTranslitEnabled** = true
- [Language](#) **uiLang** = [NoLanguage](#)



## Static Public Attributes

- static QString **s\_binglg**
- static QString **s\_binglid**
- static QByteArray **s\_bingKey**
- static const QMap< [DOnlineTranslator::Language](#), QString > **s\_bingLanguageCodes**
- static QByteArray **s\_bingToken**
- static constexpr int **s\_bingTranslateLimit** = 5001
- static const QMap< [DOnlineTranslator::Language](#), QString > **s\_genericLanguageCodes**
- static const QMap< [DOnlineTranslator::Language](#), QString > **s\_googleLanguageCodes**
- static constexpr int **s\_googleTranslateLimit** = 5000
- static constexpr int **s\_libreTranslateLimit** = 120
- static const QMap< [DOnlineTranslator::Language](#), QString > **s\_lingvaLanguageCodes**
- static const QMap< QString, QString > **s\_rfc3066LanguageCodesBing**
- static const QMap< QString, QString > **s\_rfc3066LanguageCodesGeneric**
- static const QMap< QString, QString > **s\_rfc3066LanguageCodesGoogle**
- static const QMap< QString, QString > **s\_rfc3066LanguageCodesLingva**
- static const QMap< QString, QString > **s\_rfc3066LanguageCodesYandex**
- static constexpr char **s\_textProperty** [] = "Text"
- static QString **s\_yandexKey**
- static const QMap< [DOnlineTranslator::Language](#), QString > **s\_yandexLanguageCodes**
- static constexpr int **s\_yandexTranslateLimit** = 150
- static constexpr int **s\_yandexTranslitLimit** = 180

## 6.463.1 Member Data Documentation

### 6.463.1.1 s\_bingLanguageCodes

```
const QMap< DOnlineTranslator::Language, QString > Digikam::DOnlineTranslator::Private::s_bingLanguageCodes [static]
```

#### Initial value:

```
=
{
    { DOnlineTranslator::Auto,                QStringLiteral("auto-detect")    },
    { DOnlineTranslator::Bosnian,             QStringLiteral("bs-Latn")       },
    { DOnlineTranslator::SerbianCyrillic,     QStringLiteral("sr-Cyrl")       },
    { DOnlineTranslator::SimplifiedChinese,   QStringLiteral("zh-Hans")       },
    { DOnlineTranslator::TraditionalChinese,  QStringLiteral("zh-Hant")       },
    { DOnlineTranslator::Hmong,               QStringLiteral("mww")           }
}
```

### 6.463.1.2 s\_googleLanguageCodes

```
const QMap< DOnlineTranslator::Language, QString > Digikam::DOnlineTranslator::Private::s_googleLanguageCodes [static]
```

#### Initial value:

```
=
{
    { DOnlineTranslator::Hebrew,              QStringLiteral("iw")            }
}
```

### 6.463.1.3 s\_lingvaLanguageCodes

```
const QMap< DOnlineTranslator::Language, QString > Digikam::DOnlineTranslator::Private::s_←
lingvaLanguageCodes [static]
```

#### Initial value:

```
=
{
    { DOnlineTranslator::SimplifiedChinese,   QStringLiteral("zh")           },
    { DOnlineTranslator::TraditionalChinese,  QStringLiteral("zh_HANT")     }
}
```

### 6.463.1.4 s\_rfc3066LanguageCodesBing

```
const QMap< QString, QString > Digikam::DOnlineTranslator::Private::s_rfc3066LanguageCodesBing
[static]
```

#### Initial value:

```
=
{
    { QLatin1String("bs-BG"), QLatin1String("bs-Latn") },
    { QLatin1String("sr-RS"), QLatin1String("sr-Cyrl") },
    { QLatin1String("zh-CN"), QLatin1String("zh-Hans") },
    { QLatin1String("zh-TW"), QLatin1String("zh-Hant") }
}
```

### 6.463.1.5 s\_rfc3066LanguageCodesGoogle

```
const QMap< QString, QString > Digikam::DOnlineTranslator::Private::s_rfc3066LanguageCodes←
Google [static]
```

#### Initial value:

```
=
{
    { QLatin1String("iw-IL"), QLatin1String("iw") }
}
```

### 6.463.1.6 s\_rfc3066LanguageCodesLingva

```
const QMap< QString, QString > Digikam::DOnlineTranslator::Private::s_rfc3066LanguageCodes←
Lingva [static]
```

#### Initial value:

```
=
{
    { QLatin1String("zh-TW"), QLatin1String("zh_HANT") },
    { QLatin1String("zh-MO"), QLatin1String("zh") },
    { QLatin1String("zh-HK"), QLatin1String("zh") },
    { QLatin1String("zh-SG"), QLatin1String("zh") }
}
```

### 6.463.1.7 s\_rfc3066LanguageCodesYandex

```
const QMap< QString, QString > Digikam::DOnlineTranslator::Private::s_rfc3066LanguageCodes←
Yandex [static]
```

#### Initial value:

```
=
{
    { QLatin1String("jv-ID"), QLatin1String("jv") },
    { QLatin1String("zn-CN"), QLatin1String("zn") }
}
```

### 6.463.1.8 s\_yandexLanguageCodes

```
const QMap< DOnlineTranslator::Language, QString > Digikam::DOnlineTranslator::Private::s_yandexLanguageCodes [static]
```

#### Initial value:

```
=
{
    { DOnlineTranslator::SimplifiedChinese,   QStringLiteral("zn")           },
    { DOnlineTranslator::Javanese,           QStringLiteral("jv")          }
}
```

## 6.464 Digikam::DOnlineTranslatorOption Struct Reference

Contains translation options for a single word.

### Public Member Functions

- QJsonObject [toJson](#) () const  
*Converts the object to JSON.*

### Public Attributes

- QString **gender**  
*Gender of the word.*
- QStringList **translations**  
*Associated translations for the word.*
- QString **word**  
*Word that specified for translation options.*

### 6.464.1 Detailed Description

Can be obtained from the QOnlineTranslator object.

#### Example:

```
QOnlineTranslator translator;
// Obtain translation

QTextStream out(stdout);

for (auto it = translator.translationOptions().cbegin() ; it != translator.translationOptions().cend() ;
     ++it)
{
    out << it.key() << ":" << endl; // Output the type of speech with a colon

    for (const auto &[word, gender, translations] : it.value())
    {
        out << " " << word << ": "; // Print the word
        out << translations;       // Print translations
        out << endl;
    }

    out << endl;
}
}
```

#### Possible output:

```
// verb:
// sagen: say, tell, speak, mean, utter
// sprechen: speak, talk, say, pronounce, militate, discourse
// meinen: think, mean, believe, say, opine, fancy
// heißen: mean, be called, be named, bid, tell, be titled
// äußern: express, comment, speak, voice, say, utter
// aussprechen: express, pronounce, say, speak, voice, enunciate
// vorbringen: make, put forward, raise, say, put, bring forward
// aufsagen: recite, say, speak

// noun:
// Sagen: say
// Mitspracherecht: say
```

## 6.464.2 Member Function Documentation

### 6.464.2.1 toJson()

```
QJsonObject Digikam::DOnlineTranslatorOption::toJson ( ) const [inline]
```

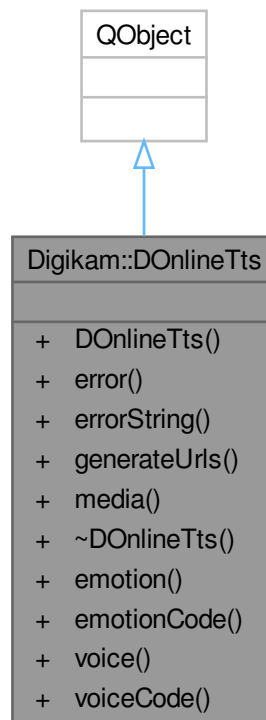
#### Returns

JSON representation

## 6.465 Digikam::DOnlineTts Class Reference

Provides TTS URL generation.

Inheritance diagram for Digikam::DOnlineTts:



#### Classes

- class [Private](#)

## Public Types

- enum [Emotion](#) { **NoEmotion** = -1 , **Neutral** , **Good** , **Evil** }  
*Defines emotion to use.*
- enum [TtsError](#) {  
[NoError](#) , [UnsupportedEngine](#) , [UnsupportedLanguage](#) , [UnsupportedVoice](#) ,  
[UnsupportedEmotion](#) }  
*Indicates all possible error conditions found during the processing of the URLs generation.*
- enum [Voice](#) {  
**NoVoice** = -1 , **Zahar** , **Ermil** , **Jane** ,  
**Oksana** , **Alyss** , **Omazh** }  
*Defines voice to use.*

## Public Member Functions

- [DOnlineTts](#) (QObject \*const parent=nullptr)  
*Create object.*
- [TtsError](#) [error](#) () const  
*Last error.*
- QString [errorString](#) () const  
*Last error string.*
- void [generateUrls](#) (const QString &text, [DOnlineTranslator::Engine](#) engine, [DOnlineTranslator::Language](#) lang, [Voice](#) voice=NoVoice, [Emotion](#) emotion=NoEmotion)  
*Create TTS urls.*
- QList< QUrl > [media](#) () const  
*Generated media.*

## Static Public Member Functions

- static [Emotion](#) [emotion](#) (const QString &emotionCode)  
*Emotion from code.*
- static QString [emotionCode](#) ([Emotion](#) emotion)  
*Code of the emotion.*
- static [Voice](#) [voice](#) (const QString &voiceCode)  
*Voice from code.*
- static QString [voiceCode](#) ([Voice](#) voice)  
*Code of the voice.*

## 6.465.1 Detailed Description

Example:

```
DOnlineTts tts;
tts.generateUrls(QLatin1String("Hello World!"), DOnlineTranslator::Google, DOnlineTranslator::English);

// Get list of Urls to play with media player.
QList<QUrl> urls = tts.media();
```

## 6.465.2 Member Enumeration Documentation

### 6.465.2.1 Emotion

```
enum Digikam::DOnlineTts::Emotion
```

Used only by Yandex.

## 6.465.2.2 TtsError

enum `Digikam::DOnlineTts::TtsError`

## Enumerator

NoError	No error condition
UnsupportedEngine	Specified engine does not support TTS
UnsupportedLanguage	Unsupported language by specified engine
UnsupportedVoice	Unsupported voice by specified engine
UnsupportedEmotion	Unsupported emotion by specified engine

## 6.465.2.3 Voice

```
enum Digikam::DOnlineTts::Voice
```

Used only by Yandex.

## 6.465.3 Constructor &amp; Destructor Documentation

## 6.465.3.1 DOnlineTts()

```
Digikam::DOnlineTts::DOnlineTts (
    QObject *const parent = nullptr ) [explicit]
```

Constructs an object with empty data and with parent. You can use [generateUrls\(\)](#) to create URLs for use in QMediaPlayer.

## Parameters

<i>parent</i>	the parent object
---------------	-------------------

## 6.465.4 Member Function Documentation

## 6.465.4.1 emotion()

```
DOnlineTts::Emotion Digikam::DOnlineTts::emotion (
    const QString & emotionCode ) [static]
```

Used only by Yandex.

## Parameters

<i>emotionCode</i>	emotion code
--------------------	--------------

## Returns

corresponding emotion

#### 6.465.4.2 emotionCode()

```
QString Digikam::DOnlineTts::emotionCode (
    Emotion emotion ) [static]
```

Used only by Yandex.

##### Parameters

<i>emotion</i>	the emotion to use
----------------	--------------------

##### Returns

code for emotion

#### 6.465.4.3 error()

```
DOnlineTts::TtsError Digikam::DOnlineTts::error ( ) const
```

Error that was found during the generating tts. If no error was found, returns `TtsError::NoError`. The text of the error can be obtained by `errorString()`.

##### Returns

last error

#### 6.465.4.4 errorString()

```
QString Digikam::DOnlineTts::errorString ( ) const
```

A human-readable description of the last tts URL generation error that occurred.

##### Returns

last error string

#### 6.465.4.5 generateUrls()

```
void Digikam::DOnlineTts::generateUrls (
    const QString & text,
    DOnlineTranslator::Engine engine,
    DOnlineTranslator::Language lang,
    Voice voice = NoVoice,
    Emotion emotion = NoEmotion )
```

Splits text into parts (engines have a limited number of characters per request) and returns list with the generated API URLs to play.



**Parameters**

<i>text</i>	the text to speak
<i>engine</i>	online translation engine
<i>lang</i>	text language
<i>voice</i>	the voice to use (used only by Yandex)
<i>emotion</i>	the emotion to use (used only by Yandex)

**6.465.4.6 media()**

```
QList< QUrl > Digikam::DOnlineTts::media ( ) const
```

**Returns**

List of generated URLs

**6.465.4.7 voice()**

```
DOnlineTts::Voice Digikam::DOnlineTts::voice (
    const QString & voiceCode ) [static]
```

Used only by Yandex.

**Parameters**

<i>voiceCode</i>	voice code
------------------	------------

**Returns**

corresponding voice

**6.465.4.8 voiceCode()**

```
QString Digikam::DOnlineTts::voiceCode (
    Voice voice ) [static]
```

**Parameters**

<i>voice</i>	the voice to use
--------------	------------------

## Returns

code for voice

## 6.466 Digikam::DOnlineTts::Private Class Reference

### Public Attributes

- [TtsError](#) **error** = [NoError](#)
- [QString](#) **errorString**
- [QList< QUrl >](#) **media**

### Static Public Attributes

- static const [QMap< Emotion, QString >](#) **s\_emotionCodes**
- static constexpr int **s\_googleTtsLimit** = 200
- static const [QMap< Voice, QString >](#) **s\_voiceCodes**
- static constexpr int **s\_yandexTtsLimit** = 1400

### 6.466.1 Member Data Documentation

#### 6.466.1.1 s\_emotionCodes

```
const QMap< DOnlineTts::Emotion, QString > Digikam::DOnlineTts::Private::s_emotionCodes [static]
```

##### Initial value:

```
=
{
    { Neutral, QStringLiteral("neutral") },
    { Good,   QStringLiteral("good")   },
    { Evil,   QStringLiteral("evil")   }
}
```

#### 6.466.1.2 s\_voiceCodes

```
const QMap< DOnlineTts::Voice, QString > Digikam::DOnlineTts::Private::s_voiceCodes [static]
```

##### Initial value:

```
=
{
    { Zahar,   QStringLiteral("zahar")   },
    { Ermil,   QStringLiteral("ermil")   },
    { Jane,    QStringLiteral("jane")    },
    { Oksana,  QStringLiteral("oksana")  },
    { Alyss,   QStringLiteral("alyss")   },
    { Omazh,   QStringLiteral("omazh")   }
}
```

## 6.467 Digikam::DownloadInfo Class Reference

### Public Member Functions

- [DownloadInfo](#) (const [DownloadInfo](#) &other)
- [DownloadInfo](#) (const [QString](#) &\_path, const [QString](#) &\_name, const [QString](#) &\_hash, const [qint64](#) &\_size)
- [DownloadInfo](#) & **operator=** (const [DownloadInfo](#) &other)

## Public Attributes

- QString [hash](#)
- QString [name](#)
- QString [path](#)
- qint64 [size](#) = 0

## 6.467.1 Member Data Documentation

### 6.467.1.1 hash

```
QString Digikam::DownloadInfo::hash
```

The file hash as SHA256.

### 6.467.1.2 name

```
QString Digikam::DownloadInfo::name
```

The file name on the server.

### 6.467.1.3 path

```
QString Digikam::DownloadInfo::path
```

The file path on the server.

### 6.467.1.4 size

```
qint64 Digikam::DownloadInfo::size = 0
```

The file size.

## 6.468 Digikam::DownloadSettings Class Reference

## Public Attributes

- bool **autoRotate** = true  
*Settings from [AdvancedSettings](#) widget.*
- int **colorLabel** = NoColorLabel  
*Pre-colorLabel of each camera file.*
- bool **convertJpeg** = false
- QString **dest**
- bool **documentName** = false
- QString **file**
- bool **fixDateTime** = false
- QString **folder**

- File path to download.*
- QString **losslessFormat**
  - New format to convert Jpeg files.*
- QString **mime**
  - Mime type from file to download.*
- QDateTime **newDateTime**
- int **pickLabel** = NoPickLabel
  - Pre-pickLabel of each camera file.*
- int **rating** = NoRating
  - Pre-rating of each camera file.*
- QString **script**
  - Settings from [ScriptingSettings](#) widget.*
- QList< int > **tagIds**
  - Pre-tags of each camera file.*
- QString **templateTitle**
  - Metadata template title.*

## 6.469 Digikam::DPixelsAliasFilter Class Reference

### Public Member Functions

- void [pixelAntiAliasing](#) (uchar \*const data, int Width, int Height, double X, double Y, uchar \*const A, uchar \*const R, uchar \*const G, uchar \*const B)
- void [pixelAntiAliasing16](#) (unsigned short \*const data, int Width, int Height, double X, double Y, unsigned short \*const A, unsigned short \*const R, unsigned short \*const G, unsigned short \*const B)

### 6.469.1 Member Function Documentation

#### 6.469.1.1 pixelAntiAliasing()

```
void Digikam::DPixelsAliasFilter::pixelAntiAliasing (
    uchar *const data,
    int Width,
    int Height,
    double X,
    double Y,
    uchar *const A,
    uchar *const R,
    uchar *const G,
    uchar *const B )
```

Function to perform pixel antialiasing with 8 bits/color/pixel images. This method is used to smooth target image in transformation method like free rotation or shear tool.

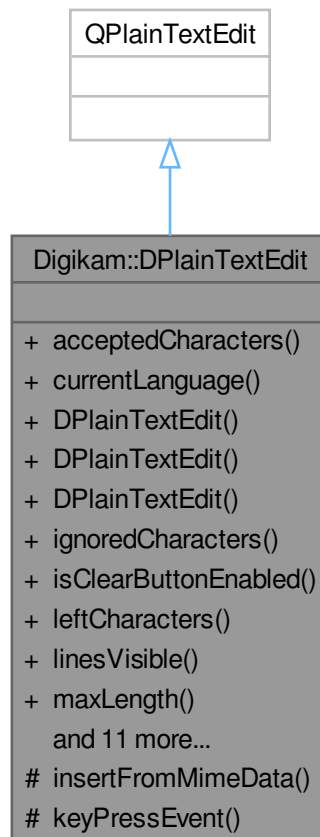
## 6.469.1.2 pixelAntiAliasing16()

```
void Digikam::DPixelsAliasFilter::pixelAntiAliasing16 (
    unsigned short *const data,
    int Width,
    int Height,
    double X,
    double Y,
    unsigned short *const A,
    unsigned short *const R,
    unsigned short *const G,
    unsigned short *const B )
```

Function to perform pixel antialiasing with 16 bits/color/pixel images. This method is used to smooth target image in transformation method like free rotation or shear tool.

## 6.470 Digikam::DPlainTextEdit Class Reference

Inheritance diagram for Digikam::DPlainTextEdit:



## Classes

- class [Private](#)

## Signals

- void [returnPressed](#) ()
- void **textEdited** (const QString &)

## Public Member Functions

- QString [acceptedCharacters](#) () const
- QString **currentLanguage** () const
- [DPlainTextEdit](#) (const QString &contents, QWidget \*const parent=nullptr)
- [DPlainTextEdit](#) (QWidget \*const parent=nullptr)
- [DPlainTextEdit](#) (unsigned int lines, QWidget \*const parent=nullptr)
- QString [ignoredCharacters](#) () const
- bool [isClearButtonEnabled](#) () const
- int [leftCharacters](#) () const
- unsigned int **linesVisible** () const
- int **maxLength** () const
- void **setAcceptedCharacters** (const QString &mask)
- void **setClearButtonEnabled** (bool enable)
- void [setCurrentLanguage](#) (const QString &lang)
- void **setIgnoredCharacters** (const QString &mask)
- void [setLinesVisible](#) (unsigned int lines)
- void **setLocalizeSettings** (const [LocalizeContainer](#) &settings)
- void [setMaxLength](#) (int length)
- void **setText** (const QString &text)
- [LocalizeContainer](#) [spellCheckSettings](#) () const
- QString [text](#) () const
- [~DPlainTextEdit](#) () override

## Protected Member Functions

- void **insertFromMimeData** (const QMimeData \*source) override
- void **keyPressEvent** (QKeyEvent \*e) override

### 6.470.1 Detailed Description

A text edit widget based on [QPlainTextEdit](#) with spell checker capabilities based on [Sonnet](#) (optional). Widget size can be constrained with the number of visible lines. A single line constraint will emulate [QLineEdit](#). See [setLinesVisible\(\)](#) for details. The maximum number of characters can be limited with [setMaxLength\(\)](#). The characters can be limited in editor by [setIgnoredCharacters\(\)](#) and [setAcceptedCharacters\(\)](#). Implementation↵  
: [dplaintextedit.cpp](#)

## 6.470.2 Constructor & Destructor Documentation

### 6.470.2.1 DPlainTextEdit() [1/3]

```
Digikam::DPlainTextEdit::DPlainTextEdit (  
    QWidget *const parent = nullptr ) [explicit]
```

Default constructor.

### 6.470.2.2 DPlainTextEdit() [2/3]

```
Digikam::DPlainTextEdit::DPlainTextEdit (  
    unsigned int lines,  
    QWidget *const parent = nullptr ) [explicit]
```

Constructor with a number of lines. Zero lines do not apply a size constraint.

### 6.470.2.3 DPlainTextEdit() [3/3]

```
Digikam::DPlainTextEdit::DPlainTextEdit (  
    const QString & contents,  
    QWidget *const parent = nullptr ) [explicit]
```

Constructor with text contents to use.

### 6.470.2.4 ~DPlainTextEdit()

```
Digikam::DPlainTextEdit::~DPlainTextEdit ( ) [override]
```

Standard destructor.

## 6.470.3 Member Function Documentation

### 6.470.3.1 acceptedCharacters()

```
QString Digikam::DPlainTextEdit::acceptedCharacters ( ) const
```

This property holds whether the edit widget handle the mask of accepted characters in text editor. The mask of characters is passed as string (ex: "abcABC"). By default the mask is empty.

### 6.470.3.2 ignoredCharacters()

```
QString Digikam::DPlainTextEdit::ignoredCharacters ( ) const
```

This property holds whether the edit widget handle the mask of ignored characters in text editor. The mask of characters is passed as string (ex: "+/!()"). By default the mask is empty.

### 6.470.3.3 isClearButtonEnabled()

```
bool Digikam::DPlainTextEdit::isClearButtonEnabled ( ) const
```

This property holds whether the edit widget displays a clear button when it is not empty. If enabled, the edit widget displays a trailing clear button when it contains some text, otherwise the edit widget does not show a clear button. This option only take effect in QLineEdit emulation mode when lines visible is set to 1. See [setLinesVisible\(\)](#) for details.

### 6.470.3.4 leftCharacters()

```
int Digikam::DPlainTextEdit::leftCharacters ( ) const
```

Return the left characters that user can enter if a limit have been previously set with `setMaxLeght()`.

### 6.470.3.5 returnPressed

```
void Digikam::DPlainTextEdit::returnPressed ( ) [signal]
```

Emmited only when mimic QLineEdit mode is enabled. See [setLinesVisible\(\)](#) for details.

### 6.470.3.6 setCurrentLanguage()

```
void Digikam::DPlainTextEdit::setCurrentLanguage (
    const QString & lang )
```

This property holds whether the edit widget handle a specific spell-checker language (2 letters code based as "en", "fr", "es", etc.). If this property is not set, spell-checker will try to auto-detect language by parsing the text. To reset this setting, pass a empty string as language. If Sonnet dependencies is not resolved, these method do nothing.

### 6.470.3.7 setLinesVisible()

```
void Digikam::DPlainTextEdit::setLinesVisible (
    unsigned int lines )
```

This property holds whether the edit widget handle visible lines used by the widget to show text. Lines must be superior or egal to 1 to apply a size constraint. Notes: if a single visible line is used, the widget emulate QLineEdit. a null value do not apply a size constraint.

### 6.470.3.8 setMaxLength()

```
void Digikam::DPlainTextEdit::setMaxLength (
    int length )
```

This property holds whether the edit widget handle the maximum of characters that user can enter in editor. By default no limit is set. A zero length reset a limit.



### 6.470.3.9 spellCheckSettings()

```
LocalizeContainer Digikam::DPlainTextEdit::spellCheckSettings ( ) const
```

This property holds whether the edit widget handle the Spellcheck settings. See [LocalizeContainer](#) class for details.

### 6.470.3.10 text()

```
QString Digikam::DPlainTextEdit::text ( ) const
```

This property holds whether the edit widget handle text contents as plain text. If ignored or accepted characters masks are set, text is filtered accordingly.

## 6.471 Digikam::DPlainTextEdit::Private Class Reference

### Public Member Functions

- void [init](#) ([DPlainTextEdit](#) \*const parent)

### Public Attributes

- QString **acceptedMask**  
*Mask of accepted characters in text editor.*
- bool **clearBtnEnable** = true
- [DTextEditClearButton](#) \* **clrBtn** = nullptr
- [LocalizeContainer](#) **container**  
*Spell checking settings container.*
- QString **ignoredMask**  
*Mask of ignored characters in text editor.*
- unsigned int **lines** = 3
- int **maxLength** = 0

### 6.471.1 Member Function Documentation

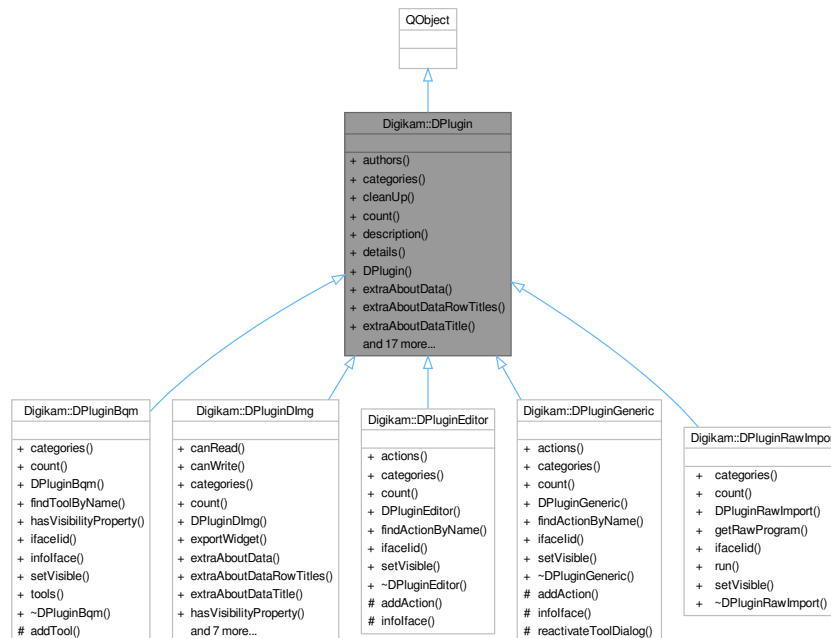
#### 6.471.1.1 init()

```
void Digikam::DPlainTextEdit::Private::init (
    DPlainTextEdit *const parent )
```

Init the text widget with the spell-checker engine (optional).

## 6.472 Digikam::DPlugin Class Reference

Inheritance diagram for Digikam::DPlugin:



### Public Member Functions

- virtual `QList< DPluginAuthor > authors ()` const =0  
*Returns authors list for the plugin.*
- virtual `QStringList categories ()` const =0
- virtual `void cleanUp ()`
- virtual `int count ()` const =0
- virtual `QString description ()` const =0  
*Returns a short description about the plugin.*
- virtual `QString details ()` const =0  
*Returns a long description about the plugin.*
- `DPlugin (QObject *const parent=nullptr)`
- virtual `QMap< QString, QStringList > extraAboutData ()` const  
*Returns a map of extra data to show in plugin about dialog.*
- virtual `QStringList extraAboutDataRowTitles ()` const  
*Returns a list of extra data row titles to show in tab of plugin about dialog.*
- virtual `QString extraAboutDataTitle ()` const  
*Returns the tab title of data returned by [extraAboutData\(\)](#).*
- virtual `QString handbookChapter ()` const
- virtual `QString handbookReference ()` const
- virtual `QString handbookSection ()` const
- virtual `bool hasVisibilityProperty ()` const
- virtual `QIcon icon ()` const  
*Returns an icon for the plugin. Default implementation return the system plugin icon.*
- virtual `QString ifaceId ()` const =0

Returns the unique top level internal identification property of the plugin interface. Must be formatted as "org.kde.digikam.digikam.\_NAME\_OF\_INTERFACE\_/\_VERSION\_". Examples: "org.kde.digikam.DPluginGeneric/1.1.0" "org.kde.digikam.DPluginEditor/1.1.0" "org.kde.digikam.DPluginBqm/1.1.0".

- virtual QString **iid** () const =0

Returns the unique internal identification property of the plugin. Must be formatted as "org.kde.digikam.plugin.\_PLUGIN\_TYPE\_.\_NAME\_OF\_PLUGIN\_". Examples: "org.kde.digikam.plugin.generic.Calendar" "org.kde.digikam.plugin.editor.AdjustCurvesTool" "org.kde.digikam.plugin.bqm.NoiseReduction".

- QString **libraryFileName** () const

Returns the file name of the library for this plugin. This string is filled at run-time by plugin loader.

- virtual QString **name** () const =0

Returns the user-visible name of the plugin.

- QStringList **pluginAuthors** () const
- void **setLibraryFileName** (const QString &)

Sets the file name of the library for this plugin. This string is filled at run-time by plugin loader.

- void **setShouldLoaded** (bool b)
- virtual void **setup** (QObject \*const parent)=0
- virtual void **setVisible** (bool b)=0
- bool **shouldLoaded** () const
- QString **version** () const
- ~DPlugin () override

## 6.472.1 Detailed Description

A digiKam external plugin abstract class.

## 6.472.2 Constructor & Destructor Documentation

### 6.472.2.1 DPlugin()

```
Digikam::DPlugin::DPlugin (
    QObject *const parent = nullptr ) [explicit]
```

Constructor with optional parent object

### 6.472.2.2 ~DPlugin()

```
Digikam::DPlugin::~~DPlugin ( ) [override]
```

Destructor

## 6.472.3 Member Function Documentation

### 6.472.3.1 categories()

```
virtual QStringList Digikam::DPlugin::categories ( ) const [pure virtual]
```

Return a list of categories as strings registered in this plugin.

Implemented in [Digikam::DPluginDImg](#), [Digikam::DPluginEditor](#), [Digikam::DPluginGeneric](#), [Digikam::DPluginRawImport](#), and [Digikam::DPluginBqm](#).

### 6.472.3.2 cleanUp()

```
virtual void Digikam::DPlugin::cleanUp ( ) [inline], [virtual]
```

Plugin method to clean up internal created objects. This method is called by plugin loader.

### 6.472.3.3 count()

```
virtual int Digikam::DPlugin::count ( ) const [pure virtual]
```

Return the amount of tools registered to all parents.

Implemented in [Digikam::DPluginDImg](#), [Digikam::DPluginEditor](#), [Digikam::DPluginGeneric](#), [Digikam::DPluginRawImport](#), and [Digikam::DPluginBqm](#).

### 6.472.3.4 extraAboutData()

```
virtual QMap< QString, QStringList > Digikam::DPlugin::extraAboutData ( ) const [inline], [virtual]
```

Reimplemented in [Digikam::DPluginDImg](#).

### 6.472.3.5 extraAboutDataRowTitles()

```
virtual QStringList Digikam::DPlugin::extraAboutDataRowTitles ( ) const [inline], [virtual]
```

Reimplemented in [Digikam::DPluginDImg](#).

### 6.472.3.6 extraAboutDataTitle()

```
virtual QString Digikam::DPlugin::extraAboutDataTitle ( ) const [inline], [virtual]
```

Reimplemented in [Digikam::DPluginDImg](#).

### 6.472.3.7 handbookChapter()

```
QString Digikam::DPlugin::handbookChapter ( ) const [virtual]
```

Return the online handbook chapter from an handbook section corresponding to this plugin. It's used in plugin dialog Help button. By default, no chapter is defined, and root page of the section is loaded by Help Button in this case. Note: a chapter is always included in a section. See [handbookSection\(\)](#) for details.

### 6.472.3.8 handbookReference()

```
QString Digikam::DPlugin::handbookReference ( ) const [virtual]
```

Return the online handbook reference from an handbook chapter corresponding to this plugin. It's used in plugin dialog Help button. By default, no reference is defined, and root page of the chapter is loaded by Help Button in this case. Note: a reference is always included in a chapter. See [handbookChapter\(\)](#) for details.

### 6.472.3.9 handbookSection()

```
QString Digikam::DPlugin::handbookSection ( ) const [virtual]
```

Return the online handbook section corresponding to this plugin. It's used in plugin dialog Help button. By default, no section is defined, and root page of the documentation is loaded by Help Button in this case.

### 6.472.3.10 hasVisibilityProperty()

```
bool Digikam::DPlugin::hasVisibilityProperty ( ) const [virtual]
```

Return true if plugin can be configured in setup dialog about the visibility property. Default implementation return true.

Reimplemented in [Digikam::DPluginDImg](#), and [Digikam::DPluginBqm](#).

### 6.472.3.11 ifaceId()

```
virtual QString Digikam::DPlugin::ifaceId ( ) const [pure virtual]
```

Implemented in [Digikam::DPluginDImg](#), [Digikam::DPluginEditor](#), [Digikam::DPluginGeneric](#), [Digikam::DPluginRawImport](#), and [Digikam::DPluginBqm](#).

### 6.472.3.12 name()

```
virtual QString Digikam::DPlugin::name ( ) const [pure virtual]
```

The user-visible name should be context free, i.e. the name should provide enough information as to what the plugin is about in the context of digiKam.

### 6.472.3.13 pluginAuthors()

```
QStringList Digikam::DPlugin::pluginAuthors ( ) const
```

Return a list of authors as strings registered in this plugin.

### 6.472.3.14 setShouldLoaded()

```
void Digikam::DPlugin::setShouldLoaded (
    bool b )
```

Accessor to adjust the should loaded plugin property. This property is adjusted by plugin loader at start-up.

### 6.472.3.15 setup()

```
virtual void Digikam::DPlugin::setup (
    QObject *const parent ) [pure virtual]
```

Plugin factory method to create all internal object instances for a given parent.

**6.472.3.16 setVisible()**

```
virtual void Digikam::DPlugin::setVisible (
    bool b ) [pure virtual]
```

Holds whether the plugin can be seen in parent view.

Implemented in [Digikam::DPluginEditor](#), [Digikam::DPluginGeneric](#), [Digikam::DPluginBqm](#), [Digikam::DPluginDImg](#), and [Digikam::DPluginRawImport](#).

**6.472.3.17 shouldLoaded()**

```
bool Digikam::DPlugin::shouldLoaded ( ) const
```

Return the should loaded property. If it's true, the plugin must be loaded in application GUI at startup by plugin loader.

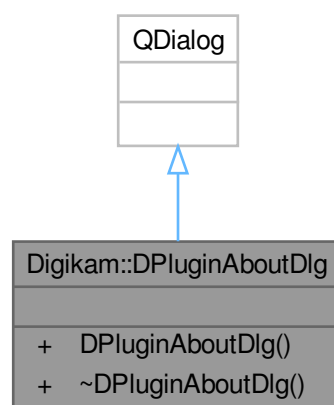
**6.472.3.18 version()**

```
QString Digikam::DPlugin::version ( ) const
```

Return the internal version used to check the binary compatibility at run-time. This is typically the same version of digiKam core used at compilation time.

**6.473 Digikam::DPluginAboutDlg Class Reference**

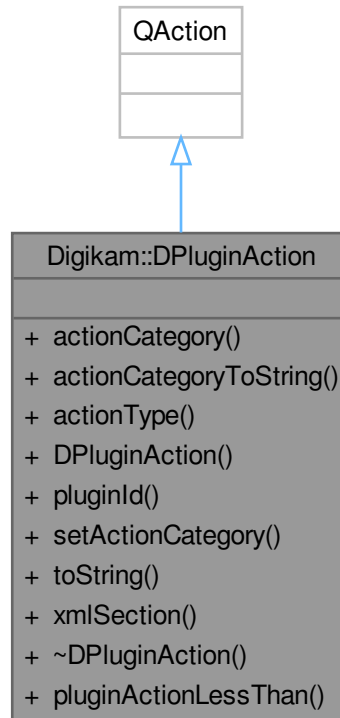
Inheritance diagram for Digikam::DPluginAboutDlg:

**Public Member Functions**

- **DPluginAboutDlg** ([DPlugin](#) \*const tool, [QWidget](#) \*const parent=nullptr)

## 6.474 Digikam::DPluginAction Class Reference

Inheritance diagram for Digikam::DPluginAction:



### Public Types

- enum [ActionCategory](#) { **InvalidCat** = -1 , [GenericExport](#) = 0 , [GenericImport](#) , [GenericTool](#) , [GenericMetadata](#) , [GenericView](#) , [EditorFile](#) , [EditorColors](#) , [EditorEnhance](#) , [EditorTransform](#) , [EditorDecorate](#) , [EditorFilters](#) }  
*Plugin action categories.*
- enum [ActionType](#) { **InvalidType** = -1 , **Generic** = 0 , **Editor** }  
*Plugin action types to resume where they can be used.*
- enum [PluginActionData](#) { **NoData** = 0 , **AlbumData** }  
*Plugin action types via QAction data container.*

### Public Member Functions

- [ActionCategory](#) **actionCategory** () const
- QString **actionCategoryToString** () const
- [ActionType](#) **actionType** () const
- **DPluginAction** (QObject \*const parent=nullptr)
- QString **pluginId** () const
- void **setActionCategory** ([ActionCategory](#) cat)
- QString **toString** () const
- QString **xmlSection** () const

## Static Public Member Functions

- static bool **pluginActionLessThan** ([DPluginAction](#) \*const a, [DPluginAction](#) \*const b)

## 6.474.1 Member Enumeration Documentation

### 6.474.1.1 ActionCategory

```
enum Digikam::DPluginAction::ActionCategory
```

#### Enumerator

GenericExport	Generic export action.
GenericImport	Generic import action.
GenericTool	Generic processing action.
GenericMetadata	Generic Metadata adjustment action.
GenericView	Generic View action (as Slideshow).
EditorFile	Image Editor file action.
EditorColors	Image Editor color correction action.
EditorEnhance	Image Editor enhance action.
EditorTransform	Image Editor transform action.
EditorDecorate	Image Editor decorate action.
EditorFilters	Image Editor special effects action.

### 6.474.1.2 ActionType

```
enum Digikam::DPluginAction::ActionType
```

#### Enumerator

InvalidType	An invalid action category.
Generic	Generic action available everywhere (AlbumView, Editor, and LightTable).
Editor	Specific action for Image Editor and Showfoto.

## 6.474.2 Member Function Documentation

### 6.474.2.1 actionType()

```
DPluginAction::ActionType Digikam::DPluginAction::actionType ( ) const
```

Return the action type depending of category.

### 6.474.2.2 pluginId()

```
QString Digikam::DPluginAction::pluginId ( ) const
```

Return the plugin id string hosting this action.



### 6.474.2.3 setActionCategory()

```
void Digikam::DPluginAction::setActionCategory (
    ActionCategory cat )
```

Manage the internal action category.

### 6.474.2.4 toString()

```
QString Digikam::DPluginAction::toString ( ) const
```

Return details as string about action properties. For debug purpose only.

### 6.474.2.5 xmlSection()

```
QString Digikam::DPluginAction::xmlSection ( ) const
```

Return the XML section to merge in KXMLGUIClient host XML definition.

## 6.475 Digikam::DPluginAuthor Class Reference

### Public Member Functions

- **DPluginAuthor** (const QString &\_name, const QString &\_email, const QString &\_year, const QString &\_role)
- **DPluginAuthor** (const QString &\_name, const QString &\_email, const QString &\_year)
- QString [toString](#) () const

### Public Attributes

- QString **email**  
*Email anti-spammed.*
- QString **name**  
*Author name and surname.*
- QString **roles**  
*Author roles, as "Developer", "Designer", "Translator", etc.*
- QString **years**  
*Copyrights years.*

## 6.475.1 Member Function Documentation

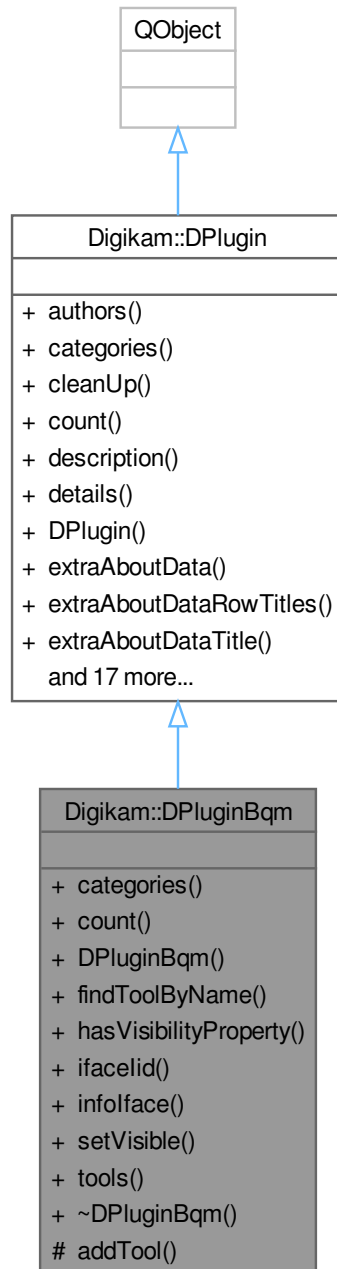
### 6.475.1.1 toString()

```
QString Digikam::DPluginAuthor::toString ( ) const
```

Return author details as string. For debug purpose only.

## 6.476 Digikam::DPluginBqm Class Reference

Inheritance diagram for Digikam::DPluginBqm:



### Signals

- void **signalVisible** (bool)

## Public Member Functions

- QStringList [categories](#) () const override
- int [count](#) () const override
- [DPluginBqm](#) (QObject \*const parent=nullptr)
- [BatchTool](#) \* [findToolByName](#) (const QString &name, QObject \*const parent) const
- bool [hasVisibilityProperty](#) () const override
- QString [ifacelid](#) () const override
- [BqmInfoface](#) \* [infoface](#) () const
- void [setVisible](#) (bool b) override
- QList< [BatchTool](#) \* > [tools](#) (QObject \*const parent) const
- ~[DPluginBqm](#) () override

## Public Member Functions inherited from [Digikam::DPlugin](#)

- virtual QList< [DPluginAuthor](#) > [authors](#) () const =0  
*Returns authors list for the plugin.*
- virtual void [cleanUp](#) ()
- virtual QString [description](#) () const =0  
*Returns a short description about the plugin.*
- virtual QString [details](#) () const =0  
*Returns a long description about the plugin.*
- [DPlugin](#) (QObject \*const parent=nullptr)
- virtual QMap< QString, QStringList > [extraAboutData](#) () const  
*Returns a map of extra data to show in plugin about dialog.*
- virtual QStringList [extraAboutDataRowTitles](#) () const  
*Returns a list of extra data row titles to show in tab of plugin about dialog.*
- virtual QString [extraAboutDataTitle](#) () const  
*Returns the tab title of data returned by [extraAboutData\(\)](#).*
- virtual QString [handbookChapter](#) () const
- virtual QString [handbookReference](#) () const
- virtual QString [handbookSection](#) () const
- virtual QIcon [icon](#) () const  
*Returns an icon for the plugin. Default implementation return the system plugin icon.*
- virtual QString [iid](#) () const =0  
*Returns the unique internal identification property of the plugin. Must be formatted as "org.kde.digikam.plugin.\_PLUGIN\_TYPE\_.NAME\_OF\_PLUGIN\_". Examples: "org.kde.digikam.plugin.generic.Calendar" "org.kde.digikam.plugin.editor.AdjustCurvesTool" "org.kde.digikam.plugin.bqm.NoiseReduction".*
- QString [libraryFileName](#) () const  
*Returns the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- virtual QString [name](#) () const =0  
*Returns the user-visible name of the plugin.*
- QStringList [pluginAuthors](#) () const
- void [setLibraryFileName](#) (const QString &)  
*Sets the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- void [setShouldLoaded](#) (bool b)
- virtual void [setup](#) (QObject \*const parent)=0
- bool [shouldLoaded](#) () const
- QString [version](#) () const
- ~[DPlugin](#) () override

## Protected Member Functions

- void **addTool** ([BatchTool](#) \*const t)

## 6.476.1 Constructor & Destructor Documentation

### 6.476.1.1 DPluginBqm()

```
Digikam::DPluginBqm::DPluginBqm (
    QObject *const parent = nullptr ) [explicit]
```

Constructor with optional parent object

### 6.476.1.2 ~DPluginBqm()

```
Digikam::DPluginBqm::~DPluginBqm ( ) [override]
```

Destructor

## 6.476.2 Member Function Documentation

### 6.476.2.1 categories()

```
QStringList Digikam::DPluginBqm::categories ( ) const [override], [virtual]
```

Return a list of batch tool group categories as strings registered in this plugin.

Implements [Digikam::DPlugin](#).

### 6.476.2.2 count()

```
int Digikam::DPluginBqm::count ( ) const [override], [virtual]
```

Return the amount of tools registered.

Implements [Digikam::DPlugin](#).

### 6.476.2.3 findToolByName()

```
BatchTool * Digikam::DPluginBqm::findToolByName (
    const QString & name,
    QObject *const parent ) const
```

Return a plugin tool instance found by name in plugin tools list for a given parent.

#### 6.476.2.4 hasVisibilityProperty()

```
bool Digikam::DPluginBqm::hasVisibilityProperty ( ) const [override], [virtual]
```

Return true if plugin can be configured in setup dialog about the visibility property. Default implementation return true.

Reimplemented from [Digikam::DPlugin](#).

#### 6.476.2.5 ifaceId()

```
QString Digikam::DPluginBqm::ifaceId ( ) const [override], [virtual]
```

Return the plugin interface identifier.

Implements [Digikam::DPlugin](#).

#### 6.476.2.6 infoIface()

```
BqmInfoIface * Digikam::DPluginBqm::infoIface ( ) const
```

Return the info interface instance.

#### 6.476.2.7 setVisible()

```
void Digikam::DPluginBqm::setVisible (
    bool b ) [override], [virtual]
```

Holds whether the plugin can be seen in parent view.

Implements [Digikam::DPlugin](#).

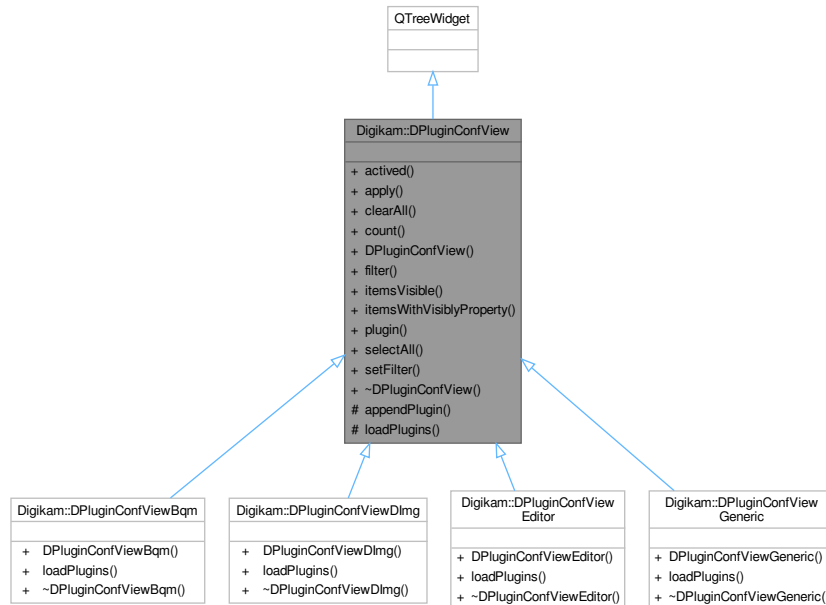
#### 6.476.2.8 tools()

```
QList< BatchTool * > Digikam::DPluginBqm::tools (
    QObject *const parent ) const
```

Return all plugin tools registered in [setup\(\)](#) method with [addTool\(\)](#) for a given parent.

## 6.477 Digikam::DPluginConfView Class Reference

Inheritance diagram for Digikam::DPluginConfView:



### Signals

- void [signalSearchResult](#) (int)

### Public Member Functions

- int [activated](#) () const
- void [apply](#) ()
- void [clearAll](#) ()
- int [count](#) () const
- [DPluginConfView](#) (QWidget \*const parent=nullptr)
- QString [filter](#) () const
- int [itemsVisible](#) () const
- int [itemsWithVisiblyProperty](#) () const
- [DPlugin](#) \* [plugin](#) (QTreeWidgetItem \*const item) const
- void [selectAll](#) () override
- void [setFilter](#) (const QString &filter, Qt::CaseSensitivity cs)

### Protected Member Functions

- QTreeWidgetItem \* [appendPlugin](#) ([DPlugin](#) \*const)
- virtual void [loadPlugins](#) ()=0

## 6.477.1 Constructor & Destructor Documentation

### 6.477.1.1 DPluginConfView()

```
Digikam::DPluginConfView::DPluginConfView (
    QWidget *const parent = nullptr ) [explicit]
```

Default constructor.

## 6.477.2 Member Function Documentation

### 6.477.2.1 actived()

```
int Digikam::DPluginConfView::actived ( ) const
```

Return the number of plugins active in the list.

### 6.477.2.2 apply()

```
void Digikam::DPluginConfView::apply ( )
```

Apply all changes about plugins selected to be hosted in host application.

### 6.477.2.3 clearAll()

```
void Digikam::DPluginConfView::clearAll ( )
```

Clear all selected plugins in the list.

### 6.477.2.4 count()

```
int Digikam::DPluginConfView::count ( ) const
```

Return the total number of plugins in the list.

### 6.477.2.5 filter()

```
QString Digikam::DPluginConfView::filter ( ) const
```

Return the current string used to filter the plugins list.

### 6.477.2.6 itemsVisible()

```
int Digikam::DPluginConfView::itemsVisible ( ) const
```

Return the number of visible plugins in the list.

### 6.477.2.7 itemsWithVisiblyProperty()

```
int Digikam::DPluginConfView::itemsWithVisiblyProperty ( ) const
```

Return the number of plugins in the list with visibly properties available.

### 6.477.2.8 selectAll()

```
void Digikam::DPluginConfView::selectAll ( ) [override]
```

Select all plugins in the list.

### 6.477.2.9 setFilter()

```
void Digikam::DPluginConfView::setFilter (
    const QString & filter,
    Qt::CaseSensitivity cs )
```

Set the string used to filter the plugins list. [signalSearchResult\(\)](#) is emitted when all is done.

### 6.477.2.10 signalSearchResult

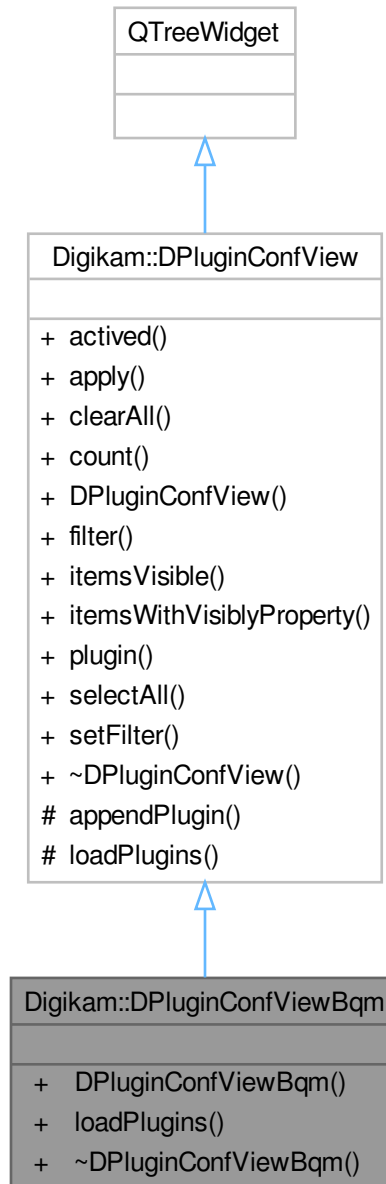
```
void Digikam::DPluginConfView::signalSearchResult (
    int ) [signal]
```

Signal emitted when filtering is done through `slotSetFilter()`. Number of plugins found is sent when item relevant of filtering match the query.



## 6.478 Digikam::DPluginConfViewBqm Class Reference

Inheritance diagram for Digikam::DPluginConfViewBqm:



### Public Member Functions

- **DPluginConfViewBqm** (QWidget \*const parent=nullptr)
- void [loadPlugins](#) () override

## Public Member Functions inherited from [Digikam::DPluginConfView](#)

- int [activated](#) () const
- void [apply](#) ()
- void [clearAll](#) ()
- int [count](#) () const
- [DPluginConfView](#) (QWidget \*const parent=nullptr)
- QString [filter](#) () const
- int [itemsVisible](#) () const
- int [itemsWithVisiblyProperty](#) () const
- [DPlugin](#) \* [plugin](#) (QTreeWidgetItem \*const item) const
- void [selectAll](#) () override
- void [setFilter](#) (const QString &filter, Qt::CaseSensitivity cs)

## Additional Inherited Members

## Signals inherited from [Digikam::DPluginConfView](#)

- void [signalSearchResult](#) (int)

## Protected Member Functions inherited from [Digikam::DPluginConfView](#)

- QTreeWidgetItem \* [appendPlugin](#) ([DPlugin](#) \*const)

## 6.478.1 Member Function Documentation

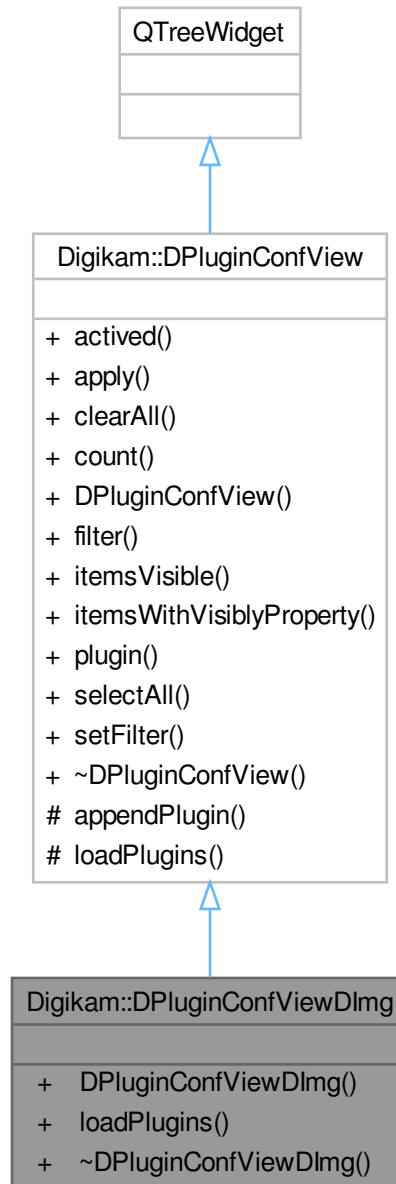
### 6.478.1.1 loadPlugins()

```
void Digikam::DPluginConfViewBqm::loadPlugins ( ) [override], [virtual]
```

Implements [Digikam::DPluginConfView](#).

## 6.479 Digikam::DPluginConfViewDImg Class Reference

Inheritance diagram for Digikam::DPluginConfViewDImg:



### Public Member Functions

- **DPluginConfViewDImg** (QWidget \*const parent=nullptr)
- void [loadPlugins](#) () override

## Public Member Functions inherited from [Digikam::DPluginConfView](#)

- int [activated](#) () const
- void [apply](#) ()
- void [clearAll](#) ()
- int [count](#) () const
- [DPluginConfView](#) (QWidget \*const parent=nullptr)
- QString [filter](#) () const
- int [itemsVisible](#) () const
- int [itemsWithVisiblyProperty](#) () const
- [DPlugin](#) \* [plugin](#) (QTreeWidgetItem \*const item) const
- void [selectAll](#) () override
- void [setFilter](#) (const QString &filter, Qt::CaseSensitivity cs)

## Additional Inherited Members

## Signals inherited from [Digikam::DPluginConfView](#)

- void [signalSearchResult](#) (int)

## Protected Member Functions inherited from [Digikam::DPluginConfView](#)

- QTreeWidgetItem \* [appendPlugin](#) ([DPlugin](#) \*const)

## 6.479.1 Member Function Documentation

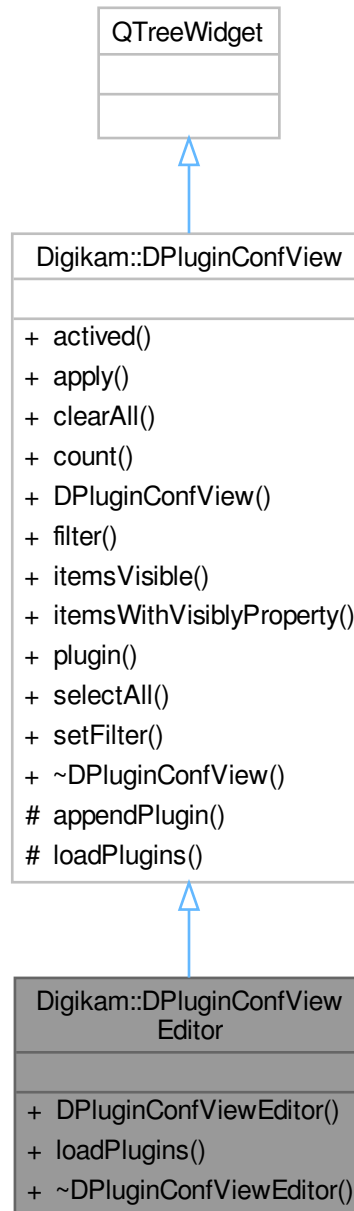
### 6.479.1.1 loadPlugins()

```
void Digikam::DPluginConfViewDImg::loadPlugins ( ) [override], [virtual]
```

Implements [Digikam::DPluginConfView](#).

## 6.480 Digikam::DPluginConfViewEditor Class Reference

Inheritance diagram for Digikam::DPluginConfViewEditor:



### Public Member Functions

- **DPluginConfViewEditor** (QWidget \*const parent=nullptr)
- void [loadPlugins](#) () override

## Public Member Functions inherited from [Digikam::DPluginConfView](#)

- int [activated](#) () const
- void [apply](#) ()
- void [clearAll](#) ()
- int [count](#) () const
- [DPluginConfView](#) (QWidget \*const parent=nullptr)
- QString [filter](#) () const
- int [itemsVisible](#) () const
- int [itemsWithVisiblyProperty](#) () const
- [DPlugin](#) \* [plugin](#) (QTreeWidgetItem \*const item) const
- void [selectAll](#) () override
- void [setFilter](#) (const QString &filter, Qt::CaseSensitivity cs)

## Additional Inherited Members

## Signals inherited from [Digikam::DPluginConfView](#)

- void [signalSearchResult](#) (int)

## Protected Member Functions inherited from [Digikam::DPluginConfView](#)

- QTreeWidgetItem \* [appendPlugin](#) ([DPlugin](#) \*const)

## 6.480.1 Member Function Documentation

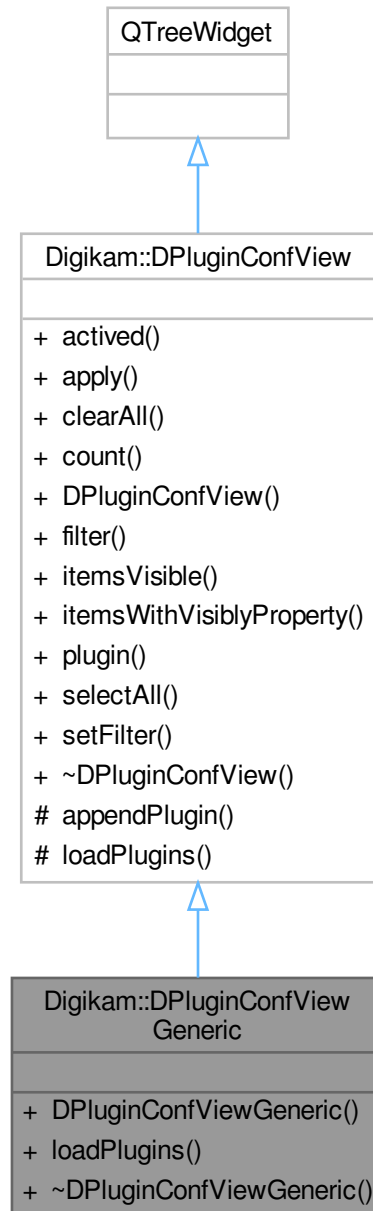
### 6.480.1.1 loadPlugins()

```
void Digikam::DPluginConfViewEditor::loadPlugins ( ) [override], [virtual]
```

Implements [Digikam::DPluginConfView](#).

## 6.481 Digikam::DPluginConfViewGeneric Class Reference

Inheritance diagram for Digikam::DPluginConfViewGeneric:



### Public Member Functions

- **DPluginConfViewGeneric** (QWidget \*const parent=nullptr)
- void [loadPlugins](#) () override

## Public Member Functions inherited from [Digikam::DPluginConfView](#)

- int [activated](#) () const
- void [apply](#) ()
- void [clearAll](#) ()
- int [count](#) () const
- [DPluginConfView](#) (QWidget \*const parent=nullptr)
- QString [filter](#) () const
- int [itemsVisible](#) () const
- int [itemsWithVisiblyProperty](#) () const
- [DPlugin](#) \* [plugin](#) (QTreeWidgetItem \*const item) const
- void [selectAll](#) () override
- void [setFilter](#) (const QString &filter, Qt::CaseSensitivity cs)

## Additional Inherited Members

## Signals inherited from [Digikam::DPluginConfView](#)

- void [signalSearchResult](#) (int)

## Protected Member Functions inherited from [Digikam::DPluginConfView](#)

- QTreeWidgetItem \* [appendPlugin](#) ([DPlugin](#) \*const)

## 6.481.1 Member Function Documentation

### 6.481.1.1 loadPlugins()

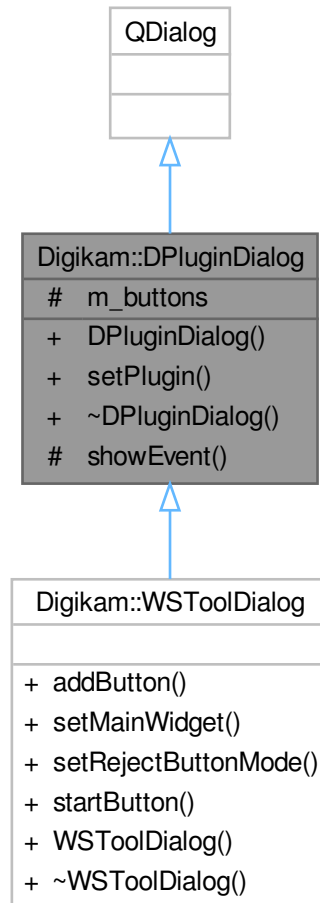
```
void Digikam::DPluginConfViewGeneric::loadPlugins ( ) [override], [virtual]
```

Implements [Digikam::DPluginConfView](#).



## 6.482 Digikam::DPluginDialog Class Reference

Inheritance diagram for Digikam::DPluginDialog:



### Public Member Functions

- **DPluginDialog** (QWidget \*const parent, const QString &objName)
- void **setPlugin** (DPlugin \*const tool)

### Protected Member Functions

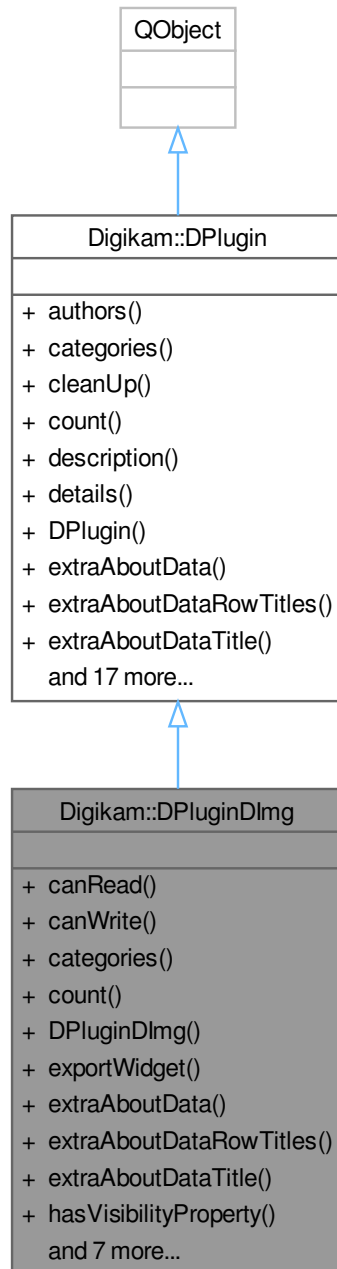
- void **showEvent** (QShowEvent \*) override

### Protected Attributes

- QDialogButtonBox \* **m\_buttons** = nullptr

## 6.483 Digikam::DPluginDImg Class Reference

Inheritance diagram for Digikam::DPluginDImg:



### Public Member Functions

- virtual int [canRead](#) (const QFileInfo &fileInfo, bool magic) const =0
- virtual int [canWrite](#) (const QString &format) const =0

- QStringList [categories](#) () const override
- int [count](#) () const override
- [DPluginDImg](#) (QObject \*const parent=nullptr)
- virtual [DImgLoaderSettings](#) \* [exportWidget](#) (const QString &format) const =0
- QMap< QString, QStringList > [extraAboutData](#) () const override
- QStringList [extraAboutDataRowTitles](#) () const override
  - Returns a list of extra data row titles to show in tab of plugin about dialog.*
- QString [extraAboutDataTitle](#) () const override
  - Returns the tab title of data returned by [extraAboutData\(\)](#).*
- bool [hasVisibilityProperty](#) () const override
- QString [ifacelid](#) () const override
- virtual [DImgLoader](#) \* [loader](#) ([DImg](#) \*const image, const [DRawDecoding](#) &rawSettings=[DRawDecoding\(\)](#)) const =0
- virtual QString [loaderName](#) () const =0
- virtual bool [previewSupported](#) () const
- void [setVisible](#) (bool) override
- virtual QString [typeMimes](#) () const =0
- [~DPluginDImg](#) () override=default

## Public Member Functions inherited from [Digikam::DPlugin](#)

- virtual QList< [DPluginAuthor](#) > [authors](#) () const =0
  - Returns authors list for the plugin.*
- virtual void [cleanUp](#) ()
- virtual QString [description](#) () const =0
  - Returns a short description about the plugin.*
- virtual QString [details](#) () const =0
  - Returns a long description about the plugin.*
- [DPlugin](#) (QObject \*const parent=nullptr)
- virtual QString [handbookChapter](#) () const
- virtual QString [handbookReference](#) () const
- virtual QString [handbookSection](#) () const
- virtual QIcon [icon](#) () const
  - Returns an icon for the plugin. Default implementation return the system plugin icon.*
- virtual QString [iid](#) () const =0
  - Returns the unique internal identification property of the plugin. Must be formatted as "org.kde.digikam.plugin.\_PLUGIN\_TYPE\_.NAME\_OF\_PLUGIN\_". Examples: "org.kde.digikam.plugin.generic.Calendar" "org.kde.digikam.plugin.editor.AdjustCurvesTool" "org.kde.digikam.plugin.bqm.NoiseReduction".*
- QString [libraryFileName](#) () const
  - Returns the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- virtual QString [name](#) () const =0
  - Returns the user-visible name of the plugin.*
- QStringList [pluginAuthors](#) () const
- void [setLibraryFileName](#) (const QString &)
  - Sets the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- void [setShouldLoaded](#) (bool b)
- virtual void [setup](#) (QObject \*const parent)=0
- bool [shouldLoaded](#) () const
- QString [version](#) () const
- [~DPlugin](#) () override

## 6.483.1 Constructor & Destructor Documentation

### 6.483.1.1 DPluginDImg()

```
Digikam::DPluginDImg::DPluginDImg (
    QObject *const parent = nullptr ) [explicit]
```

Constructor with optional parent object

### 6.483.1.2 ~DPluginDImg()

```
Digikam::DPluginDImg::~~DPluginDImg ( ) [override], [default]
```

Destructor

## 6.483.2 Member Function Documentation

### 6.483.2.1 canRead()

```
virtual int Digikam::DPluginDImg::canRead (
    const QFileInfo & fileInfo,
    bool magic ) const [pure virtual]
```

Return  $> 0$  if source file path is supported by the loader and contents can be loaded. The return value (1 - 100) is a priority. digiKam default loaders have a priority of 10, the QImage loader has a priority of 80 and the ImageMagick loader has a priority of 90. If the loader is to be used before the default loader, the value must be less than 10.

### 6.483.2.2 canWrite()

```
virtual int Digikam::DPluginDImg::canWrite (
    const QString & format ) const [pure virtual]
```

Return  $> 0$  if target file format is supported by the loader and contents can be written. The return value (1 - 100) is a priority.

### 6.483.2.3 categories()

```
QStringList Digikam::DPluginDImg::categories ( ) const [inline], [override], [virtual]
```

This kind of plugin do not use a category.

Implements [Digikam::DPlugin](#).

### 6.483.2.4 count()

```
int Digikam::DPluginDImg::count ( ) const [inline], [override], [virtual]
```

This kind of plugin only provide one tool.

Implements [Digikam::DPlugin](#).

### 6.483.2.5 exportWidget()

```
virtual DImgLoaderSettings * Digikam::DPluginDImg::exportWidget (
    const QString & format ) const [pure virtual]
```

Return a new widget instance to show settings while exporting image to specified format. Return nullptr if format is not supported or if no settings widget is available for this format.

### 6.483.2.6 extraAboutData()

```
QMap< QString, QStringList > Digikam::DPluginDImg::extraAboutData ( ) const [override], [virtual]
```

With this kind of plugin, we will display the type-mimes list on about dialog.

Reimplemented from [Digikam::DPlugin](#).

### 6.483.2.7 extraAboutDataRowTitles()

```
QStringList Digikam::DPluginDImg::extraAboutDataRowTitles ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DPlugin](#).

### 6.483.2.8 extraAboutDataTitle()

```
QString Digikam::DPluginDImg::extraAboutDataTitle ( ) const [override], [virtual]
```

Reimplemented from [Digikam::DPlugin](#).

### 6.483.2.9 hasVisibilityProperty()

```
bool Digikam::DPluginDImg::hasVisibilityProperty ( ) const [inline], [override], [virtual]
```

This kind of plugin do not need to be configurable

Reimplemented from [Digikam::DPlugin](#).

### 6.483.2.10 ifacelid()

```
QString Digikam::DPluginDImg::ifaceId ( ) const [inline], [override], [virtual]
```

Return the plugin interface identifier.

Implements [Digikam::DPlugin](#).

### 6.483.2.11 loader()

```
virtual DImgLoader * Digikam::DPluginDImg::loader (
    DImg *const image,
    const DRawDecoding & rawSettings = DRawDecoding\(\) ) const [pure virtual]
```

Return the image loader instance for the [DImg](#) instance.

### 6.483.2.12 loaderName()

```
virtual QString Digikam::DPluginDImg::loaderName ( ) const [pure virtual]
```

Return a single capitalized word to identify the format supported by the loader. Ex: jpeg => "JPG" ; tiff => "TIF", etc.

### 6.483.2.13 previewSupported()

```
virtual bool Digikam::DPluginDImg::previewSupported ( ) const [inline], [virtual]
```

Return true if the loader can read a preview image.

### 6.483.2.14 setVisible()

```
void Digikam::DPluginDImg::setVisible (
    bool ) [inline], [override], [virtual]
```

This kind of plugin do not have GUI visibility attribute.

Implements [Digikam::DPlugin](#).

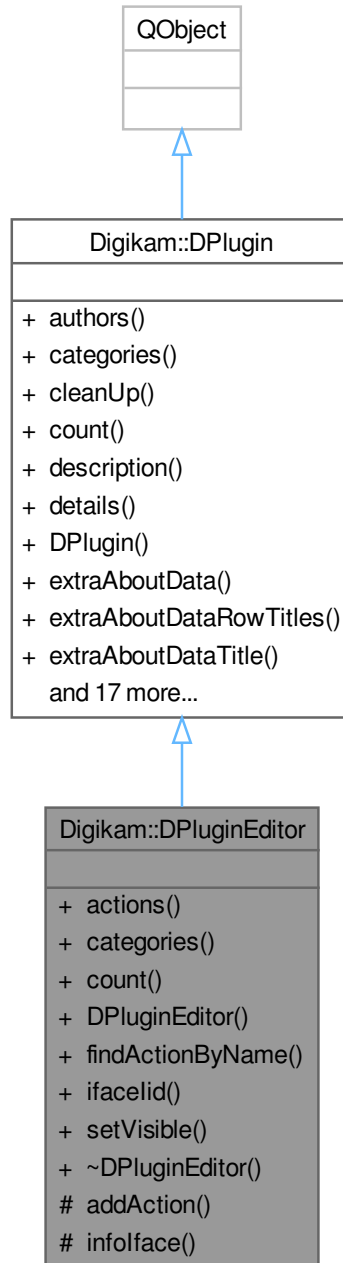
### 6.483.2.15 typeMimes()

```
virtual QString Digikam::DPluginDImg::typeMimes ( ) const [pure virtual]
```

Return the list of white-listed type-mimes supported by the loader, as a string of file-name suffix separated by spaces. Ex: "jpeg jpg thm"

## 6.484 Digikam::DPluginEditor Class Reference

Inheritance diagram for Digikam::DPluginEditor:



### Public Member Functions

- `QList< DPluginAction * > actions` (`QObject *const parent`) const
- `QStringList categories` () const override

- int `count` () const override
- `DPluginEditor` (QObject \*const parent=nullptr)
- `DPluginAction` \* `findActionByName` (const QString &name, QObject \*const parent) const
- QString `ifacelid` () const override
- void `setVisible` (bool b) override
- ~`DPluginEditor` () override

## Public Member Functions inherited from `Digikam::DPlugin`

- virtual QList< `DPluginAuthor` > `authors` () const =0  
*Returns authors list for the plugin.*
- virtual void `cleanUp` ()
- virtual QString `description` () const =0  
*Returns a short description about the plugin.*
- virtual QString `details` () const =0  
*Returns a long description about the plugin.*
- `DPlugin` (QObject \*const parent=nullptr)
- virtual QMap< QString, QStringList > `extraAboutData` () const  
*Returns a map of extra data to show in plugin about dialog.*
- virtual QStringList `extraAboutDataRowTitles` () const  
*Returns a list of extra data row titles to show in tab of plugin about dialog.*
- virtual QString `extraAboutDataTitle` () const  
*Returns the tab title of data returned by `extraAboutData()`.*
- virtual QString `handbookChapter` () const
- virtual QString `handbookReference` () const
- virtual QString `handbookSection` () const
- virtual bool `hasVisibilityProperty` () const
- virtual QIcon `icon` () const  
*Returns an icon for the plugin. Default implementation return the system plugin icon.*
- virtual QString `iid` () const =0  
*Returns the unique internal identification property of the plugin. Must be formatted as "org.kde.digikam.<->plugin.\_PLUGIN\_TYPE\_.\_NAME\_OF\_PLUGIN\_". Examples: "org.kde.digikam.plugin.generic.Calendar" "org.kde.<->digikam.plugin.editor.AdjustCurvesTool" "org.kde.digikam.plugin.bqm.NoiseReduction".*
- QString `libraryFileName` () const  
*Returns the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- virtual QString `name` () const =0  
*Returns the user-visible name of the plugin.*
- QStringList `pluginAuthors` () const
- void `setLibraryFileName` (const QString &)  
*Sets the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- void `setShouldLoaded` (bool b)
- virtual void `setup` (QObject \*const parent)=0
- bool `shouldLoaded` () const
- QString `version` () const
- ~`DPlugin` () override

## Protected Member Functions

- void `addAction` (`DPluginAction` \*const ac)
- `DInfoInterface` \* `infoiface` (QObject \*const ac) const



## 6.484.1 Constructor & Destructor Documentation

### 6.484.1.1 DPluginEditor()

```
Digikam::DPluginEditor::DPluginEditor (
    QObject *const parent = nullptr ) [explicit]
```

Constructor with optional parent object

### 6.484.1.2 ~DPluginEditor()

```
Digikam::DPluginEditor::~~DPluginEditor ( ) [override]
```

Destructor

## 6.484.2 Member Function Documentation

### 6.484.2.1 actions()

```
QList< DPluginAction * > Digikam::DPluginEditor::actions (
    QObject *const parent ) const
```

Return all plugin actions registered in [setup\(\)](#) method with [addAction\(\)](#) for a given parent.

### 6.484.2.2 categories()

```
QStringList Digikam::DPluginEditor::categories ( ) const [override], [virtual]
```

Return a list of categories as strings registered in this plugin.

Implements [Digikam::DPlugin](#).

### 6.484.2.3 count()

```
int Digikam::DPluginEditor::count ( ) const [override], [virtual]
```

Return the amount of tools registered to all parents.

Implements [Digikam::DPlugin](#).

### 6.484.2.4 findActionByName()

```
DPluginAction * Digikam::DPluginEditor::findActionByName (
    const QString & name,
    QObject *const parent ) const
```

Return a plugin action instance found by name in plugin action list for a given parent.

### 6.484.2.5 ifaceId()

```
QString Digikam::DPluginEditor::ifaceId ( ) const [inline], [override], [virtual]
```

Return the plugin interface identifier.

Implements [Digikam::DPlugin](#).

### 6.484.2.6 infoIface()

```
DInfoInterface * Digikam::DPluginEditor::infoIface (
    QObject *const ac ) const [protected]
```

Return the info interface instance for the given action.

### 6.484.2.7 setVisible()

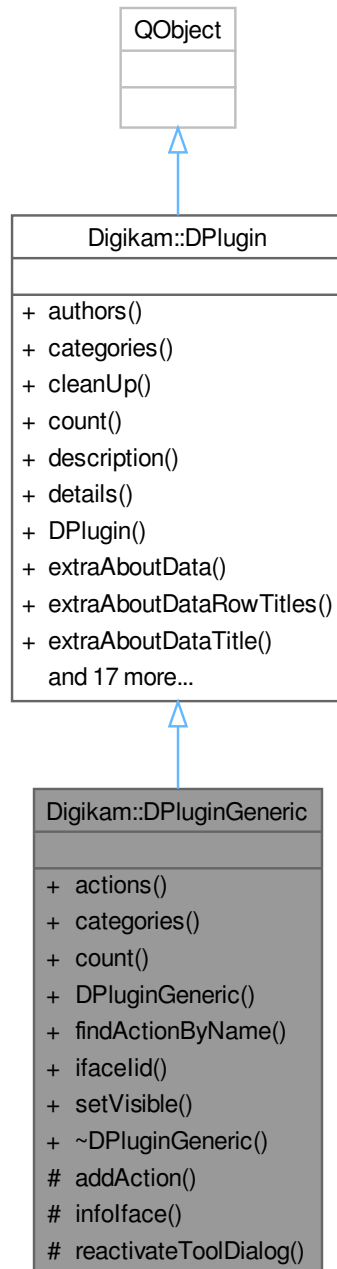
```
void Digikam::DPluginEditor::setVisible (
    bool b ) [override], [virtual]
```

Holds whether the plugin can be seen in parent view.

Implements [Digikam::DPlugin](#).

## 6.485 Digikam::DPluginGeneric Class Reference

Inheritance diagram for Digikam::DPluginGeneric:



### Public Member Functions

- `QList< DPluginAction * > actions` (`QObject *const parent`) `const`
- `QStringList categories` () `const` override

- int `count` () const override
- `DPluginGeneric` (QObject \*const parent=nullptr)
- `DPluginAction` \* `findActionByName` (const QString &name, QObject \*const parent) const
- QString `ifacelid` () const override
- void `setVisible` (bool b) override
- ~`DPluginGeneric` () override

## Public Member Functions inherited from `Digikam::DPlugin`

- virtual QList< `DPluginAuthor` > `authors` () const =0  
*Returns authors list for the plugin.*
- virtual void `cleanUp` ()
- virtual QString `description` () const =0  
*Returns a short description about the plugin.*
- virtual QString `details` () const =0  
*Returns a long description about the plugin.*
- `DPlugin` (QObject \*const parent=nullptr)
- virtual QMap< QString, QStringList > `extraAboutData` () const  
*Returns a map of extra data to show in plugin about dialog.*
- virtual QStringList `extraAboutDataRowTitles` () const  
*Returns a list of extra data row titles to show in tab of plugin about dialog.*
- virtual QString `extraAboutDataTitle` () const  
*Returns the tab title of data returned by `extraAboutData()`.*
- virtual QString `handbookChapter` () const
- virtual QString `handbookReference` () const
- virtual QString `handbookSection` () const
- virtual bool `hasVisibilityProperty` () const
- virtual QIcon `icon` () const  
*Returns an icon for the plugin. Default implementation return the system plugin icon.*
- virtual QString `iid` () const =0  
*Returns the unique internal identification property of the plugin. Must be formatted as "org.kde.digikam.plugin.\_PLUGIN\_TYPE\_.\_NAME\_OF\_PLUGIN\_". Examples: "org.kde.digikam.plugin.generic.Calendar" "org.kde.digikam.plugin.editor.AdjustCurvesTool" "org.kde.digikam.plugin.bqm.NoiseReduction".*
- QString `libraryFileName` () const  
*Returns the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- virtual QString `name` () const =0  
*Returns the user-visible name of the plugin.*
- QStringList `pluginAuthors` () const
- void `setLibraryFileName` (const QString &)  
*Sets the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- void `setShouldLoaded` (bool b)
- virtual void `setup` (QObject \*const parent)=0
- bool `shouldLoaded` () const
- QString `version` () const
- ~`DPlugin` () override

## Protected Member Functions

- void `addAction` (`DPluginAction` \*const ac)
- `DInfoInterface` \* `infoface` (QObject \*const ac) const
- bool `reactivateToolDialog` (QWidget \*const dlg) const

## 6.485.1 Constructor & Destructor Documentation

### 6.485.1.1 DPluginGeneric()

```
Digikam::DPluginGeneric::DPluginGeneric (
    QObject *const parent = nullptr ) [explicit]
```

Constructor with optional parent object

### 6.485.1.2 ~DPluginGeneric()

```
Digikam::DPluginGeneric::~~DPluginGeneric ( ) [override]
```

Destructor

## 6.485.2 Member Function Documentation

### 6.485.2.1 actions()

```
QList< DPluginAction * > Digikam::DPluginGeneric::actions (
    QObject *const parent ) const
```

Return all plugin actions registered in [setup\(\)](#) method with [addAction\(\)](#) for a given parent.

### 6.485.2.2 categories()

```
QStringList Digikam::DPluginGeneric::categories ( ) const [override], [virtual]
```

Return a list of categories as strings registered in this plugin.

Implements [Digikam::DPlugin](#).

### 6.485.2.3 count()

```
int Digikam::DPluginGeneric::count ( ) const [override], [virtual]
```

Return the amount of tools registered to all parents.

Implements [Digikam::DPlugin](#).

### 6.485.2.4 findActionByName()

```
DPluginAction * Digikam::DPluginGeneric::findActionByName (
    const QString & name,
    QObject *const parent ) const
```

Return a plugin action instance found by name in plugin action list for a given parent.

### 6.485.2.5 ifaceId()

```
QString Digikam::DPluginGeneric::ifaceId ( ) const [inline], [override], [virtual]
```

Return the plugin interface identifier.

Implements [Digikam::DPlugin](#).

### 6.485.2.6 infoIface()

```
DInfoInterface * Digikam::DPluginGeneric::infoIface (
    QObject *const ac ) const [protected]
```

Return the info interface instance for the given action object.

### 6.485.2.7 reactivateToolDialog()

```
bool Digikam::DPluginGeneric::reactivateToolDialog (
    QWidget *const dlg ) const [protected]
```

Helper function to reactivate the desktop visibility of tool widget.

### 6.485.2.8 setVisible()

```
void Digikam::DPluginGeneric::setVisible (
    bool b ) [override], [virtual]
```

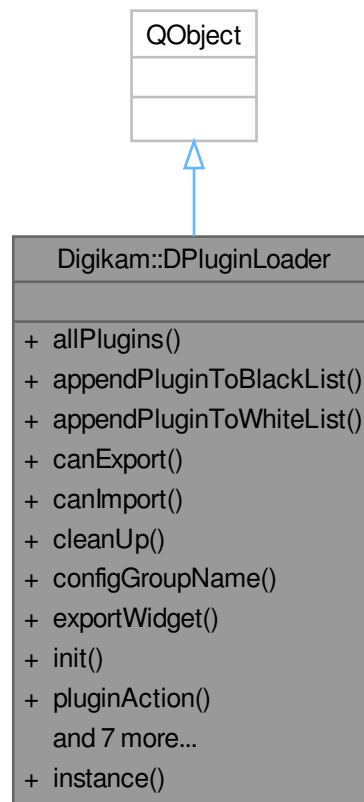
Holds whether the plugin can be seen in parent view.

Implements [Digikam::DPlugin](#).

## 6.486 Digikam::DPluginLoader Class Reference

The class that handles digiKam's external plugins.

Inheritance diagram for Digikam::DPluginLoader:



## Classes

- class [Private](#)

## Public Member Functions

- `QList< DPlugin * > allPlugins ()` const  
*Returns all available plugins.*
- void `appendPluginToBlackList (const QString &filename)`  
*appendPluginToBlackList Prevent that a plugin is loaded from the given filename*
- void `appendPluginToWhiteList (const QString &filename)`  
*appendPluginToWhiteList Add a plugin to the whitelist of tools. If the whitelist is not empty, only whitelisted tools are loaded. If a tool is both whitelisted and blacklisted, it will not be loaded.*
- bool `canExport (const QString &format)` const
- bool `canImport (const QString &format)` const
- void `cleanUp ()`
- `QString configGroupName ()` const
- `DImgLoaderSettings * exportWidget (const QString &format)` const
- void `init ()`

- [DPluginAction](#) \* **pluginAction** (const QString &actionName, QObject \*const parent) const  
*Returns the plugin action corresponding to a action name for a given parent. If not found, this returns a null pointer.*
- QList< [DPluginAction](#) \* > **pluginActions** (const QString &pluginIID, QObject \*const parent) const  
*Returns the plugin actions corresponding to a plugin internal ID string for a given parent. If not found, this returns an empty list.*
- QList< [DPluginAction](#) \* > **pluginsActions** ([DPluginAction::ActionCategory](#) cat, QObject \*const parent) const  
*Returns a list of plugin actions set as category for a given parent. If no plugin have found in this category, this returns an empty list.*
- QList< [DPluginAction](#) \* > **pluginsActions** ([DPluginAction::ActionType](#) type, QObject \*const parent) const  
*Returns a list of plugin actions set as type for a given parent. If no plugin have found in this category, this returns an empty list.*
- QString **pluginXmlSections** ([DPluginAction::ActionCategory](#) cat, QObject \*const parent) const  
*Returns all xml sections as string of plugin actions set with a kind of category for a given parent.*
- void [registerEditorPlugins](#) (QObject \*const parent)
- void [registerGenericPlugins](#) (QObject \*const parent)
- void [registerRawImportPlugins](#) (QObject \*const parent)

### Static Public Member Functions

- static [DPluginLoader](#) \* [instance](#) ()  
*instance: returns the singleton of plugin loader*

### Friends

- class [DPluginLoaderCreator](#)

## 6.486.1 Detailed Description

Ownership policy for plugins:

The [DPluginLoader](#) creates new objects and transfer ownership. In order to create the objects, the [DPluginLoader](#) internally has a list of the tools which are owned by the [DPluginLoader](#) and destroyed by it.

## 6.486.2 Member Function Documentation

### 6.486.2.1 appendPluginToBlackList()

```
void Digikam::DPluginLoader::appendPluginToBlackList (
    const QString & filename )
```

#### Parameters

<i>filename</i>	The name of the file excluding file extension to blacklist. E.g. to ignore "HtmlGalleryPlugin.so" on Linux and "HtmlGalleryPlugin.dll" on Windows, pass "HtmlGalleryPlugin"
-----------------	---



### 6.486.2.2 appendPluginToWhiteList()

```
void Digikam::DPluginLoader::appendPluginToWhiteList (
    const QString & filename )
```

#### Parameters

<i>filename</i>	The name of the file excluding file extension to whitelist. E.g. to not ignore "HtmlGalleryPlugin.so" on Linux and "HtmlGalleryPlugin.dll" on Windows, pass "HtmlGalleryPlugin"
-----------------	---

### 6.486.2.3 canExport()

```
bool Digikam::DPluginLoader::canExport (
    const QString & format ) const
```

Return true if format is supported by a [DPluginDImg](#) to export image.

### 6.486.2.4 canImport()

```
bool Digikam::DPluginLoader::canImport (
    const QString & format ) const
```

Return true if format is supported by a [DPluginDImg](#) to import image.

### 6.486.2.5 cleanUp()

```
void Digikam::DPluginLoader::cleanUp ( )
```

Unload all loaded plugins. Call this method before the main instance is closed.

### 6.486.2.6 configGroupName()

```
QString Digikam::DPluginLoader::configGroupName ( ) const
```

Return the config group name used to store the list of plugins to load at startup.

### 6.486.2.7 exportWidget()

```
DImgLoaderSettings * Digikam::DPluginLoader::exportWidget (
    const QString & format ) const
```

Return a new widget instance from a [DPluginDImg](#) to show settings while exporting image to specified format. Return nullptr if format is not supported or if no settings widget is available for this format.

### 6.486.2.8 init()

```
void Digikam::DPluginLoader::init ( )
```

Init plugin loader. Call this method to parse and load relevant plugins installed on your system.

### 6.486.2.9 instance()

```
DPluginLoader * Digikam::DPluginLoader::instance ( ) [static]
```

#### Returns

[DPluginLoader](#) global instance

### 6.486.2.10 registerEditorPlugins()

```
void Digikam::DPluginLoader::registerEditorPlugins (
    QObject *const parent )
```

Register all Editor plugin actions to parent object.

### 6.486.2.11 registerGenericPlugins()

```
void Digikam::DPluginLoader::registerGenericPlugins (
    QObject *const parent )
```

Register all Generic plugin actions to parent object.

### 6.486.2.12 registerRawImportPlugins()

```
void Digikam::DPluginLoader::registerRawImportPlugins (
    QObject *const parent )
```

Register all Raw Import plugin to parent object.

## 6.487 Digikam::DPluginLoader::Private Class Reference

### Public Member Functions

- bool [appendPlugin](#) (QObject \*const obj, QPluginLoader \*const loader)
- void [loadPlugins](#) ()
- QFileInfoList [pluginEntriesList](#) () const

## Public Attributes

- QList< [DPlugin](#) \* > **allPlugins**
- QStringList **blacklist**
- QStringList **DKBlacklist**  
*Showfoto specific plugins to ignore in digiKam.*
- bool **pluginsLoaded** = false
- QStringList **whitelist**

## 6.487.1 Member Function Documentation

### 6.487.1.1 `appendPlugin()`

```
bool Digikam::DPluginLoader::Private::appendPlugin (
    QObject *const obj,
    QPluginLoader *const loader )
```

Append object to the given plugins list.

### 6.487.1.2 `loadPlugins()`

```
void Digikam::DPluginLoader::Private::loadPlugins ( )
```

Stage to load plugins in memory

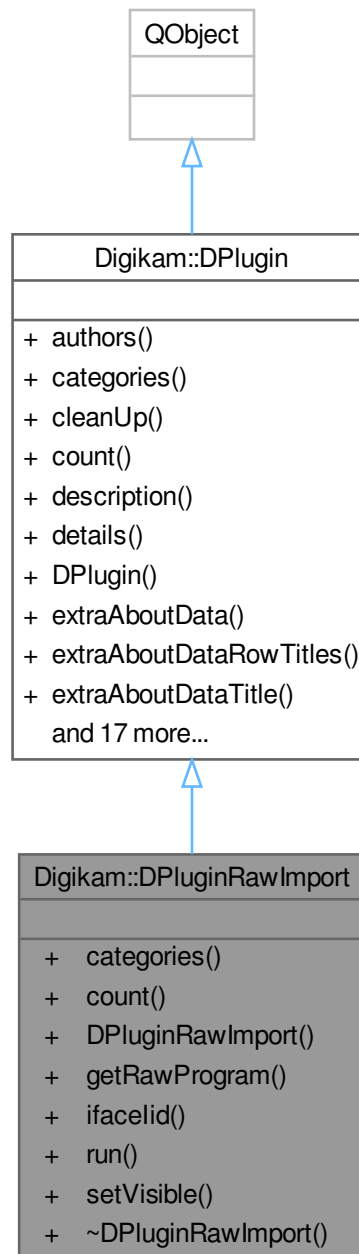
### 6.487.1.3 `pluginEntriesList()`

```
QFileInfoList Digikam::DPluginLoader::Private::pluginEntriesList ( ) const
```

Try to find plugin files from Qt5 plugins install dir:

## 6.488 Digikam::DPluginRawImport Class Reference

Inheritance diagram for Digikam::DPluginRawImport:



### Signals

- void [signalDecodedImage](#) (const [Digikam::LoadingDescription](#) &, const [Digikam::DImg](#) &)
- void [signalLoadRaw](#) (const [Digikam::LoadingDescription](#) &)

## Public Member Functions

- QStringList [categories](#) () const override
- int [count](#) () const override
- [DPluginRawImport](#) (QObject \*const parent=nullptr)
- virtual QString [getRawProgram](#) () const
- QString [ifacelid](#) () const override
- virtual bool [run](#) (const QString &path, const [DRawDecoding](#) &def)=0
- void [setVisible](#) (bool) override
- [~DPluginRawImport](#) () override=default

## Public Member Functions inherited from [Digikam::DPlugin](#)

- virtual QList< [DPluginAuthor](#) > [authors](#) () const =0  
*Returns authors list for the plugin.*
- virtual void [cleanUp](#) ()
- virtual QString [description](#) () const =0  
*Returns a short description about the plugin.*
- virtual QString [details](#) () const =0  
*Returns a long description about the plugin.*
- [DPlugin](#) (QObject \*const parent=nullptr)
- virtual QMap< QString, QStringList > [extraAboutData](#) () const  
*Returns a map of extra data to show in plugin about dialog.*
- virtual QStringList [extraAboutDataRowTitles](#) () const  
*Returns a list of extra data row titles to show in tab of plugin about dialog.*
- virtual QString [extraAboutDataTitle](#) () const  
*Returns the tab title of data returned by [extraAboutData\(\)](#).*
- virtual QString [handbookChapter](#) () const
- virtual QString [handbookReference](#) () const
- virtual QString [handbookSection](#) () const
- virtual bool [hasVisibilityProperty](#) () const
- virtual QIcon [icon](#) () const  
*Returns an icon for the plugin. Default implementation return the system plugin icon.*
- virtual QString [iid](#) () const =0  
*Returns the unique internal identification property of the plugin. Must be formatted as "org.kde.digikam.plugin.\_PLUGIN\_TYPE\_.NAME\_OF\_PLUGIN\_". Examples: "org.kde.digikam.plugin.generic.Calendar" "org.kde.digikam.plugin.editor.AdjustCurvesTool" "org.kde.digikam.plugin.bqm.NoiseReduction".*
- QString [libraryFileName](#) () const  
*Returns the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- virtual QString [name](#) () const =0  
*Returns the user-visible name of the plugin.*
- QStringList [pluginAuthors](#) () const
- void [setLibraryFileName](#) (const QString &)  
*Sets the file name of the library for this plugin. This string is filled at run-time by plugin loader.*
- void [setShouldLoaded](#) (bool b)
- virtual void [setup](#) (QObject \*const parent)=0
- bool [shouldLoaded](#) () const
- QString [version](#) () const
- [~DPlugin](#) () override

## 6.488.1 Constructor & Destructor Documentation

### 6.488.1.1 DPluginRawImport()

```
Digikam::DPluginRawImport::DPluginRawImport (
    QObject *const parent = nullptr ) [explicit]
```

Constructor with optional parent object

### 6.488.1.2 ~DPluginRawImport()

```
Digikam::DPluginRawImport::~~DPluginRawImport ( ) [override], [default]
```

Destructor

## 6.488.2 Member Function Documentation

### 6.488.2.1 categories()

```
QStringList Digikam::DPluginRawImport::categories ( ) const [inline], [override], [virtual]
```

This kind of plugin do not use a category.

Implements [Digikam::DPlugin](#).

### 6.488.2.2 count()

```
int Digikam::DPluginRawImport::count ( ) const [inline], [override], [virtual]
```

This kind of plugin only provide one tool.

Implements [Digikam::DPlugin](#).

### 6.488.2.3 getRawProgram()

```
QString Digikam::DPluginRawImport::getRawProgram ( ) const [virtual]
```

Return the path to the raw program, or empty if not found.

### 6.488.2.4 ifaceId()

```
QString Digikam::DPluginRawImport::ifaceId ( ) const [inline], [override], [virtual]
```

Return the plugin interface identifier.

Implements [Digikam::DPlugin](#).

**6.488.2.5 run()**

```
virtual bool Digikam::DPluginRawImport::run (
    const QString & path,
    const DRawDecoding & def ) [pure virtual]
```

Function to re-implement used to invoke Raw processor for a Raw file path and a Default Raw decoding settings.

**6.488.2.6 setVisible()**

```
void Digikam::DPluginRawImport::setVisible (
    bool ) [inline], [override], [virtual]
```

This kind of plugin do not have GUI visibility attribute.

Implements [Digikam::DPlugin](#).

**6.488.2.7 signalDecodedImage**

```
void Digikam::DPluginRawImport::signalDecodedImage (
    const Digikam::LoadingDescription & ,
    const Digikam::DImg & ) [signal]
```

Signal emitted to notify host application to load pre-decoded Raw preprocessed with these decoding settings.

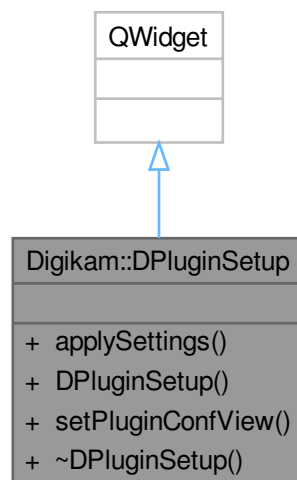
**6.488.2.8 signalLoadRaw**

```
void Digikam::DPluginRawImport::signalLoadRaw (
    const Digikam::LoadingDescription & ) [signal]
```

Signal emitted to notify host application to load Raw with these decoding settings.

**6.489 Digikam::DPluginSetup Class Reference**

Inheritance diagram for Digikam::DPluginSetup:



### Public Member Functions

- void **applySettings** ()
- **DPluginSetup** (QWidget \*const parent=nullptr)
- void **setPluginConfView** (DPluginConfView \*const view)

## 6.490 Digikam::DPointSelect Class Reference

Inheritance diagram for Digikam::DPointSelect:





## Signals

- void `valueChanged` (int x, int y)

## Public Member Functions

- QRect `contentsRect` () const
- `DPointSelect` (QWidget \*const parent)
- QSize `minimumSizeHint` () const override
- void `setMarkerColor` (const QColor &col)
- void `setRange` (int minX, int minY, int maxX, int maxY)
- void `setValues` (int xPos, int yPos)
- void `setXValue` (int xPos)
- void `setYValue` (int yPos)
- int `xValue` () const
- int `yValue` () const

## Protected Member Functions

- virtual void `drawContents` (QPainter \*)
- virtual void `drawMarker` (QPainter \*p, int xp, int yp)
- void `mouseMoveEvent` (QMouseEvent \*e) override
- void `mousePressEvent` (QMouseEvent \*e) override
- void `paintEvent` (QPaintEvent \*e) override
- void `valuesFromPosition` (int x, int y, int &xVal, int &yVal) const
- void `wheelEvent` (QWheelEvent \*) override

## Properties

- int `xValue`
- int `yValue`

## Friends

- class `Private`

## 6.490.1 Constructor & Destructor Documentation

### 6.490.1.1 DPointSelect()

```
Digikam::DPointSelect::DPointSelect (
    QWidget *const parent ) [explicit]
```

Constructs a two-dimensional selector widget which has a value range of [0..100] in both directions.

## 6.490.2 Member Function Documentation

### 6.490.2.1 contentsRect()

```
QRect Digikam::DPointSelect::contentsRect ( ) const
```

#### Returns

the rectangle on which subclasses should draw.

### 6.490.2.2 drawContents()

```
virtual void Digikam::DPointSelect::drawContents (
    QPainter * ) [inline], [protected], [virtual]
```

Override this function to draw the contents of the widget. The default implementation does nothing.

Draw within [contentsRect\(\)](#) only.

Reimplemented in [Digikam::DHueSaturationSelector](#).

### 6.490.2.3 drawMarker()

```
void Digikam::DPointSelect::drawMarker (
    QPainter * p,
    int xp,
    int yp ) [protected], [virtual]
```

Override this function to draw the marker which indicates the currently selected value pair.

### 6.490.2.4 minimumSizeHint()

```
QSize Digikam::DPointSelect::minimumSizeHint ( ) const [override]
```

Reimplemented to give the widget a minimum size

### 6.490.2.5 setMarkerColor()

```
void Digikam::DPointSelect::setMarkerColor (
    const QColor & col )
```

Sets the color used to draw the marker

#### Parameters

<i>col</i>	the color
------------	-----------

### 6.490.2.6 setRange()

```
void Digikam::DPointSelect::setRange (
    int minX,
    int minY,
    int maxX,
    int maxY )
```

Sets the range of possible values.

### 6.490.2.7 setValues()

```
void Digikam::DPointSelect::setValues (
    int xPos,
    int yPos )
```

Sets the current values in horizontal and vertical direction.

#### Parameters

<i>xPos</i>	the horizontal value
<i>yPos</i>	the vertical value

### 6.490.2.8 setXValue()

```
void Digikam::DPointSelect::setXValue (
    int xPos )
```

Sets the current horizontal value

#### Parameters

<i>xPos</i>	the horizontal value
-------------	----------------------

### 6.490.2.9 setYValue()

```
void Digikam::DPointSelect::setYValue (
    int yPos )
```

Sets the current vertical value

#### Parameters

<i>yPos</i>	the vertical value
-------------	--------------------

#### 6.490.2.10 valueChanged

```
void Digikam::DPointSelect::valueChanged (
    int x,
    int y ) [signal]
```

This signal is emitted whenever the user chooses a value, e.g. by clicking with the mouse on the widget.

#### 6.490.2.11 valuesFromPosition()

```
void Digikam::DPointSelect::valuesFromPosition (
    int x,
    int y,
    int & xVal,
    int & yVal ) const [protected]
```

Converts a pixel position to its corresponding values.

#### 6.490.2.12 xValue()

```
int Digikam::DPointSelect::xValue ( ) const
```

##### Returns

the current value in horizontal direction.

#### 6.490.2.13 yValue()

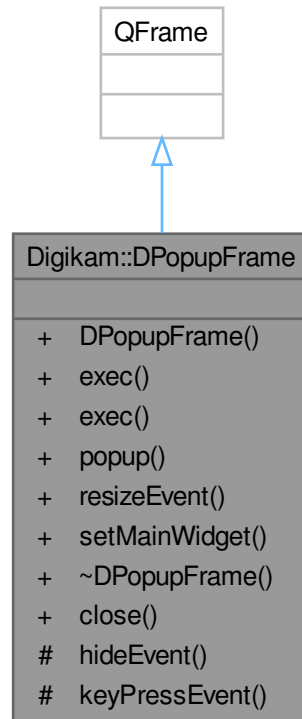
```
int Digikam::DPointSelect::yValue ( ) const
```

##### Returns

the current value in vertical direction.

## 6.491 Digikam::DPopupFrame Class Reference

Inheritance diagram for Digikam::DPopupFrame:



### Public Slots

- void `close` (int r)

### Signals

- void `leaveModality` ()

### Public Member Functions

- `DPopupFrame` (QWidget \*const parent=nullptr)
- int `exec` (const QPoint &p)
- int `exec` (int x, int y)
- void `popup` (const QPoint &p)
- void `resizeEvent` (QResizeEvent \*e) override
- void `setMainWidget` (QWidget \*const m)
- `~DPopupFrame` () override

## Protected Member Functions

- void [hideEvent](#) (QHideEvent \*e) override
- void [keyPressEvent](#) (QKeyEvent \*e) override

## Friends

- class **Private**

## 6.491.1 Constructor & Destructor Documentation

### 6.491.1.1 DPopupFrame()

```
Digikam::DPopupFrame::DPopupFrame (
    QWidget *const parent = nullptr ) [explicit]
```

The constructor. Creates a dialog without buttons.

### 6.491.1.2 ~DPopupFrame()

```
Digikam::DPopupFrame::~~DPopupFrame ( ) [override]
```

The destructor

## 6.491.2 Member Function Documentation

### 6.491.2.1 close

```
void Digikam::DPopupFrame::close (
    int r ) [slot]
```

Close the popup window. This is called from the main widget, usually. *r* is the result returned from [exec\(\)](#).

### 6.491.2.2 exec() [1/2]

```
int Digikam::DPopupFrame::exec (
    const QPoint & p )
```

Execute the popup window.

### 6.491.2.3 exec() [2/2]

```
int Digikam::DPopupFrame::exec (
    int x,
    int y )
```

Execute the popup window.

#### 6.491.2.4 hideEvent()

```
void Digikam::DPopupFrame::hideEvent (
    QHideEvent * e ) [override], [protected]
```

Catch hide events.

#### 6.491.2.5 keyPressEvent()

```
void Digikam::DPopupFrame::keyPressEvent (
    QKeyEvent * e ) [override], [protected]
```

Catch key press events.

#### 6.491.2.6 popup()

```
void Digikam::DPopupFrame::popup (
    const QPoint & p )
```

Open the popup window at position pos.

#### 6.491.2.7 resizeEvent()

```
void Digikam::DPopupFrame::resizeEvent (
    QResizeEvent * e ) [override]
```

The resize event. Simply resizes the main widget to the whole widgets client size.

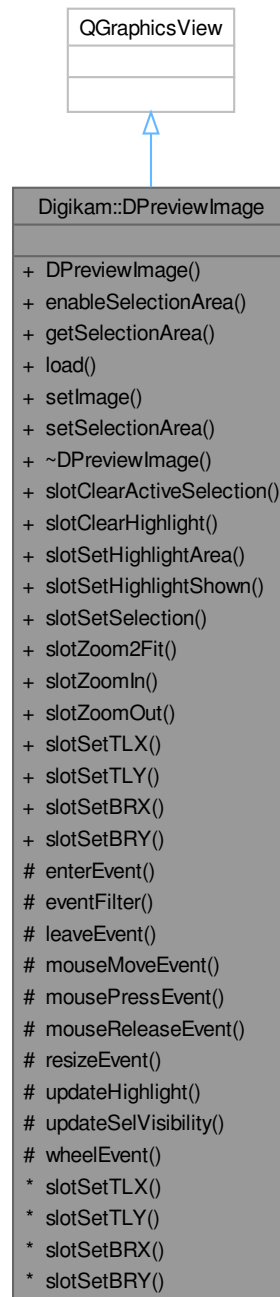
#### 6.491.2.8 setMainWidget()

```
void Digikam::DPopupFrame::setMainWidget (
    QWidget *const m )
```

Set the main widget. You cannot set the main widget from the constructor, since it must be a child of the frame itself. Be careful: the size is set to the main widgets size. It is up to you to set the main widgets correct size before setting it as the main widget.

## 6.492 Digikam::DPreviewImage Class Reference

Inheritance diagram for Digikam::DPreviewImage:



### Public Slots

- void `slotClearActiveSelection ()`
- void `slotClearHighlight ()`



- void [slotSetHighlightArea](#) (float tl\_x, float tl\_y, float br\_x, float br\_y)
  - void [slotSetHighlightShown](#) (int percentage, const QColor &highLightColor=Qt::white)
  - void [slotSetSelection](#) (float tl\_x, float tl\_y, float br\_x, float br\_y)
  - void **slotZoom2Fit** ()
  - void **slotZoomIn** ()
  - void **slotZoomOut** ()
- 
- void **slotSetTLX** (float ratio)
    - Selection area specific slots (TL = TopLeft, BR = BottomRight)*
  - void **slotSetTLY** (float ratio)
  - void **slotSetBRX** (float ratio)
  - void **slotSetBRY** (float ratio)

### Public Member Functions

- **DPreviewImage** (QWidget \*const parent)
- void **enableSelectionArea** (bool b)
- QRectF **getSelectionArea** () const
- bool **load** (const QUrl &file) const
- bool **setImage** (const QImage &img) const
- void [setSelectionArea](#) (const QRectF &rectangle)

### Protected Member Functions

- void **enterEvent** (QEnterEvent \*) override
- bool **eventFilter** (QObject \*, QEvent \*) override
- void **leaveEvent** (QEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override
- void **updateHighlight** ()
- void **updateSelVisibility** ()
- void **wheelEvent** (QWheelEvent \*) override

## 6.492.1 Member Function Documentation

### 6.492.1.1 setSelectionArea()

```
void Digikam::DPreviewImage::setSelectionArea (
    const QRectF & rectangle )
```

Sets a selection area and show it

#### Parameters

<i>rectangle</i>	This rectangle should have height and width of 1.0
------------------	--

### 6.492.1.2 slotClearHighlight

```
void Digikam::DPreviewImage::slotClearHighlight ( ) [slot]
```

This function removes the highlight area.

### 6.492.1.3 slotSetHighlightArea

```
void Digikam::DPreviewImage::slotSetHighlightArea (
    float tl_x,
    float tl_y,
    float br_x,
    float br_y ) [slot]
```

This function is used to darken everything except what is inside the given area.

#### Note

all parameters must be in the range 0.0 -> 1.0.

#### Parameters

<i>tl</i> <sub>↔</sub> <i>_x</i>	is the x coordinate of the top left corner 0=0 1=image with.
<i>tl</i> <sub>↔</sub> <i>_y</i>	is the y coordinate of the top left corner 0=0 1=image height.
<i>br</i> <sub>↔</sub> <i>_x</i>	is the x coordinate of the bottom right corner 0=0 1=image with.
<i>br</i> <sub>↔</sub> <i>_y</i>	is the y coordinate of the bottom right corner 0=0 1=image height.

### 6.492.1.4 slotSetHighlightShown

```
void Digikam::DPreviewImage::slotSetHighlightShown (
    int percentage,
    const QColor & highlightColor = Qt::white ) [slot]
```

This function sets the percentage of the highlighted area that is visible. The rest is hidden. This stacks with the previous highlight area.

#### Parameters

<i>percentage</i>	is the percentage of the highlighted area that is shown.
<i>highlightColor</i>	is the color to use to hide the highlighted area of the image.

### 6.492.1.5 slotSetSelection

```
void Digikam::DPreviewImage::slotSetSelection (
    float tl_x,
```

```
float tl_y,  
float br_x,  
float br_y ) [slot]
```

This function is used to set a selection without the user setting it.

#### Note

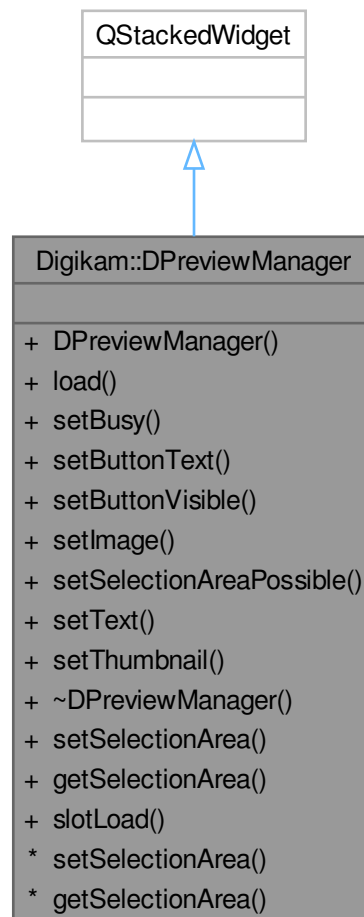
all parameters must be in the range 0.0 -> 1.0.

#### Parameters

<i>tl</i> <sub>↔</sub> <i>_x</i>	is the x coordinate of the top left corner 0=0 1=image width.
<i>tl</i> <sub>↔</sub> <i>_y</i>	is the y coordinate of the top left corner 0=0 1=image height.
<i>br</i> <sub>↔</sub> <i>_x</i>	is the x coordinate of the bottom right corner 0=0 1=image width.
<i>br</i> <sub>↔</sub> <i>_y</i>	is the y coordinate of the bottom right corner 0=0 1=image height.

## 6.493 Digikam::DPreviewManager Class Reference

Inheritance diagram for Digikam::DPreviewManager:



### Public Types

- enum `DisplayMode` { `MessageMode = 0` , `PreviewMode` }

### Public Slots

- void `slotLoad` (const `QUrl` &url)

### Signals

- void `signalButtonClicked` ()

## Public Member Functions

- **DPreviewManager** (QWidget \*const parent)
  - bool **load** (const QUrl &file, bool fit=true)
  - void **setBusy** (bool b, const QString &text=QString())
  - void **setButtonText** (const QString &text)
  - void **setButtonVisible** (bool b)
  - void **setImage** (const QImage &img, bool fit=true)
  - void **setSelectionAreaPossible** (bool b)
  - void **setText** (const QString &text, const QColor &color=Qt::white)
  - void **setThumbnail** (const QPixmap &preview=QPixmap())
- 
- void [setSelectionArea](#) (const QRectF &rectangle)
  - QRectF **getSelectionArea** () const

## 6.493.1 Member Function Documentation

### 6.493.1.1 setSelectionArea()

```
void Digikam::DPreviewManager::setSelectionArea (  
    const QRectF & rectangle )
```

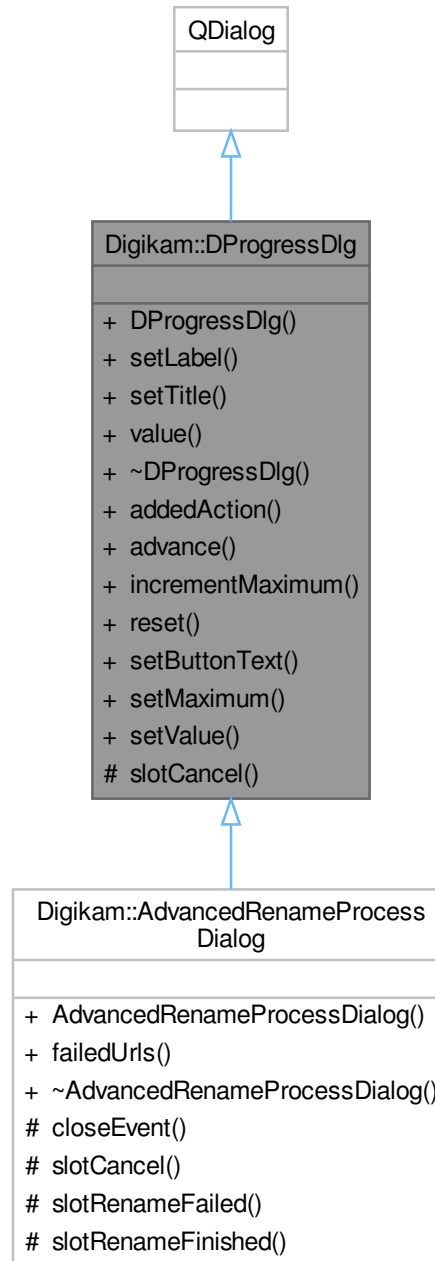
Manage a selection area and show it

#### Parameters

<i>rectangle</i>	This rectangle should have height and width of 1.0
------------------	--

## 6.494 Digikam::DProgressDlg Class Reference

Inheritance diagram for Digikam::DProgressDlg:



### Public Slots

- void **addedAction** (const QPixmap &icon, const QString &text)
- void **advance** (int offset)

- void **incrementMaximum** (int added)
- void **reset** ()
- void **setButtonText** (const QString &text)
- void **setMaximum** (int max)
- void **setValue** (int value)

### Signals

- void **signalCancelPressed** ()

### Public Member Functions

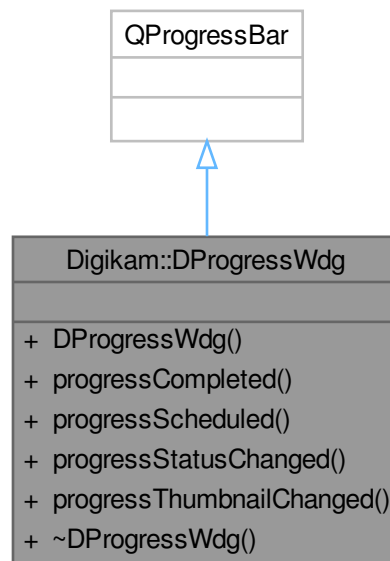
- **DProgressDlg** (QWidget \*const parent=nullptr, const QString &caption=QString())
- void **setLabel** (const QString &text)
- void **setTitle** (const QString &text)
- int **value** () const

### Protected Slots

- void **slotCancel** ()

## 6.495 Digikam::DProgressWdg Class Reference

Inheritance diagram for Digikam::DProgressWdg:



## Signals

- void [signalProgressCanceled](#) ()

## Public Member Functions

- **DProgressWdg** (QWidget \*const parent)
- void [progressCompleted](#) ()
- void [progressScheduled](#) (const QString &title, bool canBeCanceled, bool hasThumb)
- void [progressStatusChanged](#) (const QString &status)
- void [progressThumbnailChanged](#) (const QPixmap &thumb)

## 6.495.1 Member Function Documentation

### 6.495.1.1 progressCompleted()

```
void Digikam::DProgressWdg::progressCompleted ( )
```

Call this method to query progress manager that process is done.

### 6.495.1.2 progressScheduled()

```
void Digikam::DProgressWdg::progressScheduled (
    const QString & title,
    bool canBeCanceled,
    bool hasThumb )
```

Call this method to start a new instance of progress notification into progress manager You can pass title string to name progress item, and set it as cancelable. In this case, [signalProgressCanceled\(\)](#) is fired when user press cancel button from progress manager. This item can also accept a thumbnail that you can change through `progressThumbnailChanged()`.

### 6.495.1.3 progressStatusChanged()

```
void Digikam::DProgressWdg::progressStatusChanged (
    const QString & status )
```

Change status string in progress manager

### 6.495.1.4 progressThumbnailChanged()

```
void Digikam::DProgressWdg::progressThumbnailChanged (
    const QPixmap & thumb )
```

Change thumbnail in progress manager



6.495.1.5 signalProgressCanceled

```
void Digikam::DProgressWdg::signalProgressCanceled ( ) [signal]
```

Fired when user press cancel button from progress manager.

## 6.496 Digikam::DragDropModelImplementation Class Reference

Inheritance diagram for Digikam::DragDropModelImplementation:



## Public Member Functions

- virtual Qt::ItemFlags [dragDropFlags](#) (const QModelIndex &index) const
- Qt::ItemFlags [dragDropFlagsV2](#) (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const
- [DragDropModelImplementation](#) ()=default
- bool **dropMimeData** (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool **isDragEnabled** (const QModelIndex &index) const
- virtual bool **isDropEnabled** (const QModelIndex &index) const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const
- QStringList **mimeTypes** () const
- void [setDragDropHandler](#) ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions [supportedDropActions](#) () const

## Protected Attributes

- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr

## 6.496.1 Constructor & Destructor Documentation

### 6.496.1.1 DragDropModelImplementation()

```
Digikam::DragDropModelImplementation::DragDropModelImplementation ( ) [default]
```

A class providing a sample implementation for a QAbstractItemModel redirecting drag-and-drop support to a handler. Include the macro `DECLARE_Model_DRAG_DROP_METHODS` in your derived QAbstractItemModel class.

## 6.496.2 Member Function Documentation

### 6.496.2.1 dragDropFlags()

```
Qt::ItemFlags Digikam::DragDropModelImplementation::dragDropFlags (
    const QModelIndex & index ) const [virtual]
```

Call from your flags() method, adding the relevant drag drop flags. Default implementation enables both drag and drop on the index if a drag drop handler is set. Reimplement to fine-tune. Note: There is an alternative below.

### 6.496.2.2 dragDropFlagsV2()

```
Qt::ItemFlags Digikam::DragDropModelImplementation::dragDropFlagsV2 (
    const QModelIndex & index ) const
```

This is an alternative approach to [dragDropFlags\(\)](#). [dragDropFlagsV2](#) calls the virtual methods [isDragEnabled\(\)](#) and [isDropEnabled\(\)](#) which you then reimplement. Use simple [dragDropFlags\(\)](#) if you need not customization, or reimplement [dragDropFlags\(\)](#) if you fine-tune it yourself.

### 6.496.2.3 setDragDropHandler()

```
void Digikam::DragDropModelImplementation::setDragDropHandler (
    AbstractItemDragDropHandler * handler )
```

Set a drag drop handler.

### 6.496.2.4 supportedDropActions()

```
Qt::DropActions Digikam::DragDropModelImplementation::supportedDropActions ( ) const
```

Implements the relevant QAbstractItemModel methods for drag and drop. All functionality is redirected to the handler. dropMimeData() always returns false, leaving implementation to the view.



## Protected Member Functions

- virtual `QAbstractItemView * asView ()=0`
- bool **decodelsCutSelection** (const `QMimeData *mimeData`)
- virtual `AbstractItemDragDropHandler * dragDropHandler () const =0`
- void **dragEnterEvent** (`QDragEnterEvent *event`)
- void **dragMoveEvent** (`QDragMoveEvent *e`)
- void **dropEvent** (`QDropEvent *e`)
- void **encodelsCutSelection** (`QMimeData *mime`, bool `isCutSelection`)
- virtual `QModelIndex mapIndexForDragDrop` (const `QModelIndex &index`) const =0
- virtual `QPixmap pixmapForDrag` (const `QList< QModelIndex > &indexes`) const =0
- void **startDrag** (`Qt::DropActions supportedActions`)

## 6.497.1 Member Function Documentation

### 6.497.1.1 asView()

```
virtual QAbstractItemView * Digikam::DragDropViewImplementation::asView ( ) [protected], [pure virtual]
```

This one is implemented by `DECLARE_VIEW_DRAG_DROP_METHODS`.

### 6.497.1.2 dragDropHandler()

```
virtual AbstractItemDragDropHandler * Digikam::DragDropViewImplementation::dragDropHandler ( ) const [protected], [pure virtual]
```

You need to implement these three methods Returns the drag drop handler.

Implemented in [Digikam::ItemCategorizedView](#), [Digikam::TableViewTreeView](#), [Digikam::VersionsTreeView](#), [ShowFoto::ShowfotoCategorizedView](#), and [Digikam::ImportCategorizedView](#).

### 6.497.1.3 dragEnterEvent()

```
void Digikam::DragDropViewImplementation::dragEnterEvent ( QDragEnterEvent * event ) [protected]
```

Implements the relevant `QAbstractItemView` methods for drag and drop.

### 6.497.1.4 mapIndexForDragDrop()

```
virtual QModelIndex Digikam::DragDropViewImplementation::mapIndexForDragDrop ( const QModelIndex & index ) const [protected], [pure virtual]
```

Maps the given index of the view's model to an index of the handler's model, which can be a source model of the view's model.

Implemented in [Digikam::TableViewTreeView](#), [Digikam::VersionsTreeView](#), and [Digikam::ItemViewCategorized](#).

### 6.497.1.5 pixmapForDrag()

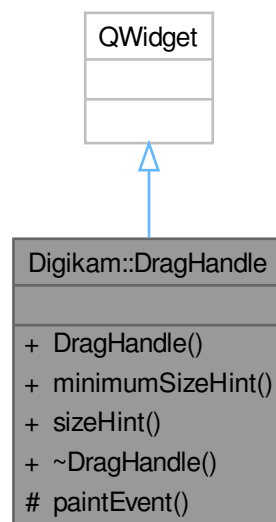
```
virtual QPixmap Digikam::DragDropViewImplementation::pixmapForDrag (
    const QList< QModelIndex > & indexes ) const [protected], [pure virtual]
```

Creates a pixmap for dragging the given indexes.

Implemented in [Digikam::TableViewTreeView](#), [Digikam::VersionsTreeView](#), and [Digikam::ItemViewCategorized](#).

## 6.498 Digikam::DragHandle Class Reference

Inheritance diagram for Digikam::DragHandle:



### Public Member Functions

- **DragHandle** (QDockWidget \*const)
- QSize **minimumSizeHint** () const override
- QSize **sizeHint** () const override

### Protected Member Functions

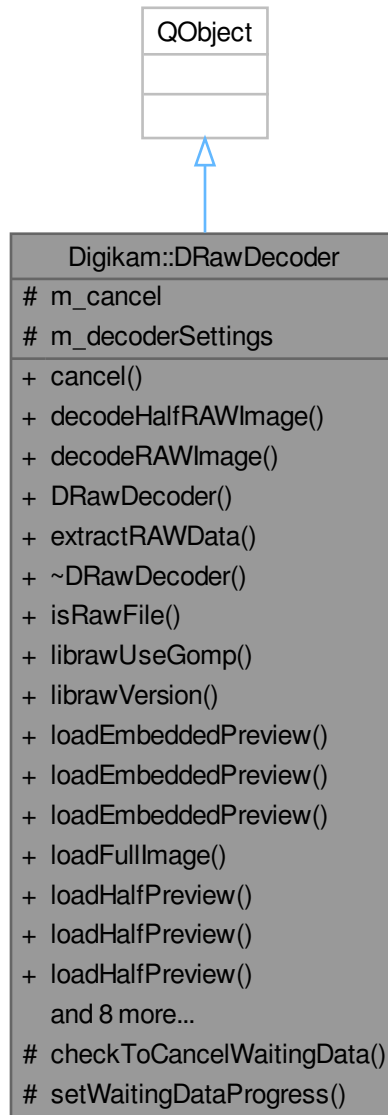
- void **paintEvent** (QPaintEvent \*) override

### 6.498.1 Detailed Description

An alternative handle for QDockWidget's that looks like a toolbar handle.

## 6.499 Digikam::DRawDecoder Class Reference

Inheritance diagram for Digikam::DRawDecoder:



### Classes

- class [Private](#)

### Public Member Functions

- void [cancel](#) ()

- bool [decodeHalfRAWImage](#) (const QString &filePath, const [DRawDecoderSettings](#) &[DRawDecoderSettings](#), QByteArray &imageData, int &width, int &height, int &rgbmax)
- bool [decodeRAWImage](#) (const QString &filePath, const [DRawDecoderSettings](#) &[DRawDecoderSettings](#), QByteArray &imageData, int &width, int &height, int &rgbmax)
- [DRawDecoder](#) ()
- bool [extractRAWData](#) (const QString &filePath, QByteArray &rawData, [DRawInfo](#) &identify, unsigned int shotSelect=0)
- [~DRawDecoder](#) () override

### Static Public Member Functions

- static bool [isRawFile](#) (const QUrl &url)
- static int [librawUseGomp](#) ()
- static QString [librawVersion](#) ()
- static bool [loadEmbeddedPreview](#) (QByteArray &imgData, const QBuffer &inBuffer)
- static bool [loadEmbeddedPreview](#) (QByteArray &imgData, const QString &path)
- static bool [loadEmbeddedPreview](#) (QImage &image, const QString &path)
- static bool [loadFullImage](#) (QImage &image, const QString &path, const [DRawDecoderSettings](#) &settings=[DRawDecoderSettings](#)())
- static bool [loadHalfPreview](#) (QByteArray &imgData, const QBuffer &inBuffer)
- static bool [loadHalfPreview](#) (QByteArray &imgData, const QString &path)
- static bool [loadHalfPreview](#) (QImage &image, const QString &path, bool rotate=true)
- static bool [loadRawPreview](#) (QByteArray &imgData, const QBuffer &inBuffer)
- static bool [loadRawPreview](#) (QByteArray &imgData, const QString &path)
- static bool [loadRawPreview](#) (QImage &image, const QString &path)
- static bool [rawFileIdentify](#) ([DRawInfo](#) &identify, const QString &path)
- static QString [rawFiles](#) ()
- static QStringList [rawFilesList](#) ()
- static int [rawFilesVersion](#) ()
- static QStringList [supportedCamera](#) ()

### Protected Member Functions

- virtual bool [checkToCancelWaitingData](#) ()
- virtual void [setWaitingDataProgress](#) (double value)

### Protected Attributes

- bool [m\\_cancel](#) = false
- [DRawDecoderSettings](#) [m\\_decoderSettings](#)

### Friends

- class [Private](#)

## 6.499.1 Constructor & Destructor Documentation

### 6.499.1.1 DRawDecoder()

`Digikam::DRawDecoder::DRawDecoder ( )`

Standard constructor.



### 6.499.1.2 ~DRawDecoder()

```
Digikam::DRawDecoder::~~DRawDecoder ( ) [override]
```

Standard destructor.

## 6.499.2 Member Function Documentation

### 6.499.2.1 cancel()

```
void Digikam::DRawDecoder::cancel ( )
```

To cancel 'decodeHalfRAWImage' and 'decodeRAWImage' methods running in a separate thread.

### 6.499.2.2 checkToCancelWaitingData()

```
bool Digikam::DRawDecoder::checkToCancelWaitingData ( ) [protected], [virtual]
```

Re-implement this method to control the cancelisation of loop which wait data from RAW decoding process with your proper environment. By default, this method check if `m_cancel` is true.

### 6.499.2.3 decodeHalfRAWImage()

```
bool Digikam::DRawDecoder::decodeHalfRAWImage (
    const QString & filePath,
    const DRawDecoderSettings & DRawDecoderSettings,
    QByteArray & imageData,
    int & width,
    int & height,
    int & rgbmax )
```

Extract a small size of decode RAW data from 'filePath' picture file using 'DRawDecoderSettings' settings. This is a cancelable method which require a class instance to run because RAW pictures decoding can take a while.

This method return:

- A byte array container 'imageData' with picture data. Pixels order is RGB. Color depth can be 8 or 16. In 8 bits you can access to color component using (uchar\*), in 16 bits using (ushort\*).
- Size size of image in number of pixels ('width' and 'height').
- The max average of RGB components from decoded picture.
- 'false' is returned if decoding failed, else 'true'.

#### 6.499.2.4 decodeRAWImage()

```
bool Digikam::DRawDecoder::decodeRAWImage (
    const QString & filePath,
    const DRawDecoderSettings & DRawDecoderSettings,
    QByteArray & imageData,
    int & width,
    int & height,
    int & rgbmax )
```

Extract a full size of RAW data from 'filePath' picture file using 'DRawDecoderSettings' settings. This is a cancelable method which require a class instance to run because RAW pictures decoding can take a while.

This method return:

- A byte array container 'imageData' with picture data. Pixels order is RGB. Color depth can be 8 or 16. In 8 bits you can access to color component using (uchar\*), in 16 bits using (ushort\*).
- Size size of image in number of pixels ('width' and 'height').
- The max average of RGB components from decoded picture.
- 'false' is returned if decoding failed, else 'true'.

#### 6.499.2.5 extractRAWData()

```
bool Digikam::DRawDecoder::extractRAWData (
    const QString & filePath,
    QByteArray & rawData,
    DRawInfo & identify,
    unsigned int shotSelect = 0 )
```

Extract Raw image data undemosaiced and without post processing from 'filePath' picture file. This is a cancelable method which require a class instance to run because RAW pictures loading can take a while.

This method return:

- A byte array container 'rawData' with raw data.
- All info about Raw image into 'identify' container.
- 'false' is returned if loading failed, else 'true'.

#### 6.499.2.6 librawUseGomp()

```
int Digikam::DRawDecoder::librawUseGomp ( ) [static]
```

Return true or false if LibRaw use parallel demosaicing or not (libgomp support). Return -1 if undefined.

#### 6.499.2.7 librawVersion()

```
QString Digikam::DRawDecoder::librawVersion ( ) [static]
```

Return LibRaw version string.

**6.499.2.8 loadEmbeddedPreview()** [1/3]

```
bool Digikam::DRawDecoder::loadEmbeddedPreview (
    QByteArray & imgData,
    const QBuffer & inBuffer ) [static]
```

Get the embedded JPEG preview image from RAW image passed in QBuffer as a QByteArray which will include Exif Data. This is fast and non cancelable. This method does not require a class instance to run.

**6.499.2.9 loadEmbeddedPreview()** [2/3]

```
bool Digikam::DRawDecoder::loadEmbeddedPreview (
    QByteArray & imgData,
    const QString & path ) [static]
```

Get the embedded JPEG preview image from RAW picture as a QByteArray which will include Exif Data. This is fast and non cancelable. This method does not require a class instance to run.

**6.499.2.10 loadEmbeddedPreview()** [3/3]

```
bool Digikam::DRawDecoder::loadEmbeddedPreview (
    QImage & image,
    const QString & path ) [static]
```

Get the embedded JPEG preview image from RAW picture as a QImage. This is fast and non cancelable This method does not require a class instance to run.

**6.499.2.11 loadFullImage()**

```
bool Digikam::DRawDecoder::loadFullImage (
    QImage & image,
    const QString & path,
    const DRawDecoderSettings & settings = DRawDecoderSettings() ) [static]
```

Get the full decoded RAW picture. This is a more slower than [loadHalfPreview\(\)](#) method and non cancelable. This method does not require a class instance to run.

**6.499.2.12 loadHalfPreview()** [1/3]

```
bool Digikam::DRawDecoder::loadHalfPreview (
    QByteArray & imgData,
    const QBuffer & inBuffer ) [static]
```

Get the half decoded RAW picture passed in QBuffer as JPEG data in QByteArray. This is slower than [loadEmbeddedPreview\(\)](#) method and non cancelable. This method does not require a class instance to run.

**6.499.2.13 loadHalfPreview()** [2/3]

```
bool Digikam::DRawDecoder::loadHalfPreview (
    QByteArray & imgData,
    const QString & path ) [static]
```

Get the half decoded RAW picture as JPEG data in QByteArray. This is slower than [loadEmbeddedPreview\(\)](#) method and non cancelable. This method does not require a class instance to run.

**6.499.2.14 loadHalfPreview()** [3/3]

```
bool Digikam::DRawDecoder::loadHalfPreview (
    QImage & image,
    const QString & path,
    bool rotate = true ) [static]
```

Get the half decoded RAW picture. This is slower than [loadEmbeddedPreview\(\)](#) method and non cancelable. This method does not require a class instance to run.

**6.499.2.15 loadRawPreview()** [1/3]

```
static bool Digikam::DRawDecoder::loadRawPreview (
    QByteArray & imgData,
    const QBuffer & inBuffer ) [static]
```

Get the preview of RAW picture passed in QBuffer as a QByteArray holding JPEG data. It tries [loadEmbeddedPreview\(\)](#) first and if it fails, calls [loadHalfPreview\(\)](#).

**6.499.2.16 loadRawPreview()** [2/3]

```
static bool Digikam::DRawDecoder::loadRawPreview (
    QByteArray & imgData,
    const QString & path ) [static]
```

Get the preview of RAW picture as a QByteArray holding JPEG data. It tries [loadEmbeddedPreview\(\)](#) first and if it fails, calls [loadHalfPreview\(\)](#).

**6.499.2.17 loadRawPreview()** [3/3]

```
bool Digikam::DRawDecoder::loadRawPreview (
    QImage & image,
    const QString & path ) [static]
```

Get the preview of RAW picture as a QImage. It tries [loadEmbeddedPreview\(\)](#) first and if it fails, calls [loadHalfPreview\(\)](#).

#### 6.499.2.18 rawFileIdentify()

```
bool Digikam::DRawDecoder::rawFileIdentify (
    DRawInfo & identify,
    const QString & path ) [static]
```

Get the camera settings which have taken RAW file. Look into rawinfo.h for more details. This is a fast and non cancelable method which do not require a class instance to run.

#### 6.499.2.19 rawFiles()

```
QString Digikam::DRawDecoder::rawFiles ( ) [static]
```

Return the string of all RAW file type mime supported.

#### 6.499.2.20 rawFilesList()

```
QStringList Digikam::DRawDecoder::rawFilesList ( ) [static]
```

Return the list of all RAW file type mime supported, as a QStringList, without wildcard and suffix dot.

#### 6.499.2.21 rawFilesVersion()

```
int Digikam::DRawDecoder::rawFilesVersion ( ) [static]
```

Returns a version number for the list of supported RAW file types. This version is incremented if the list of supported formats has changed between library releases.

#### 6.499.2.22 setWaitingDataProgress()

```
void Digikam::DRawDecoder::setWaitingDataProgress (
    double value ) [protected], [virtual]
```

Re-implement this method to control the pseudo progress value during RAW decoding (when ddraw run with an internal loop without feedback) with your proper environment. By default, this method does nothing. Progress value average for this stage is 0%-n%, with 'n' == 40% max (see [setWaitingDataProgress\(\)](#) method).

#### 6.499.2.23 supportedCamera()

```
QStringList Digikam::DRawDecoder::supportedCamera ( ) [static]
```

Provide a list of supported RAW Camera name.

## 6.499.3 Member Data Documentation

### 6.499.3.1 m\_cancel

```
bool Digikam::DRawDecoder::m_cancel = false [protected]
```

Used internally to cancel RAW decoding operation. Normally, you don't need to use it directly, excepted if you derivated this class. Usual way is to use [cancel\(\)](#) method

### 6.499.3.2 m\_decoderSettings

```
DRawDecoderSettings Digikam::DRawDecoder::m_decoderSettings [protected]
```

The settings container used to perform RAW pictures decoding. See 'drawdecodingsetting.h' for details.

## 6.500 Digikam::DRawDecoder::Private Class Reference

### Public Member Functions

- void **exifParserCallback** (int tag, int type, int len, unsigned int ord, void \*ifp, INT64 base)
- bool **loadFromLibraw** (const QString &filePath, QByteArray &imageData, int &width, int &height, int &rgb-max)
- **Private** ([DRawDecoder](#) \*const p)
- int **progressCallback** (enum LibRaw\_progress p, int iteration, int expected)
- double **progressValue** () const
- void **setProgress** (double value)

### Static Public Member Functions

- static void **createPPMHeader** (QByteArray &imgData, libraw\_processed\_image\_t \*const img)
- static void **fillIdentifyInfo** (LibRaw \*const raw, [DRawInfo](#) &identify)
- static bool **loadEmbeddedPreview** (QByteArray &, LibRaw \*const raw)
- static bool **loadHalfPreview** (QImage &, LibRaw \*const raw, bool rotate=true)

### Friends

- class [DRawDecoder](#)

## 6.501 Digikam::DRawDecoderSettings Class Reference

### Public Types

- enum [DecodingQuality](#) {  
**BILINEAR** = 0, **VNG** = 1, **PPG** = 2, **AHD** = 3,  
**DCB** = 4, **DHT** = 11, **AAHD** = 12 }  
*RAW decoding Interpolation methods.*
- enum [InputColorSpace](#) { **NOINPUTCS** = 0, **EMBEDDED**, **CUSTOMINPUTCS** }
- enum [NoiseReduction](#) { **NONR** = 0, **WAVELETSNR**, **FBDDNR** }
- enum [OutputColorSpace](#) {  
**RAWCOLOR** = 0, **SRGB**, **ADOBERGB**, **WIDEGAMMUT**,  
**PROPHOTO**, **CUSTOMOUTPUTCS** }
- enum [WhiteBalance](#) {  
**NONE** = 0, **CAMERA** = 1, **AUTO** = 2, **CUSTOM** = 3,  
**AERA** = 4 }

## Public Member Functions

- [DRawDecoderSettings](#) ()=default
- [DRawDecoderSettings](#) (const [DRawDecoderSettings](#) &o)
- [DRawDecoderSettings](#) & **operator=** (const [DRawDecoderSettings](#) &o)
- bool **operator==** (const [DRawDecoderSettings](#) &o) const
- void [optimizeTimeLoading](#) ()
- [~DRawDecoderSettings](#) ()=default

## Public Attributes

- bool [autoBrightness](#) = true
- int [blackPoint](#) = 0
- double [brightness](#) = 1.0
- int [customWhiteBalance](#) = 6500
- double **customWhiteBalanceGreen** = 1.0
- bool [dcbEnhanceFI](#) = false
- int [dcbIterations](#) = -1
  - For DCB interpolation.*
- QString [deadPixelMap](#)
- bool [DontStretchPixels](#) = false
- bool [enableBlackPoint](#) = false
- bool [enableWhitePoint](#) = false
- bool [expoCorrection](#) = false
- double [expoCorrectionHighlight](#) = 0.0
- double [expoCorrectionShift](#) = 1.0
- bool [fixColorsHighlights](#) = false
- bool [halfSizeColorImage](#) = false
- [InputColorSpace](#) [inputColorSpace](#) = NOINPUTCS
- QString [inputProfile](#)
- int [medianFilterPasses](#) = 0
- int [NRThreshold](#) = 0
- [NoiseReduction](#) [NRType](#) = NONR
- [OutputColorSpace](#) [outputColorSpace](#) = SRGB
- QString [outputProfile](#)
- [DecodingQuality](#) [RAWQuality](#) = BILINEAR
- bool [RGBInterpolate4Colors](#) = false
- bool [sixteenBitsImage](#) = false
- int [unclipColors](#) = 0
- [WhiteBalance](#) [whiteBalance](#) = CAMERA
- QRect [whiteBalanceArea](#)
- int [whitePoint](#) = 0

## 6.501.1 Member Enumeration Documentation

### 6.501.1.1 DecodingQuality

enum [Digikam::DRawDecoderSettings::DecodingQuality](#)

**Note**

from original ddraw demosaic

Bilinear: use high-speed but low-quality bilinear interpolation (default - for slow computer). In this method, the red value of a non-red pixel is computed as the average of the adjacent red pixels, and similar for blue and green. VNG: use Variable Number of Gradients interpolation. This method computes gradients near the pixel of interest and uses the lower gradients (representing smoother and more similar parts of the image) to make an estimate. PPG↔: use Patterned Pixel Grouping interpolation. Pixel Grouping uses assumptions about natural scenery in making estimates. It has fewer color artifacts on natural images than the Variable Number of Gradients method. AHD: use Adaptive Homogeneity-Directed interpolation. This method selects the direction of interpolation so as to maximize a homogeneity metric, thus typically minimizing color artifacts. DCB: DCB interpolation (see [www.linuxphoto.org/html/dcb.html](http://www.linuxphoto.org/html/dcb.html) for details) DHT: DHT interpolation. AAHD: Enhanced Adaptive AHD interpolation.

**6.501.1.2 InputColorSpace**

enum `Digikam::DRawDecoderSettings::InputColorSpace`

Input color profile used to decoded image NOINPUTCS: No input color profile. EMBEDDED: Use the camera profile embedded in RAW file if exist. CUSTOMINPUTCS: Use a custom input color space profile.

**6.501.1.3 NoiseReduction**

enum `Digikam::DRawDecoderSettings::NoiseReduction`

Noise Reduction method to apply before demosaicing NONR: No noise reduction. WAVELETSNR: wavelets correction to erase noise while preserving real detail. It's applied after interpolation. FBDDNR: Fake Before Demosaicing Denoising noise reduction. It's applied before interpolation.

**6.501.1.4 OutputColorSpace**

enum `Digikam::DRawDecoderSettings::OutputColorSpace`

Output RGB color space used to decoded image RAWCOLOR: No output color profile (Linear RAW). SRGB: Use standard sRGB color space. ADOBERGB: Use standard Adobe RGB color space. WIDEGAMMUT: Use standard RGB Wide Gamut color space. PROPHOTO: Use standard RGB Pro Photo color space. CUSTOMOUTPUTCS: Use a custom workspace color profile.

**6.501.1.5 WhiteBalance**

enum `Digikam::DRawDecoderSettings::WhiteBalance`

White balances alternatives NONE: no white balance used : reverts to standard daylight D65 WB. CAMERA: Use the camera embedded WB if available. Reverts to NONE if not. AUTO: Averages an auto WB on the entire image. CUSTOM: Let use set it's own temperature and green factor (later converted to RGBG factors). AERA: Let use an area from image to average white balance (see `whiteBalanceArea` for details).



## 6.501.2 Constructor & Destructor Documentation

### 6.501.2.1 DRawDecoderSettings() [1/2]

```
Digikam::DRawDecoderSettings::DRawDecoderSettings ( ) [default]
```

Standard constructor with default settings

### 6.501.2.2 DRawDecoderSettings() [2/2]

```
Digikam::DRawDecoderSettings::DRawDecoderSettings (
    const DRawDecoderSettings & o )
```

Equivalent to the copy constructor

### 6.501.2.3 ~DRawDecoderSettings()

```
Digikam::DRawDecoderSettings::~~DRawDecoderSettings ( ) [default]
```

Standard destructor

## 6.501.3 Member Function Documentation

### 6.501.3.1 operator==( )

```
bool Digikam::DRawDecoderSettings::operator==(
    const DRawDecoderSettings & o ) const
```

Compare for equality

### 6.501.3.2 optimizeTimeLoading()

```
void Digikam::DRawDecoderSettings::optimizeTimeLoading ( )
```

Method to use a settings to optimize time loading, for example to compute image histogram

## 6.501.4 Member Data Documentation

### 6.501.4.1 autoBrightness

```
bool Digikam::DRawDecoderSettings::autoBrightness = true
```

If false, use a fixed white level, ignoring the image histogram.

#### 6.501.4.2 blackPoint

```
int Digikam::DRawDecoderSettings::blackPoint = 0
```

Black Point value of output image.

#### 6.501.4.3 brightness

```
double Digikam::DRawDecoderSettings::brightness = 1.0
```

Brightness of output image.

#### 6.501.4.4 customWhiteBalance

```
int Digikam::DRawDecoderSettings::customWhiteBalance = 6500
```

The temperature and the green multiplier of the custom white balance

#### 6.501.4.5 dcbEnhanceFl

```
bool Digikam::DRawDecoderSettings::dcbEnhanceFl = false
```

Turn on the DCB interpolation with enhance interpolated colors.

#### 6.501.4.6 dcbIterations

```
int Digikam::DRawDecoderSettings::dcbIterations = -1
```

Number of DCB median filtering correction passes. -1 : disable (default) 1-10 : DCB correction passes

#### 6.501.4.7 deadPixelMap

```
QString Digikam::DRawDecoderSettings::deadPixelMap
```

Path to text file including dead pixel list.

#### 6.501.4.8 DontStretchPixels

```
bool Digikam::DRawDecoderSettings::DontStretchPixels = false
```

For cameras with non-square pixels, do not stretch the image to its correct aspect ratio. In any case, this option guarantees that each output pixel corresponds to one RAW pixel.

#### 6.501.4.9 enableBlackPoint

```
bool Digikam::DRawDecoderSettings::enableBlackPoint = false
```

Turn on the black point setting to decode RAW image.

#### 6.501.4.10 enableWhitePoint

```
bool Digikam::DRawDecoderSettings::enableWhitePoint = false
```

Turn on the white point setting to decode RAW image.

#### 6.501.4.11 expoCorrection

```
bool Digikam::DRawDecoderSettings::expoCorrection = false
```

Turn on the Exposure Correction before interpolation.

#### 6.501.4.12 expoCorrectionHighlight

```
double Digikam::DRawDecoderSettings::expoCorrectionHighlight = 0.0
```

Amount of highlight preservation for exposure correction before interpolation in E.V. Usable range is from 0.0 (linear exposure shift, highlights may blow) to 1.0 (maximum highlights preservation) This settings can only take effect if `expoCorrectionShift > 1.0`.

#### 6.501.4.13 expoCorrectionShift

```
double Digikam::DRawDecoderSettings::expoCorrectionShift = 1.0
```

Shift of Exposure Correction before interpolation in linear scale. Usable range is from 0.25 (darken image 1 stop : -2EV) to 8.0 (lighten ~1.5 photographic stops : +3EV).

#### 6.501.4.14 fixColorsHighlights

```
bool Digikam::DRawDecoderSettings::fixColorsHighlights = false
```

If true, images with overblown channels are processed much more accurate, without 'pink clouds' (and blue highlights under tungsten lamps).

#### 6.501.4.15 halfSizeColorImage

```
bool Digikam::DRawDecoderSettings::halfSizeColorImage = false
```

Half-size color image decoding (twice as fast as "enableRAWQuality"). Turn on this option to reduce time loading to render histogram for example, no to render an image to screen.

#### 6.501.4.16 inputColorSpace

```
InputColorSpace Digikam::DRawDecoderSettings::inputColorSpace = NOINPUTCS
```

The input color profile used to decoded RAW data. See OutputColorProfile values for details.

#### 6.501.4.17 inputProfile

```
QString Digikam::DRawDecoderSettings::inputProfile
```

Path to custom input ICC profile to define the camera's raw colorspace.

#### 6.501.4.18 medianFilterPasses

```
int Digikam::DRawDecoderSettings::medianFilterPasses = 0
```

After interpolation, clean up color artifacts by repeatedly applying a 3x3 median filter to the R-G and B-G channels.

#### 6.501.4.19 NRThreshold

```
int Digikam::DRawDecoderSettings::NRThreshold = 0
```

Noise reduction threshold value. Null value disable NR. Range is between 100 and 1000. For IMPULSENK : set the amount of Luminance impulse denoise.

#### 6.501.4.20 NRType

```
NoiseReduction Digikam::DRawDecoderSettings::NRType = NONR
```

Noise reduction method to apply before demosaicing.

#### 6.501.4.21 outputColorSpace

```
OutputColorSpace Digikam::DRawDecoderSettings::outputColorSpace = SRGB
```

The output color profile used to decoded RAW data. See OutputColorProfile values for details.

#### 6.501.4.22 outputProfile

```
QString Digikam::DRawDecoderSettings::outputProfile
```

Path to custom output ICC profile to define the color workspace.

#### 6.501.4.23 RAWQuality

`DecodingQuality` Digikam::DRawDecoderSettings::RAWQuality = BILINEAR

RAW quality decoding factor value. See DecodingQuality values for details.

#### 6.501.4.24 RGBInterpolate4Colors

`bool` Digikam::DRawDecoderSettings::RGBInterpolate4Colors = false

Turn on RAW file decoding using RGB interpolation as four colors.

#### 6.501.4.25 sixteenBitsImage

`bool` Digikam::DRawDecoderSettings::sixteenBitsImage = false

Turn on RAW file decoding in 16 bits per color per pixel instead 8 bits.

#### 6.501.4.26 unclipColors

`int` Digikam::DRawDecoderSettings::unclipColors = 0

Unclip Highlight color level: 0 = Clip all highlights to solid white. 1 = Leave highlights unclipped in various shades of pink. 2 = Blend clipped and unclipped values together for a gradual fade to white. 3-9 = Reconstruct highlights. Low numbers favor whites; high numbers favor colors.

#### 6.501.4.27 whiteBalance

`WhiteBalance` Digikam::DRawDecoderSettings::whiteBalance = CAMERA

White balance type to use. See WhiteBalance values for detail

#### 6.501.4.28 whiteBalanceArea

`QRect` Digikam::DRawDecoderSettings::whiteBalanceArea

Rectangle used to calculate the white balance by averaging the region of image.

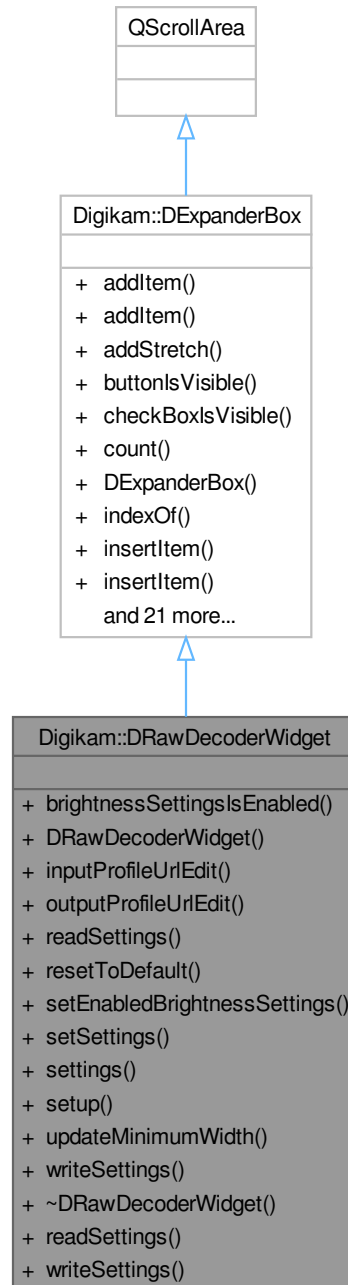
#### 6.501.4.29 whitePoint

`int` Digikam::DRawDecoderSettings::whitePoint = 0

White Point value of output image.

## 6.502 Digikam::DRawDecoderWidget Class Reference

Inheritance diagram for Digikam::DRawDecoderWidget:



### Public Types

- enum **AdvancedSettingsOptions** { **SIXTEENBITS** = 0x00000001 , **COLORSPACE** = 0x00000002 , **POST-PROCESSING** = 0x00000004 , **BLACKWHITEPOINTS** = 0x00000008 }
- enum **SettingsTabs** { **DEMOSAICING** = 0 , **WHITEBALANCE** , **CORRECTIONS** , **COLORMANAGEMENT** }

## Signals

- void **signalSettingsChanged** ()
- void **signalSixteenBitsImageToggled** (bool)

## Signals inherited from [Digikam::DExpanderBox](#)

- void **signalItemButtonPressed** (int index)
- void **signalItemExpanded** (int index, bool b)
- void **signalItemToggled** (int index, bool b)

## Public Member Functions

- bool **brightnessSettingsIsEnabled** () const
- [DRawDecoderWidget](#) (QWidget \*const parent, int advSettings=COLORSPACE)  
*The widget to host the Raw Decoder settings.*
- [DFileSelector](#) \* **inputProfileUriEdit** () const
- [DFileSelector](#) \* **outputProfileUriEdit** () const
- void **readSettings** (KConfigGroup &group) override
- void **resetToDefault** ()
- void **setEnabledBrightnessSettings** (bool b)
- void **setSettings** (const [DRawDecoderSettings](#) &settings)
- [DRawDecoderSettings](#) **settings** () const
- void **setup** (int advSettings)
- void **updateMinimumWidth** ()
- void **writeSettings** (KConfigGroup &group) override

## Public Member Functions inherited from [Digikam::DExpanderBox](#)

- void **addItem** (QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **addItem** (QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **addStretch** ()
- bool **buttonIsVisible** (int index) const
- bool **checkboxIsVisible** (int index) const
- int **count** () const
- [DExpanderBox](#) (QWidget \*const parent=nullptr)
- int **indexOf** ([DLabelExpander](#) \*const widget) const
- void **insertItem** (int index, QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertItem** (int index, QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertStretch** (int index)
- bool **isChecked** (int index) const
- bool **isItemEnabled** (int index) const
- bool **isItemExpanded** (int index) const
- QIcon **itemIcon** (int index) const
- QString **itemText** (int index) const
- QString **itemToolTip** (int index) const
- void **removeItem** (int index)
- void **setButtonIcon** (int index, const QIcon &icon)
- void **setButtonVisible** (int index, bool b)

- void **setCheckBoxVisible** (int index, bool b)
- void **setChecked** (int index, bool b)
- void **setItemEnabled** (int index, bool enabled)
- void **setItemExpanded** (int index, bool b)
- void **setItemIcon** (int index, const QIcon &icon)
- void **setItemText** (int index, const QString &txt)
- void **setItemToolTip** (int index, const QString &tip)
- [DLabelExpander](#) \* **widget** (int index) const

### Static Public Member Functions

- static void **readSettings** ([DRawDecoderSettings](#) &setting, const KConfigGroup &group)
- static void **writeSettings** (const [DRawDecoderSettings](#) &setting, KConfigGroup &group)

## 6.502.1 Constructor & Destructor Documentation

### 6.502.1.1 DRawDecoderWidget()

```
Digikam::DRawDecoderWidget::DRawDecoderWidget (
    QWidget *const parent,
    int advSettings = COLORSPACE ) [explicit]
```

#### Parameters

<i>parent</i>	the parent widget instance
<i>advSettings</i>	the default value is COLORSPACE

## 6.502.2 Member Function Documentation

### 6.502.2.1 readSettings()

```
void Digikam::DRawDecoderWidget::readSettings (
    KConfigGroup & group ) [override], [virtual]
```

Reimplemented from [Digikam::DExpanderBox](#).

### 6.502.2.2 writeSettings()

```
void Digikam::DRawDecoderWidget::writeSettings (
    KConfigGroup & group ) [override], [virtual]
```

Reimplemented from [Digikam::DExpanderBox](#).



## 6.503 Digikam::DRawDecoding Class Reference

### Public Member Functions

- [DRawDecoding](#) ()
- [DRawDecoding](#) (const [DRawDecoderSettings](#) &prm)
- bool [operator==](#) (const [DRawDecoding](#) &other) const
- void [optimizeTimeLoading](#) ()
- bool [postProcessingSettingsIsDirty](#) () const
- void [resetPostProcessingSettings](#) ()
- void [writeToFilterAction](#) ([FilterAction](#) &action, const QString &prefix=QString()) const
- [~DRawDecoding](#) ()=default

### Static Public Member Functions

- static void [decodingSettingsFromXml](#) (const QDomElement &elm, [DRawDecoderSettings](#) &prm)
- static void [decodingSettingsToXml](#) (const [DRawDecoderSettings](#) &prm, QDomElement &elm)
- static [DRawDecoding fromFilterAction](#) (const [FilterAction](#) &action, const QString &prefix=QString())

### Public Attributes

- [BCGContainer](#) *bcg*  
*Post Processing settings -----.*
- [CurvesContainer](#) *curvesAdjust*
- [DRawDecoderSettings](#) *rawPrm*
- [WBContainer](#) *wb*

## 6.503.1 Constructor & Destructor Documentation

### 6.503.1.1 DRawDecoding() [1/2]

```
Digikam::DRawDecoding::DRawDecoding ( )
```

Standard constructor with default settings

### 6.503.1.2 DRawDecoding() [2/2]

```
Digikam::DRawDecoding::DRawDecoding (
    const DRawDecoderSettings & prm ) [explicit]
```

Copy constructor. Creates a copy of a [DRawDecoderSettings](#) object.

### 6.503.1.3 ~DRawDecoding()

```
Digikam::DRawDecoding::~~DRawDecoding ( ) [default]
```

Standard destructor

## 6.503.2 Member Function Documentation

### 6.503.2.1 decodingSettingsToXml()

```
void Digikam::DRawDecoding::decodingSettingsToXml (
    const DRawDecoderSettings & prm,
    QDomElement & elm ) [static]
```

Used by BQM to read/store Queue Raw decoding settings from/to configuration file

### 6.503.2.2 operator==( )

```
bool Digikam::DRawDecoding::operator==(
    const DRawDecoding & other ) const
```

Equality operator.

### 6.503.2.3 optimizeTimeLoading()

```
void Digikam::DRawDecoding::optimizeTimeLoading ( )
```

Method to use a settings to optimize time loading, for example to compute image histogram

### 6.503.2.4 postProcessingSettingsIsDirty()

```
bool Digikam::DRawDecoding::postProcessingSettingsIsDirty ( ) const
```

Method to check is a post-processing setting have been changed

### 6.503.2.5 resetPostProcessingSettings()

```
void Digikam::DRawDecoding::resetPostProcessingSettings ( )
```

Method to reset to default values all Raw processing settings.

## 6.503.3 Member Data Documentation

### 6.503.3.1 bcg

```
BCGContainer Digikam::DRawDecoding::bcg
```

BCG correction values.

### 6.503.3.2 curvesAdjust

[CurvesContainer](#) Digikam::DRawDecoding::curvesAdjust

Curve adjustments.

### 6.503.3.3 rawPrm

[DRawDecoderSettings](#) Digikam::DRawDecoding::rawPrm

All Raw decoding settings provided by RawEngine.

### 6.503.3.4 wb

[WBContainer](#) Digikam::DRawDecoding::wb

White Balance correction values.

## 6.504 Digikam::DRawInfo Class Reference

### Public Types

- enum [ImageOrientation](#) {  
**ORIENTATION\_NONE** = 0 , **ORIENTATION\_180** = 3 , **ORIENTATION\_Mirror90CCW** = 4 , **ORIENTATION\_90CCW** = 5 ,  
**ORIENTATION\_90CW** = 6 }

### Public Member Functions

- [DRawInfo](#) ()
- [~DRawInfo](#) ()=default

### Public Attributes

- double **altitude** = 0.0F
- float [ambientAcceleration](#) = -1000.0F
- float [ambientElevationAngle](#) = -1000.0F
- float [ambientHumidity](#) = -1000.0F
- float [ambientPressure](#) = -1000.0F
- float [ambientTemperature](#) = -1000.0F
- float [ambientWaterDepth](#) = 1000.0F
- float [aperture](#) = -1.0F
- float [baselineExposure](#) = -999.0F
- unsigned int [blackPoint](#) = 0
- unsigned int [blackPointCh](#) [4] = { 0 }
- float [cameraColorMatrix1](#) [3][4]
- float **cameraColorMatrix2** [3][4]
- double [cameraMult](#) [4] = { 0.0 }

- float **cameraXYZMatrix** [4][3]
- QString **colorKeys**
- QDateTime **dateTime**
- double **daylightMult** [3] = { 0.0 }
- QString **description**
- QString **DNGVersion**
- float **exposureIndex** = -1.0F
- int **exposureProgram** = -1
- float **exposureTime** = -1.0F
- QString **filterPattern**
- QString **firmware**
- int **flashUsed** = -1
- float **focalLength** = -1.0F
- int **focalLengthIn35mmFilm** = -1  
*Valid value is unsigned.*
- QSize **fullSize**
- bool **hasGpsInfo** = false  
*true if GPS info are parsed from RAW file.*
- bool **hasIccProfile** = false
- QByteArray **iccData**
- QString **imageID**
- QSize **imageSize**
- bool **isDecodable** = false
- double **latitude** = 0.0F
- unsigned int **leftMargin** = 0
- QString **lensMake**
- QString **lensModel**
- QString **lensSerial**
- QString **localizedCameraModel**
- double **longitude** = 0.0F
- QString **make**
- float **maxAperture** = -1.0F  
*Valid value is unsigned.*
- int **meteringMode** = -1
- QString **model**
- **ImageOrientation** **orientation** = ORIENTATION\_NONE
- QString **originalRawFileName**
- QSize **outputSize**
- QString **owner**
- float **pixelAspectRatio** = 1.0F
- int **rawColors** = -1
- QString **rawDataUniqueID**
- int **rawImages** = -1
- float **sensitivity** = -1.0F
- unsigned int **serialNumber** = 0
- QString **software**
- QByteArray **thumbnail**
- QSize **thumbSize**
- unsigned int **topMargin** = 0
- QString **uniqueCameraModel**
- unsigned int **whitePoint** = 0
- QByteArray **xmpData**

## 6.504.1 Member Enumeration Documentation

### 6.504.1.1 ImageOrientation

enum `Digikam::DRawInfo::ImageOrientation`

The RAW image orientation values

## 6.504.2 Constructor & Destructor Documentation

### 6.504.2.1 DRawInfo()

```
Digikam::DRawInfo::DRawInfo ( ) [explicit]
```

Standard constructor < NOTE: see bug #253911 : [y][x] not [x][y]

### 6.504.2.2 ~DRawInfo()

```
Digikam::DRawInfo::~~DRawInfo ( ) [default]
```

Standard destructor

## 6.504.3 Member Data Documentation

### 6.504.3.1 ambientAcceleration

```
float Digikam::DRawInfo::ambientAcceleration = -1000.0F
```

Directionless camera acceleration in units of mGal, or 10<sup>-5</sup> m/s<sup>2</sup>. -1000 is an invalid acceleration.

### 6.504.3.2 ambientElevationAngle

```
float Digikam::DRawInfo::ambientElevationAngle = -1000.0F
```

Camera elevation angle in degrees. -1000 is an invalid angle.

### 6.504.3.3 ambientHumidity

```
float Digikam::DRawInfo::ambientHumidity = -1000.0F
```

Ambient relative humidity in percent. -1000 is an invalid humidity.

### 6.504.3.4 ambientPressure

```
float Digikam::DRawInfo::ambientPressure = -1000.0F
```

Ambient air pressure in hPa or mbar. -1000 is an invalid pressure.

### 6.504.3.5 ambientTemperature

```
float Digikam::DRawInfo::ambientTemperature = -1000.0F
```

Ambient temperature in Celsius degrees. -1000 is an invalid temperature.

### 6.504.3.6 ambientWaterDepth

```
float Digikam::DRawInfo::ambientWaterDepth = 1000.0F
```

Depth under water in metres, negative for above water. 1000 is an invalid water depth.

### 6.504.3.7 aperture

```
float Digikam::DRawInfo::aperture = -1.0F
```

Aperture value in APEX.

### 6.504.3.8 baselineExposure

```
float Digikam::DRawInfo::baselineExposure = -999.0F
```

Exposure compensation to be applied during raw conversion. -999 is an invalid exposure.

### 6.504.3.9 blackPoint

```
unsigned int Digikam::DRawInfo::blackPoint = 0
```

Black level from Raw histogram.

### 6.504.3.10 blackPointCh

```
unsigned int Digikam::DRawInfo::blackPointCh[4] = { 0 }
```

Channel black levels from Raw histogram.

### 6.504.3.11 cameraColorMatrix1

```
float Digikam::DRawInfo::cameraColorMatrix1[3][4]
```

Camera Color [Matrix](#)

### 6.504.3.12 cameraMult

```
double Digikam::DRawInfo::cameraMult[4] = { 0.0 }
```

Camera multipliers used for White Balance adjustments

### 6.504.3.13 colorKeys

```
QString Digikam::DRawInfo::colorKeys
```

The used Color Keys

### 6.504.3.14 dateTime

```
QDateTime Digikam::DRawInfo::dateTime
```

Date & time when the picture has been taken.

### 6.504.3.15 daylightMult

```
double Digikam::DRawInfo::daylightMult[3] = { 0.0 }
```

White color balance settings.

### 6.504.3.16 description

```
QString Digikam::DRawInfo::description
```

The image description of raw image.

### 6.504.3.17 DNGVersion

```
QString Digikam::DRawInfo::DNGVersion
```

The DNG version. NOTE: it is only shown with DNG RAW files.

### 6.504.3.18 exposureIndex

```
float Digikam::DRawInfo::exposureIndex = -1.0F
```

Exposure Index from the camera. Valid value is unsigned.

### 6.504.3.19 exposureProgram

```
int Digikam::DRawInfo::exposureProgram = -1
```

The exposure program used by camera. Valid value is unsigned.

### 6.504.3.20 exposureTime

```
float Digikam::DRawInfo::exposureTime = -1.0F
```

1/exposureTime = exposure time in seconds.

#### 6.504.3.21 filterPattern

```
QString Digikam::DRawInfo::filterPattern
```

The demosaicing filter pattern.

#### 6.504.3.22 firmware

```
QString Digikam::DRawInfo::firmware
```

The Firmware name or version which create raw image.

#### 6.504.3.23 flashUsed

```
int Digikam::DRawInfo::flashUsed = -1
```

Describe how flash has been used by camera. Valid value is unsigned.

#### 6.504.3.24 focalLength

```
float Digikam::DRawInfo::focalLength = -1.0F
```

Focal Length value in mm.

#### 6.504.3.25 fullSize

```
QSize Digikam::DRawInfo::fullSize
```

The full RAW image dimensions in pixels.

#### 6.504.3.26 hasIccProfile

```
bool Digikam::DRawInfo::hasIccProfile = false
```

True if RAW file include an ICC color profile.

#### 6.504.3.27 iccData

```
QByteArray Digikam::DRawInfo::iccData
```

ICC color profilr container extracted from RAW file, if present.

#### 6.504.3.28 imageID

```
QString Digikam::DRawInfo::imageID
```

An unique image ID generated by camera.



### 6.504.3.29 imageSize

```
QSize Digikam::DRawInfo::imageSize
```

The image dimensions in pixels.

### 6.504.3.30 isDecodable

```
bool Digikam::DRawInfo::isDecodable = false
```

True is RAW file is decodable by dcrw.

### 6.504.3.31 latitude

```
double Digikam::DRawInfo::latitude = 0.0F
```

GPS information

### 6.504.3.32 leftMargin

```
unsigned int Digikam::DRawInfo::leftMargin = 0
```

Left margin of raw image.

### 6.504.3.33 lensModel

```
QString Digikam::DRawInfo::lensModel
```

Description of lens properties.

### 6.504.3.34 localizedCameraModel

```
QString Digikam::DRawInfo::localizedCameraModel
```

Localized name for the camera model that created the raw file

### 6.504.3.35 make

```
QString Digikam::DRawInfo::make
```

The camera maker.

### 6.504.3.36 meteringMode

```
int Digikam::DRawInfo::meteringMode = -1
```

The metering mode used by camera. Valid value is unsigned.

### 6.504.3.37 model

```
QString Digikam::DRawInfo::model
```

The camera model.

### 6.504.3.38 orientation

```
ImageOrientation Digikam::DRawInfo::orientation = ORIENTATION_NONE
```

The raw image orientation

### 6.504.3.39 originalRawFileName

```
QString Digikam::DRawInfo::originalRawFileName
```

The original RAW file name.

### 6.504.3.40 outputSize

```
QSize Digikam::DRawInfo::outputSize
```

The output dimensions in pixels.

### 6.504.3.41 owner

```
QString Digikam::DRawInfo::owner
```

The artist name who have picture owner.

### 6.504.3.42 pixelAspectRatio

```
float Digikam::DRawInfo::pixelAspectRatio = 1.0F
```

The pixel Aspect Ratio if != 1.0. NOTE: if == 1.0, libraw CLI tool do not show this value. Default value = 1.0. This can be unavailable (depending of camera model).

### 6.504.3.43 rawColors

```
int Digikam::DRawInfo::rawColors = -1
```

The number of RAW colors.

#### 6.504.3.44 rawDataUniqueID

```
QString Digikam::DRawInfo::rawDataUniqueID
```

An unique RAW data ID.

#### 6.504.3.45 rawImages

```
int Digikam::DRawInfo::rawImages = -1
```

The number of RAW images.

#### 6.504.3.46 sensitivity

```
float Digikam::DRawInfo::sensitivity = -1.0F
```

The sensitivity in ISO used by camera to take the picture.

#### 6.504.3.47 serialNumber

```
unsigned int Digikam::DRawInfo::serialNumber = 0
```

Serial number of raw image.

#### 6.504.3.48 software

```
QString Digikam::DRawInfo::software
```

The software name which process raw image.

#### 6.504.3.49 thumbnail

```
QByteArray Digikam::DRawInfo::thumbnail
```

Thumbnail image data extracted from raw file.

#### 6.504.3.50 thumbSize

```
QSize Digikam::DRawInfo::thumbSize
```

The thumb dimensions in pixels.

#### 6.504.3.51 topMargin

```
unsigned int Digikam::DRawInfo::topMargin = 0
```

Top margin of raw image.

**6.504.3.52 uniqueCameraModel**

```
QString Digikam::DRawInfo::uniqueCameraModel
```

Non-localized name for the camera model that created the raw file

**6.504.3.53 whitePoint**

```
unsigned int Digikam::DRawInfo::whitePoint = 0
```

White level from Raw histogram.

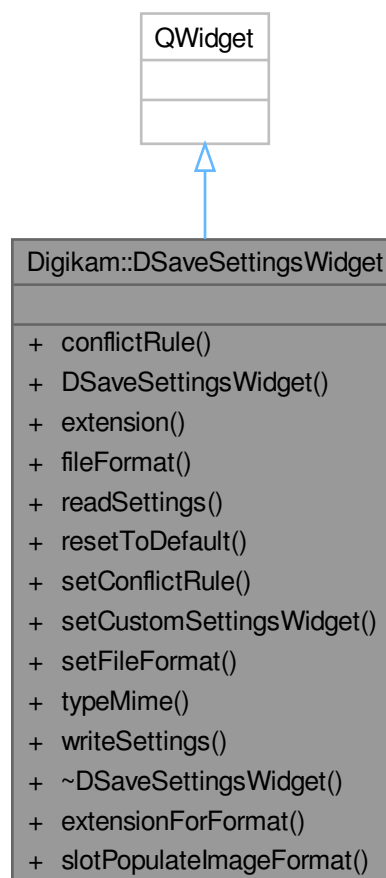
**6.504.3.54 xmpData**

```
QByteArray Digikam::DRawInfo::xmpData
```

Xmp metadata container extracted from RAW file, if present.

**6.505 Digikam::DSaveSettingsWidget Class Reference**

Inheritance diagram for Digikam::DSaveSettingsWidget:



## Public Types

- enum **OutputFormat** { **OUTPUT\_PNG** = 0 , **OUTPUT\_TIFF** , **OUTPUT\_JPEG** , **OUTPUT\_PPM** }

## Public Slots

- void **slotPopulateImageFormat** (bool sixteenBits)

## Signals

- void **signalConflictButtonChanged** (int)
- void **signalSaveFormatChanged** ()

## Public Member Functions

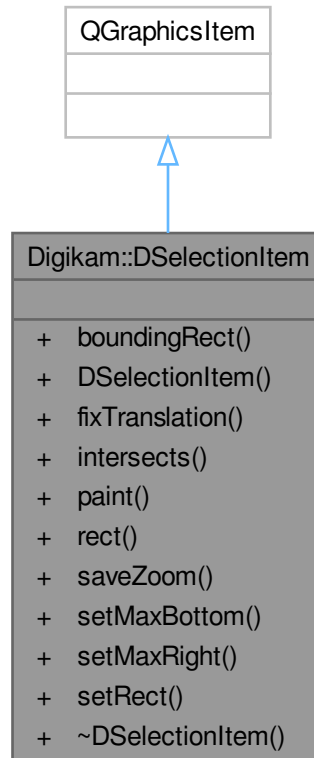
- FileSaveConflictBox::ConflictRule **conflictRule** () const
- **DSaveSettingsWidget** (QWidget \*const parent)
- QString **extension** () const
- OutputFormat **fileFormat** () const
- void **readSettings** (KConfigGroup &group)
- void **resetToDefault** ()
- void **setConflictRule** (FileSaveConflictBox::ConflictRule r)
- void **setCustomSettingsWidget** (QWidget \*const custom)
- void **setFileFormat** (OutputFormat f)
- QString **typeMime** () const
- void **writeSettings** (KConfigGroup &group)

## Static Public Member Functions

- static QString **extensionForFormat** (OutputFormat format)

## 6.506 Digikam::DSelectionItem Class Reference

Inheritance diagram for Digikam::DSelectionItem:



### Public Types

- enum **Intersects** {  
**None** , **Top** , **TopRight** , **Right** ,  
**BottomRight** , **Bottom** , **BottomLeft** , **Left** ,  
**TopLeft** , **Move** }

### Public Member Functions

- `QRectF` **boundingRect** () const override
- **DSelectionItem** (const `QRectF` &rect)
- `QPointF` **fixTranslation** (`QPointF` dp) const
- `Intersects` **intersects** (`QPointF` &point)
- void **paint** (`QPainter` \*painter, const `QStyleOptionGraphicsItem` \*option, `QWidget` \*widget) override
- `QRectF` **rect** () const
- void **saveZoom** (qreal zoom)
- void **setMaxBottom** (qreal maxBottom)
- void **setMaxRight** (qreal maxRight)
- void **setRect** (const `QRectF` &rect)

## 6.507 Digikam::DSelector Class Reference

Inheritance diagram for Digikam::DSelector:



### Public Member Functions

- Qt::ArrowType [arrowDirection](#) () const
- QRect [contentsRect](#) () const

- **DSelector** (Qt::Orientation o, QWidget \*const parent=nullptr)
- **DSelector** (QWidget \*const parent=nullptr)
- bool **indent** () const
- void **setArrowDirection** (Qt::ArrowType direction)
- void **setIndent** (bool i)

### Protected Member Functions

- virtual void **drawArrow** (QPainter \*painter, const QPoint &pos)
- virtual void **drawContents** (QPainter \*)
- void **mouseMoveEvent** (QMouseEvent \*e) override
- void **mousePressEvent** (QMouseEvent \*e) override
- void **mouseReleaseEvent** (QMouseEvent \*e) override
- void **paintEvent** (QPaintEvent \*) override
- void **wheelEvent** (QWheelEvent \*) override

### Properties

- Qt::ArrowType **arrowDirection**
- bool **indent**
- int **maxValue**
- int **minValue**
- int **value**

### Friends

- class **Private**

## 6.507.1 Detailed Description

**DSelector** is the base class for other widgets which provides the ability to choose from a one-dimensional range of values. An example is the **KGradientSelector** which allows to choose from a range of colors.

A custom drawing routine for the widget surface has to be provided by the subclass.

## 6.507.2 Member Function Documentation

### 6.507.2.1 arrowDirection()

```
Qt::ArrowType Digikam::DSelector::arrowDirection ( ) const
```

#### Returns

the current arrow direction



### 6.507.2.2 contentsRect()

```
QRect Digikam::DSelector::contentsRect ( ) const
```

#### Returns

the rectangle on which subclasses should draw.

### 6.507.2.3 drawArrow()

```
void Digikam::DSelector::drawArrow (
    QPainter * painter,
    const QPoint & pos ) [protected], [virtual]
```

Override this function to draw the cursor which indicates the current value.

### 6.507.2.4 drawContents()

```
virtual void Digikam::DSelector::drawContents (
    QPainter * ) [inline], [protected], [virtual]
```

Override this function to draw the contents of the control. The default implementation does nothing.

Draw only within [contentsRect\(\)](#).

Reimplemented in [Digikam::DColorValueSelector](#).

### 6.507.2.5 indent()

```
bool Digikam::DSelector::indent ( ) const
```

#### Returns

whether the indent option is set.

### 6.507.2.6 setArrowDirection()

```
void Digikam::DSelector::setArrowDirection (
    Qt::ArrowType direction )
```

Sets the arrow direction.

### 6.507.2.7 setIndent()

```
void Digikam::DSelector::setIndent (
    bool i )
```

Sets the indent option of the widget to i. This determines whether a shaded frame is drawn.

## 6.508 Digikam::DServiceInfo Class Reference

### Public Member Functions

- **DServiceInfo** (const [DServiceInfo](#) &other)
- **DServiceInfo** (const QString &\_name, const QString &\_exec, const QString &\_icon, const QString &\_topt, bool \_term)
- bool **isEmpty** () const
- [DServiceInfo](#) & **operator=** (const [DServiceInfo](#) &other)

### Public Attributes

- QString **exec**
- QString **icon**
- QString **name**
- bool **term** = false
- QString **topt**

## 6.509 Digikam::DServiceMenu Class Reference

### Static Public Member Functions

- static QIcon [getIconFromService](#) (const [DServiceInfo](#) &sinfo)
- static bool **runFiles** (const [DServiceInfo](#) &serviceInfo, const QList< QUrl > &urls)
- static bool [runFiles](#) (const KService::Ptr &service, const QList< QUrl > &urls)
- static bool [runFiles](#) (const QString &appCmd, const QList< QUrl > &urls, const KService::Ptr &service=KService::Ptr(), const [DServiceInfo](#) &serviceInfo=[DServiceInfo](#)())
- static QList< [DServiceInfo](#) > **servicesForOpen** (const QList< QUrl > &urls)
- static KService::List [servicesForOpenWith](#) (const QList< QUrl > &urls)

### 6.509.1 Member Function Documentation

#### 6.509.1.1 [getIconFromService\(\)](#)

```
QIcon Digikam::DServiceMenu::getIconFromService (
    const DServiceInfo & sinfo ) [static]
```

Return the QIcon depending on the operating system.

#### 6.509.1.2 [runFiles\(\)](#) [1/2]

```
bool Digikam::DServiceMenu::runFiles (
    const KService::Ptr & service,
    const QList< QUrl > & urls ) [static]
```

Linux only: open file urls with the service.

### 6.509.1.3 runFiles() [2/2]

```
bool Digikam::DServiceMenu::runFiles (
    const QString & appCmd,
    const QList< QUrl > & urls,
    const KService::Ptr & service = KService::Ptr(),
    const DServiceInfo & serviceInfo = DServiceInfo() ) [static]
```

Linux only: open file urls with the application command.

### 6.509.1.4 servicesForOpenWith()

```
KService::List Digikam::DServiceMenu::servicesForOpenWith (
    const QList< QUrl > & urls ) [static]
```

Linux only: return list of service available on desktop to open files.

## 6.510 Digikam::DSliderSpinBox Class Reference

Inheritance diagram for Digikam::DSliderSpinBox:



### Public Slots

- void **setValue** (int [value](#))  
Set the value, don't use [setValue\(\)](#)

## Signals

- void **valueChanged** (int [value](#))

## Public Member Functions

- **DSliderSpinBox** (QWidget \*const parent=nullptr)
- int **fastSliderStep** () const
- int **maximum** () const
- int **minimum** () const
- void **setFastSliderStep** (int step)
- void **setMaximum** (int maximum)
- void **setMinimum** (int minimum)
- void **setPageStep** (int [value](#))
- void **setRange** (int minimum, int maximum)
- void **setSingleStep** (int [value](#))
- int **value** ()

*Get the value, don't use [value\(\)](#)*

## Public Member Functions inherited from [Digikam::DAbstractSliderSpinBox](#)

- void **hideEdit** ()
- bool **isDragging** () const
- virtual QSize **minimumSize** () const
- QSize **minimumSizeHint** () const override
- void **setBlockUpdateSignalOnDrag** (bool block)
- void **setExponentRatio** (double dbl)
- void **setPrefix** (const QString &prefix)
- void **setSuffix** (const QString &suffix)
- void **showEdit** ()
- QSize **sizeHint** () const override

## Protected Member Functions

- void **setInternalValue** (int [value](#), bool blockUpdateSignal) override
- QString **valueString** () const override

## Protected Member Functions inherited from [Digikam::DAbstractSliderSpinBox](#)

- void **changeEvent** (QEvent \*e) override
- **DAbstractSliderSpinBox** (QWidget \*const parent, DAbstractSliderSpinBoxPrivate \*const q)
- QRect **downButtonRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- bool **eventFilter** (QObject \*recv, QEvent \*e) override
- void **focusInEvent** (QFocusEvent \*e) override
- void **keyPressEvent** (QKeyEvent \*e) override
- void **mouseMoveEvent** (QMouseEvent \*e) override
- void **mousePressEvent** (QMouseEvent \*e) override
- void **mouseReleaseEvent** (QMouseEvent \*e) override
- void **paint** (QPainter &painter)
- void **paintBreeze** (QPainter &painter)
- void **paintEvent** (QPaintEvent \*e) override
- void **paintFusion** (QPainter &painter)
- void **paintPlastique** (QPainter &painter)
- QStyleOptionProgressBar **progressBarOptions** () const
- QRect **progressRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- QStyleOptionSpinBox **spinBoxOptions** () const
- QRect **upButtonRect** (const QStyleOptionSpinBox &spinBoxOptions) const
- int **valueForX** (int x, Qt::KeyboardModifiers modifiers=Qt::NoModifier) const
- void **wheelEvent** (QWheelEvent \*e) override

## Properties

- int **maximum**
- int **minimum**

## Additional Inherited Members

### Protected Slots inherited from [Digikam::DAbstractSliderSpinBox](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override
- void **editLostFocus** ()

### Protected Attributes inherited from [Digikam::DAbstractSliderSpinBox](#)

- DAbstractSliderSpinBoxPrivate \*const **d\_ptr**

## 6.510.1 Member Function Documentation

### 6.510.1.1 setInternalValue()

```
void Digikam::DSliderSpinBox::setInternalValue (
    int value,
    bool blockUpdateSignal ) [override], [protected], [virtual]
```

Sets the slider internal value. Inheriting classes should respect blockUpdateSignal so that, in specific cases, we have a performance improvement. See setIgnoreMouseMoveEvents.

Implements [Digikam::DAbstractSliderSpinBox](#).

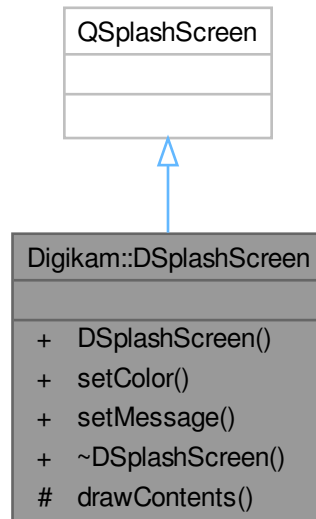
### 6.510.1.2 valueString()

```
QString Digikam::DSliderSpinBox::valueString ( ) const [override], [protected], [virtual]
```

Implements [Digikam::DAbstractSliderSpinBox](#).

## 6.511 Digikam::DSplashScreen Class Reference

Inheritance diagram for Digikam::DSplashScreen:



### Public Member Functions

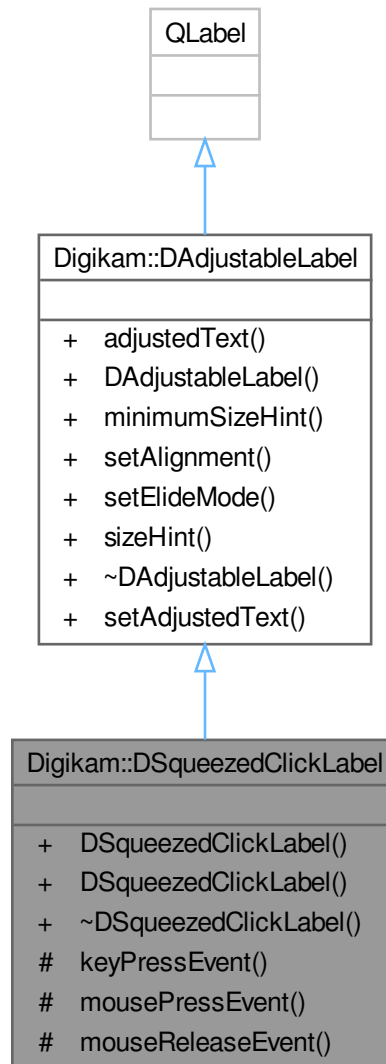
- void **setColor** (const QColor &color)
- void **setMessage** (const QString &message)

### Protected Member Functions

- void **drawContents** (QPainter \*) override

## 6.512 Digikam::DSqueezedClickLabel Class Reference

Inheritance diagram for Digikam::DSqueezedClickLabel:



### Signals

- void **activated** ()
- void **leftClicked** ()

### Public Member Functions

- **DSqueezedClickLabel** (const QString &text, QWidget \*const parent=nullptr)
- **DSqueezedClickLabel** (QWidget \*const parent=nullptr)



**Public Member Functions inherited from [Digikam::DAdjustableLabel](#)**

- QString **adjustedText** () const
- **DAdjustableLabel** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setAlignment** (Qt::Alignment align)
- void **setElideMode** (Qt::TextElideMode mode)
- QSize **sizeHint** () const override

**Protected Member Functions**

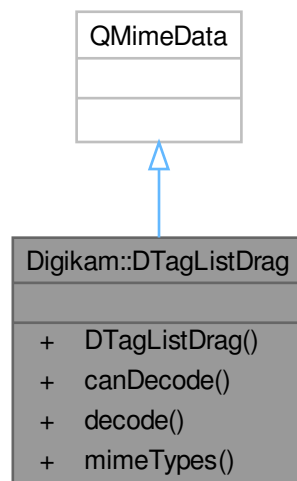
- void **keyPressEvent** (QKeyEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override

**Additional Inherited Members****Public Slots inherited from [Digikam::DAdjustableLabel](#)**

- void **setAdjustedText** (const QString &text=QString())

**6.513 Digikam::DTagListDrag Class Reference**

Inheritance diagram for Digikam::DTagListDrag:

**Public Member Functions**

- **DTagListDrag** (const QList< int > &tagIDs)

### Static Public Member Functions

- static bool **canDecode** (const QMimeData \*e)
- static bool **decode** (const QMimeData \*e, QList< int > &tagIDs)
- static QStringList **mimeTypes** ()

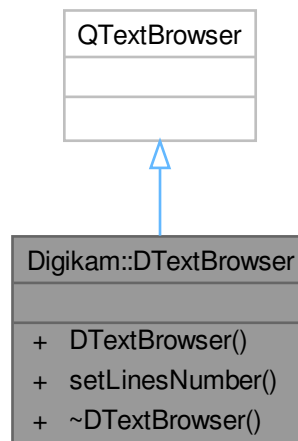
### 6.513.1 Detailed Description

Provides a drag object for a list of tags

When a tag is moved through drag'n'drop an object of this class is created.

## 6.514 Digikam::DTextBrowser Class Reference

Inheritance diagram for Digikam::DTextBrowser:

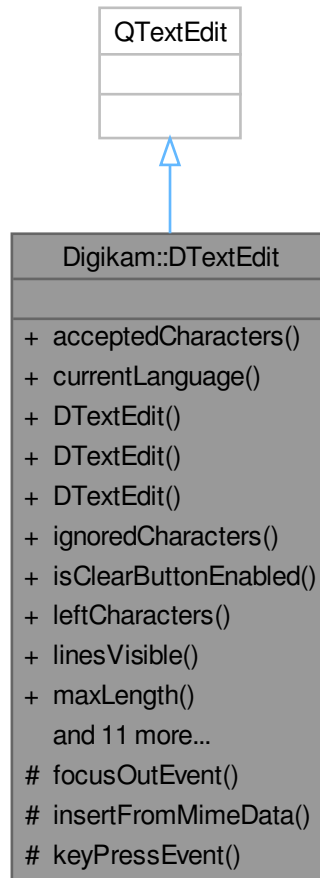


### Public Member Functions

- **DTextBrowser** (const QString &text, QWidget \*const parent=nullptr)
- void **setLinesNumber** (int l)

## 6.515 Digikam::DTextEdit Class Reference

Inheritance diagram for Digikam::DTextEdit:



### Classes

- class [Private](#)

### Signals

- void `editingFinished()`
- void `returnPressed()`
- void `textEdited(const QString &)`

## Public Member Functions

- QString [acceptedCharacters](#) () const
- QString [currentLanguage](#) () const
- [DTextEdit](#) (const QString &contents, QWidget \*const parent=nullptr)
- [DTextEdit](#) (QWidget \*const parent=nullptr)
- [DTextEdit](#) (unsigned int lines, QWidget \*const parent=nullptr)
- QString [ignoredCharacters](#) () const
- bool [isClearButtonEnabled](#) () const
- int [leftCharacters](#) () const
- unsigned int [linesVisible](#) () const
- int [maxLength](#) () const
- void [setAcceptedCharacters](#) (const QString &mask)
- void [setClearButtonEnabled](#) (bool enable)
- void [setCurrentLanguage](#) (const QString &lang)
- void [setIgnoredCharacters](#) (const QString &mask)
- void [setLinesVisible](#) (unsigned int lines)
- void [setLocalizeSettings](#) (const [LocalizeContainer](#) &settings)
- void [setMaxLength](#) (int length)
- void [setText](#) (const QString &text)
- [LocalizeContainer](#) [spellCheckSettings](#) () const
- QString [text](#) () const
- [~DTextEdit](#) () override

## Protected Member Functions

- void [focusOutEvent](#) (QFocusEvent \*e) override
- void [insertFromMimeData](#) (const QMimeData \*source) override
- void [keyPressEvent](#) (QKeyEvent \*e) override

### 6.515.1 Detailed Description

A text edit widget based on QTextEdit with spell checker capabilities based on Sonnet (optional). Widget size can be constrained with the number of visible lines. A single line constraint will emulate QLineEdit. See [setLinesVisible\(\)](#) for details. The maximum number of characters can be limited with [setMaxLenght\(\)](#). The characters can be limited in editor by [setIgnoredCharacters\(\)](#) and [setAcceptedCharacters\(\)](#). Implementation: dtextedit.cpp

### 6.515.2 Constructor & Destructor Documentation

#### 6.515.2.1 DTextEdit() [1/3]

```
Digikam::DTextEdit::DTextEdit (
    QWidget *const parent = nullptr ) [explicit]
```

Default constructor.

#### 6.515.2.2 DTextEdit() [2/3]

```
Digikam::DTextEdit::DTextEdit (
    unsigned int lines,
    QWidget *const parent = nullptr ) [explicit]
```

Constructor with a number of lines. Zero lines do not apply a size constraint.

### 6.515.2.3 DTextEdit() [3/3]

```
Digikam::DTextEdit::DTextEdit (
    const QString & contents,
    QWidget *const parent = nullptr ) [explicit]
```

Constructor with text contents to use.

### 6.515.2.4 ~DTextEdit()

```
Digikam::DTextEdit::~DTextEdit ( ) [override]
```

Standard destructor.

## 6.515.3 Member Function Documentation

### 6.515.3.1 acceptedCharacters()

```
QString Digikam::DTextEdit::acceptedCharacters ( ) const
```

This property holds whether the edit widget handle the mask of accepted characters in text editor. The mask of characters is passed as string (ex: "abcABC"). By default the mask is empty.

### 6.515.3.2 ignoredCharacters()

```
QString Digikam::DTextEdit::ignoredCharacters ( ) const
```

This property holds whether the edit widget handle the mask of ignored characters in text editor. The mask of characters is passed as string (ex: "+!()"). By default the mask is empty.

### 6.515.3.3 isClearButtonEnabled()

```
bool Digikam::DTextEdit::isClearButtonEnabled ( ) const
```

This property holds whether the edit widget displays a clear button when it is not empty. If enabled, the edit widget displays a trailing clear button when it contains some text, otherwise the edit widget does not show a clear button. This option only take effect in QLineEdit emulation mode when lines visible is set to 1. See [setLinesVisible\(\)](#) for details.

### 6.515.3.4 leftCharacters()

```
int Digikam::DTextEdit::leftCharacters ( ) const
```

Return the left characters that user can enter if a limit have been previously set with [setMaxLeght\(\)](#).

### 6.515.3.5 returnPressed

```
void Digikam::DTextEdit::returnPressed ( ) [signal]
```

Emitted only when mimic QLineEdit mode is enabled. See [setLinesVisible\(\)](#) for details.

### 6.515.3.6 setCurrentLanguage()

```
void Digikam::DTextEdit::setCurrentLanguage (
    const QString & lang )
```

This property holds whether the edit widget handle a specific spell-checker language (2 letters code based as "en", "fr", "es", etc.). If this property is not set, spell-checker will try to auto-detect language by parsing the text. To reset this setting, pass a empty string as language. If Sonnet dependencies is not resolved, these method do nothing.

### 6.515.3.7 setLinesVisible()

```
void Digikam::DTextEdit::setLinesVisible (
    unsigned int lines )
```

This property holds whether the edit widget handle visible lines used by the widget to show text. Lines must be superior or equal to 1 to apply a size constraint. Notes: if a single visible line is used, the widget will emulate QLineEdit. a null value do not apply a size constraint.

### 6.515.3.8 setMaxLength()

```
void Digikam::DTextEdit::setMaxLength (
    int length )
```

This property holds whether the edit widget handle the maximum of characters that user can enter in editor. By default no limit is set. A zero length reset a limit.

### 6.515.3.9 spellCheckSettings()

```
LocalizeContainer Digikam::DTextEdit::spellCheckSettings ( ) const
```

This property holds whether the edit widget handle the Spellcheck settings. See [LocalizeContainer](#) class for details.

### 6.515.3.10 text()

```
QString Digikam::DTextEdit::text ( ) const
```

This property holds whether the edit widget handle text contents as plain text. If ignored or accepted characters masks are set, text is filtered accordingly.

## 6.516 Digikam::DTextEdit::Private Class Reference

### Public Member Functions

- void `init` (`DTextEdit` \*const parent)

### Public Attributes

- QString `acceptedMask`  
*Mask of accepted characters in text editor.*
- bool `clearBtnEnable` = true
- `DTextEditClearButton` \* `clrBtn` = nullptr
- `LocalizeContainer` `container`  
*Spell checking settings container.*
- QString `ignoredMask`  
*Mask of ignored characters in text editor.*
- unsigned int `lines` = 3
- int `maxLength` = 0

### 6.516.1 Member Function Documentation

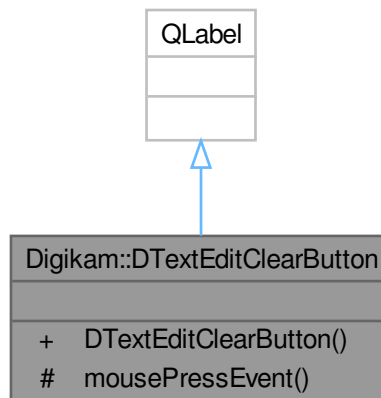
#### 6.516.1.1 `init()`

```
void Digikam::DTextEdit::Private::init (
    DTextEdit *const parent )
```

Init the text widget with the spell-checker engine (optional).

## 6.517 Digikam::DTextEditClearButton Class Reference

Inheritance diagram for Digikam::DTextEditClearButton:



**Signals**

- void **clicked** ()

**Public Member Functions**

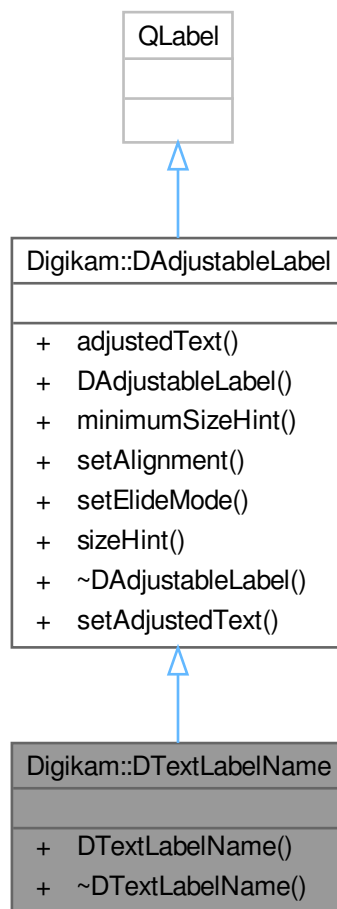
- **DTextEditClearButton** (QWidget \*const parent)

**Protected Member Functions**

- void **mousePressEvent** (QMouseEvent \*e) override

**6.518 Digikam::DTextLabelName Class Reference**

Inheritance diagram for Digikam::DTextLabelName:





### Public Member Functions

- **DTextLabelName** (const QString &name, QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DAdjustableLabel](#)

- QString **adjustedText** () const
- **DAdjustableLabel** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setAlignment** (Qt::Alignment align)
- void **setElideMode** (Qt::TextElideMode mode)
- QSize **sizeHint** () const override

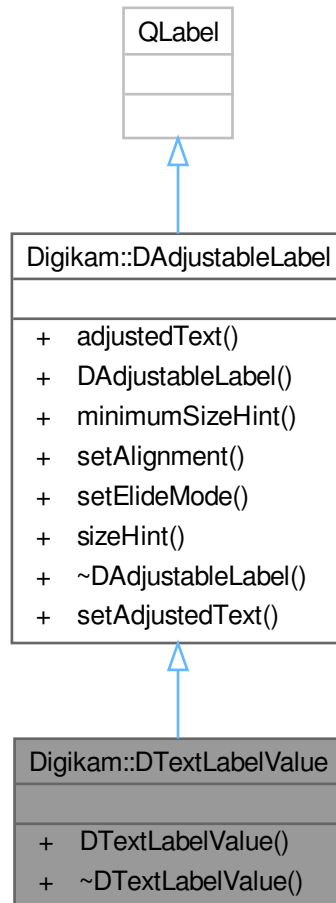
### Additional Inherited Members

### Public Slots inherited from [Digikam::DAdjustableLabel](#)

- void **setAdjustedText** (const QString &text=QString())

## 6.519 Digikam::DTextLabelValue Class Reference

Inheritance diagram for Digikam::DTextLabelValue:



### Public Member Functions

- **DTextLabelValue** (const QString &value, QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DAdjustableLabel](#)

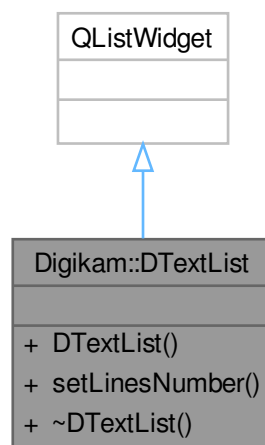
- QString **adjustedText** () const
- **DAdjustableLabel** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setAlignment** (Qt::Alignment align)
- void **setElideMode** (Qt::TextElideMode mode)
- QSize **sizeHint** () const override

**Additional Inherited Members****Public Slots inherited from [Digikam::DAdjustableLabel](#)**

- void **setAdjustedText** (const QString &text=QString())

**6.520 Digikam::DTextList Class Reference**

Inheritance diagram for Digikam::DTextList:

**Public Member Functions**

- **DTextList** (const QStringList &list, QWidget \*const parent=nullptr)
- void **setLinesNumber** (int l)

**6.521 Digikam::DToolTipStyleSheet Class Reference****Public Member Functions**

- QString **breakString** (const QString &input) const
- **DToolTipStyleSheet** (const QFont &font=QFontDatabase::systemFont(QFontDatabase::GeneralFont))
- QString **elidedText** (const QString &input, Qt::TextElideMode mode) const
- QString **imageAsBase64** (const QImage &img) const

**Public Attributes**

- QString **cellBeg**
- QString **cellEnd**
- QString **cellMid**
- QString **cellSpecBeg**
- QString **cellSpecEnd**
- QString **cellSpecMid**
- QString **headBeg**
- QString **headEnd**
- const int **maxStringLength**
- QString **tipFooter**
- QString **tipHeader**
- QString **unavailable**

**6.522 Digikam::DTrash Class Reference****Static Public Member Functions**

- static bool [deleteDirRecursivley](#) (const QString &dirToDelete, const QDateTime &deleteTime)  
*Deletes a whole folder from the collection.*
- static bool [deleteImage](#) (const QString &imagePath, const QDateTime &deleteTime)  
*Deletes image to the trash of a particular collection.*
- static void [extractJsonForItem](#) (const QString &collPath, const QString &baseName, [DTrashItemInfo](#) &item↔Info)  
*Extracts the data from json file and gives it to [DTrashItemInfo](#).*

**Static Public Attributes**

- static const QString **DELETIONTIMESTAMP\_JSON\_KEY** = QLatin1String("deletiontimestamp")
- static const QString **FILES\_FOLDER** = QLatin1String("files")
- static const QString **IMAGEID\_JSON\_KEY** = QLatin1String("imageid")
- static const QString **INFO\_FILE\_EXTENSION** = QLatin1String(".dtrashinfo")
- static const QString **INFO\_FOLDER** = QLatin1String("info")
- static const QString **PATH\_JSON\_KEY** = QLatin1String("path")
- static const QString **TRASH\_FOLDER** = QLatin1String(".dtrash")

**6.522.1 Member Function Documentation****6.522.1.1 deleteDirRecursivley()**

```
bool Digikam::DTrash::deleteDirRecursivley (
    const QString & dirToDelete,
    const QDateTime & deleteTime ) [static]
```

**Parameters**

<i>dirToDelete</i>	path to folder
<i>deleteTime</i>	delete time from the image

**Returns**

true if folder was deleted

**6.522.1.2 deleteImage()**

```
bool Digikam::DTrash::deleteImage (
    const QString & imagePath,
    const QDateTime & deleteTime ) [static]
```

**Parameters**

<i>imagePath</i>	path to image
<i>deleteTime</i>	delete time from the image

**Returns**

true if the image was deleted

**6.522.1.3 extractJsonForItem()**

```
void Digikam::DTrash::extractJsonForItem (
    const QString & collPath,
    const QString & baseName,
    DTrashItemInfo & itemInfo ) [static]
```

**Parameters**

<i>collPath</i>	path to collection
<i>baseName</i>	name of the file inside the trash
<i>itemInfo</i>	item to extract data to it

**6.523 Digikam::DTrashItemInfo Class Reference****Public Member Functions**

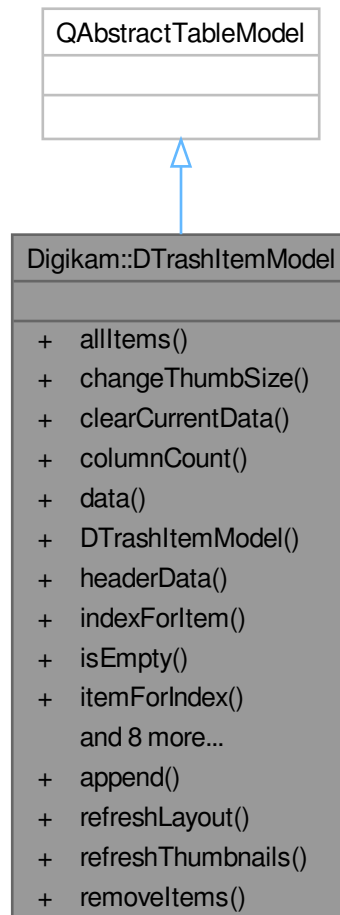
- bool **isNull** () const
- bool **operator==** (const [DTrashItemInfo](#) &itemInfo) const

**Public Attributes**

- QString **collectionPath**
- QString **collectionRelativePath**
- QDateTime **deletionTimestamp**
- qlonglong **imageId** = -1
- QString **jsonFilePath**
- QString **trashPath**

## 6.524 Digikam::DTrashItemModel Class Reference

Inheritance diagram for Digikam::DTrashItemModel:



### Public Types

- enum `DTrashColumn` { `DTrashThumb = 0` , `DTrashRelPath` , `DTrashTimeStamp` , `DTrashNumCol` }

### Public Slots

- void `append` (const `DTrashItemInfo` &itemInfo)  
*appends item to model data and informs the view*
- void `refreshLayout` ()  
*refreshes the view layout*
- void `refreshThumbnails` (const `LoadingDescription` &desc, const `QPixmap` &pix)  
*refreshes the thumbnails*
- void `removeItems` (const `QModelIndexList` &indexes)  
*removes list of items for given indexes from model data and informs the view*

## Signals

- void **dataChange** ()
- void **signalLoadingFinished** ()
- void **signalLoadingStarted** ()

## Public Member Functions

- DTrashItemInfoList **allItems** ()  
*returns a list of all items in model*
- void **changeThumbSize** (int size)  
*Changes the thumbnail size.*
- void **clearCurrentData** ()  
*Clears all data from model and informs the view.*
- int **columnCount** (const QModelIndex &) const override
- QVariant **data** (const QModelIndex &index, int role) const override
- **DTrashItemModel** (QObject \*const parent, QWidget \*const widget)
- QVariant **headerData** (int section, Qt::Orientation orientation, int role) const override
- QModelIndex **indexForItem** (const DTrashItemInfo &itemInfo) const  
*returns the index for the DTrashItemInfo in model*
- bool **isEmpty** ()
- DTrashItemInfo **itemForIndex** (const QModelIndex &index)  
*returns DTrashItemInfo for specific index in model*
- DTrashItemInfoList **itemsForIndexes** (const QList< QModelIndex > &indexes)  
*returns DTrashItemInfoList for given indexes in model*
- void **loadItemsForCollection** (const QString &colPath)  
*Runs a thread to list all items from a collection trash.*
- bool  **pixmapForItem** (const QString &path, QPixmap &pix) const  
*loads a thumbnail for item in trash for showing*
- int **rowCount** (const QModelIndex &) const override  
*QAbstractItemModel interface.*
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- void **stopLoadingTrash** ()  
*Stop loading of trash.*
- QString **trashAlbumPath** () const

## 6.524.1 Member Function Documentation

### 6.524.1.1 append

```
void Digikam::DTrashItemModel::append (
    const DTrashItemInfo & itemInfo ) [slot]
```

#### Parameters

<i>itemInfo</i>	item to append
-----------------	----------------

### 6.524.1.2 changeThumbSize()

```
void Digikam::DTrashItemModel::changeThumbSize (
    int size )
```

#### Parameters

<i>size</i>	size to change to
-------------	-------------------

### 6.524.1.3 isEmpty()

```
bool Digikam::DTrashItemModel::isEmpty ( )
```

#### Returns

true if there is no data in the model

### 6.524.1.4 loadItemsForCollection()

```
void Digikam::DTrashItemModel::loadItemsForCollection (
    const QString & colPath )
```

#### Parameters

<i>colPath</i>	path to collection to load items for
----------------	--------------------------------------

### 6.524.1.5 pixmapForItem()

```
bool Digikam::DTrashItemModel::pixmapForItem (
    const QString & path,
    QPixmap & pix ) const
```

#### Parameters

<i>path</i>	path of image in trash
<i>pix</i>	Pixmap to fill

#### Returns

true if there is an available thumbnail

### 6.524.1.6 refreshThumbnails

```
void Digikam::DTrashItemModel::refreshThumbnails (
    const LoadingDescription & desc,
    const QPixmap & pix ) [slot]
```



## Parameters

<i>desc</i>	loading description from thumbnail load thread
<i>pix</i>	pixmap from thumbnail load thread

**6.524.1.7 removeItems**

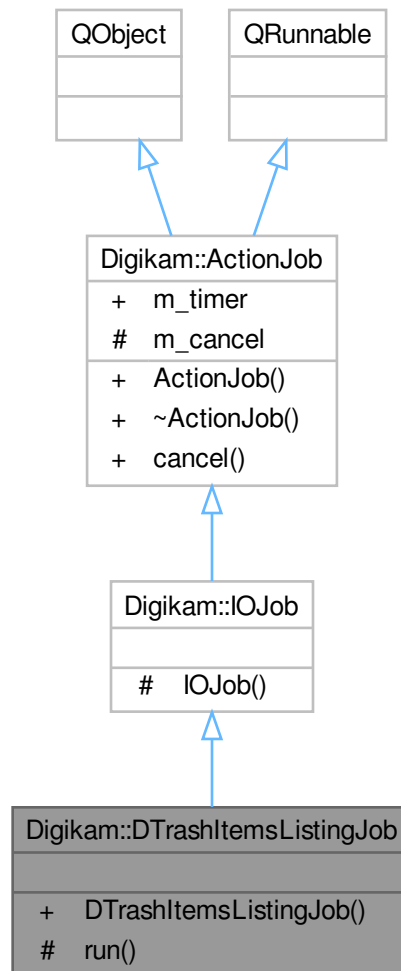
```
void Digikam::DTrashItemModel::removeItems (  
    const QModelIndexList & indexes ) [slot]
```

## Parameters

<i>indexes</i>	indexes to remove
----------------	-------------------

## 6.525 Digikam::DTrashItemsListingJob Class Reference

Inheritance diagram for Digikam::DTrashItemsListingJob:



### Signals

- void **trashItemInfo** (const [DTrashItemInfo](#) &info)

### Signals inherited from [Digikam::IOJob](#)

- void **signalError** (const QString &errMsg)
- void **signalOneProcessed** (const QUrl &url)

### Signals inherited from [Digikam::ActionJob](#)

- void **signalDone** ()
- void **signalProgress** (int)
- void **signalStarted** ()

### Public Member Functions

- [DTrashItemsListingJob](#) (const QString &collectionPath)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Protected Member Functions

- void [run](#) () override

### Additional Inherited Members

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Public Attributes inherited from [Digikam::ActionJob](#)

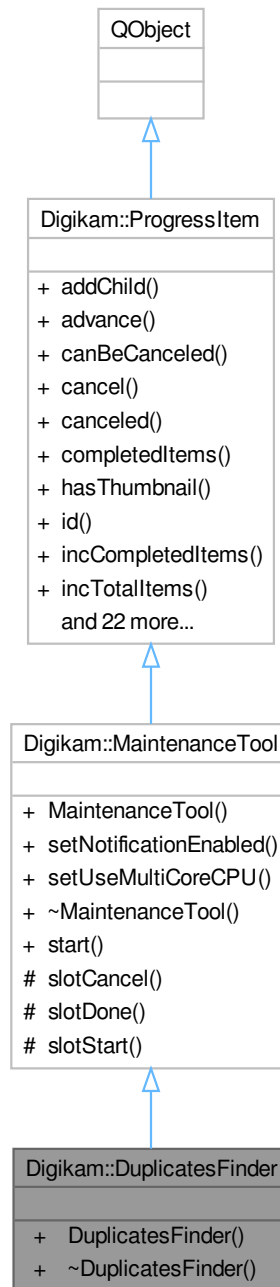
- QElapsedTimer [m\\_timer](#)

### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.526 Digikam::DuplicatesFinder Class Reference

Inheritance diagram for Digikam::DuplicatesFinder:



### Signals

- void **signalScanNotification** (const QString &msg, int type)

## Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

## Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void [progressItemCanceledById](#) (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- [DuplicatesFinder](#) (const AlbumList &albums, const AlbumList &tags, int albumTagRelation=0, int min←→ Similarity=90, int maxSimilarity=100, int searchResultRestriction=0, [Haarface::RefImageSelMethod](#) method=[Haarface::RefImageSelMethod::OlderOrLarger](#), const AlbumList &referenceImageAlbum={}, [ProgressItem](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- [MaintenanceTool](#) (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)
- virtual void [setUseMultiCoreCPU](#) (bool)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool **advance** (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool **canBeCanceled** () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool **hasThumbnail** () const
- const QString & **id** () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & **label** () const
- [ProgressItem](#) \* **parent** () const
- unsigned int **progress** () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool canBeCanceled, bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)
- void **setProgress** (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void **setShowAtStart** (bool showAtStart)
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void **setStatus** (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void **setThumbnail** (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void **setUsesBusyIndicator** (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool **showAtStart** () const
- const QString & **status** () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool **usesBusyIndicator** () const

## Additional Inherited Members

## Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

## Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

### 6.526.1 Constructor & Destructor Documentation

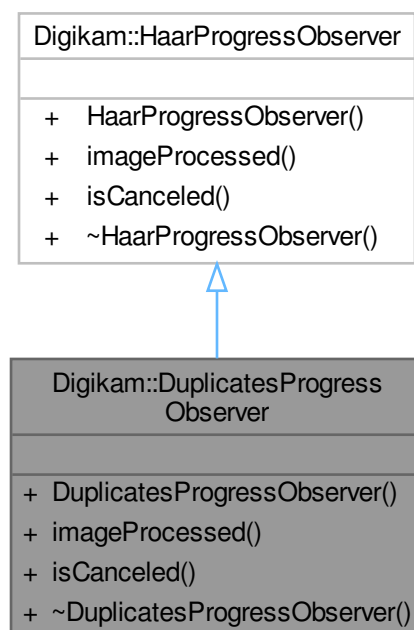
#### 6.526.1.1 DuplicatesFinder()

```
Digikam::DuplicatesFinder::DuplicatesFinder (
    const AlbumList & albums,
    const AlbumList & tags,
    int albumTagRelation = 0,
    int minSimilarity = 90,
    int maxSimilarity = 100,
    int searchResultRestriction = 0,
    HaarIface::RefImageSelMethod method = HaarIface::RefImageSelMethod::OlderOrLarger,
    const AlbumList & referenceImageAlbum = {},
    ProgressItem *const parent = nullptr )
```

Version to find all duplicates over a specific list to PAlbums and TAlbums

## 6.527 Digikam::DuplicatesProgressObserver Class Reference

Inheritance diagram for Digikam::DuplicatesProgressObserver:



## Public Member Functions

- **DuplicatesProgressObserver** ([SearchesJob](#) \*const thread)
- void **imageProcessed** (const [ItemInfo](#) &inf, const QImage &img, int dup) override
- bool **isCanceled** () override

## 6.527.1 Member Function Documentation

### 6.527.1.1 imageProcessed()

```
void Digikam::DuplicatesProgressObserver::imageProcessed (
    const ItemInfo & inf,
    const QImage & img,
    int dup ) [override], [virtual]
```

Implements [Digikam::HaarProgressObserver](#).

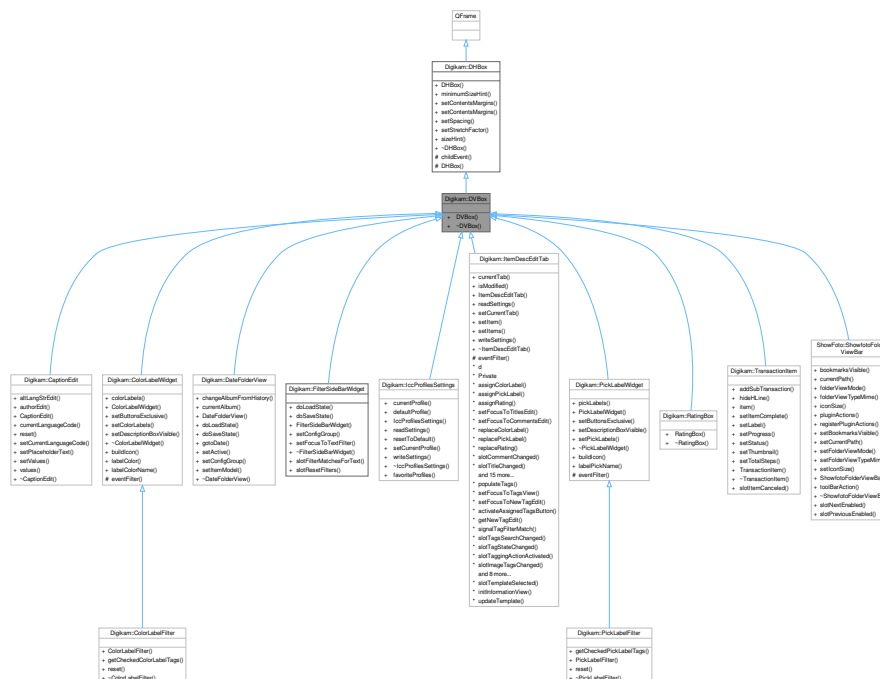
### 6.527.1.2 isCanceled()

```
bool Digikam::DuplicatesProgressObserver::isCanceled ( ) [override], [virtual]
```

Reimplemented from [Digikam::HaarProgressObserver](#).

## 6.528 Digikam::DVBox Class Reference

Inheritance diagram for [Digikam::DVBox](#):





### Public Member Functions

- **DVBox** (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentMargins** (const QMargins &margins)
- void **setContentMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DHBox](#)

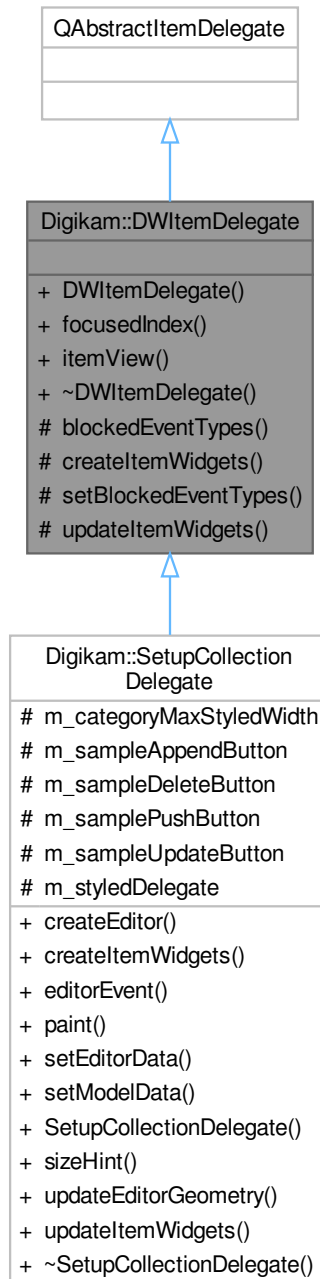
- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.528.1 Detailed Description

A Vertical widget to host children widgets

## 6.529 Digikam::DWItemDelegate Class Reference

Inheritance diagram for Digikam::DWItemDelegate:



### Public Member Functions

- [DWItemDelegate](#) (`QAbstractItemView *const itemView`, `QObject *const parent=nullptr`)
- `QPersistentModelIndex` [focusedIndex](#) () const
- `QAbstractItemView *` [itemView](#) () const

## Protected Member Functions

- `QList< QEvent::Type > blockedEventTypes (QWidget *const widget) const`
- `virtual QList< QWidget * > createItemWidgets (const QModelIndex &index) const =0`
- `void setBlockedEventTypes (QWidget *const widget, const QList< QEvent::Type > &types) const`
- `virtual void updateItemWidgets (const QList< QWidget * > &widgets, const QStyleOptionViewItem &option, const QModelIndex &index) const =0`

## Friends

- class `DWItemDelegateEventListener`
- class `DWItemDelegatePool`

## 6.529.1 Detailed Description

This class allows to create item delegates embedding simple widgets to interact with items. For instance you can add push buttons, line edits, etc. to your delegate and use them to modify the state of your model.

## 6.529.2 Constructor & Destructor Documentation

### 6.529.2.1 DWItemDelegate()

```
Digikam::DWItemDelegate::DWItemDelegate (
    QAbstractItemView *const itemView,
    QObject *const parent = nullptr ) [explicit]
```

Creates a new [ItemDelegate](#) to be used with a given itemview.

#### Parameters

<i>itemView</i>	the item view the new delegate will monitor
<i>parent</i>	the parent of this delegate

## 6.529.3 Member Function Documentation

### 6.529.3.1 blockedEventTypes()

```
QList< QEvent::Type > Digikam::DWItemDelegate::blockedEventTypes (
    QWidget *const widget ) const [protected]
```

Retrieves the list of blocked event types for the given widget.

#### Parameters

<i>widget</i>	the specified widget.
---------------	-----------------------

**Returns**

the list of blocked event types, can be empty if no events are blocked.

**6.529.3.2 createItemWidgets()**

```
virtual QList< QWidget * > Digikam::DWItemDelegate::createItemWidgets (
    const QModelIndex & index ) const [protected], [pure virtual]
```

Creates the list of widgets needed for an item.

**Note**

No initialization of the widgets is supposed to happen here. The widgets will be initialized based on needs for a given item.

If you want to connect some widget signals to any slot, you should do it here.

- index the index to create widgets for.

**Note**

If you want to know the index for which you are creating widgets, it is available as a QModelIndex [Q\\_↔](#) PROPERTY called "goya:creatingWidgetsForIndex". Ensure to add `Q_DECLARE_METATYPE(QModelIndex)` before your method definition to tell QVariant about QModelIndex.

**Returns**

the list of newly created widgets which will be used to interact with an item.

**See also**

[updateItemWidgets\(\)](#)

Implemented in [Digikam::SetupCollectionDelegate](#).

**6.529.3.3 focusedIndex()**

```
QPersistentModelIndex Digikam::DWItemDelegate::focusedIndex ( ) const
```

Retrieves the currently focused index. An invalid index if none is focused.

**Returns**

the current focused index, or `QPersistentModelIndex()` if none is focused.

### 6.529.3.4 itemView()

```
QAbstractItemView * Digikam::DWItemDelegate::itemView ( ) const
```

Retrieves the item view this delegate is monitoring.

#### Returns

the item view this delegate is monitoring

### 6.529.3.5 setBlockedEventTypes()

```
void Digikam::DWItemDelegate::setBlockedEventTypes (
    QWidget *const widget,
    const QList< QEvent::Type > & types ) const [protected]
```

Sets the list of event `types` that a widget will block.

Blocked events are not passed to the view. This way you can prevent an item from being selected when a button is clicked for instance.

#### Parameters

<i>widget</i>	the widget which must block events
<i>types</i>	the list of event types the widget must block

### 6.529.3.6 updateItemWidgets()

```
virtual void Digikam::DWItemDelegate::updateItemWidgets (
    const QList< QWidget * > & widgets,
    const QStyleOptionViewItem & option,
    const QPersistentModelIndex & index ) const [protected], [pure virtual]
```

Updates a list of widgets for its use inside of the delegate (painting or event handling).

#### Note

All the positioning and sizing should be done in item coordinates.

#### Warning

Do not make widget connections in here, since this method will be called very regularly.

#### Parameters

<i>widgets</i>	the widgets to update
<i>option</i>	the current set of style options for the view.
<i>index</i>	the model index of the item currently manipulated.

Implemented in [Digikam::SetupCollectionDelegate](#).

## 6.530 Digikam::DWItemDelegatePool Class Reference

### Public Types

- enum **UpdateWidgetsEnum** { **UpdateWidgets** = 0 , **NotUpdateWidgets** }

### Public Member Functions

- [DWItemDelegatePool](#) ([DWItemDelegate](#) \*const delegate)
- `QList< QWidget * >` [findWidgets](#) (const `QPersistentModelIndex` &index, const `QStyleOptionViewItem` &option, `UpdateWidgetsEnum` updateWidgets=`UpdateWidgets`) const  
*Returns the widget associated to `index` and `widget`.*
- void **fullClear** ()
- `QList< QWidget * >` **invalidIndexesWidgets** () const

### Friends

- class **DWItemDelegate**
- class **DWItemDelegatePrivate**

## 6.530.1 Constructor & Destructor Documentation

### 6.530.1.1 DWItemDelegatePool()

```
Digikam::DWItemDelegatePool::DWItemDelegatePool (
    DWItemDelegate *const delegate ) [explicit]
```

Creates a new `ItemDelegatePool`.

#### Parameters

<i>delegate</i>	the <a href="#">ItemDelegate</a> for this pool.
-----------------	---

## 6.530.2 Member Function Documentation

### 6.530.2.1 findWidgets()

```
QList< QWidget * > Digikam::DWItemDelegatePool::findWidgets (
    const QPersistentModelIndex & index,
    const QStyleOptionViewItem & option,
    UpdateWidgetsEnum updateWidgets = UpdateWidgets ) const
```

## Parameters

<i>index</i>	The index to search into.
<i>option</i>	a QStyleOptionViewItem.
<i>updateWidgets</i>	a flag to force to update widgets.

## Returns

A QList of the pointers to the widgets found.

## 6.531 Digikam::DWItemDelegatePoolPrivate Class Reference

### Public Member Functions

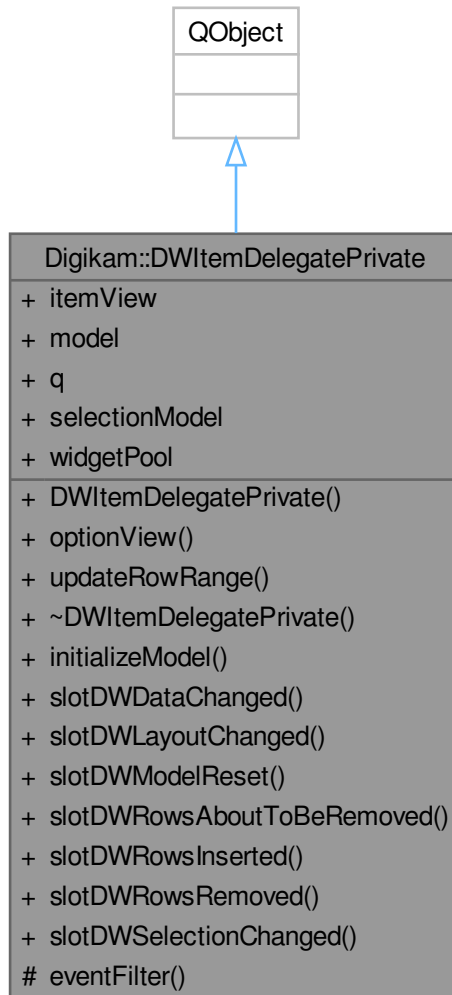
- **DWItemDelegatePoolPrivate** ([DWItemDelegate](#) \*const dd)

### Public Attributes

- bool **clearing** = false
- [DWItemDelegate](#) \* **delegate** = nullptr
- DWItemDelegateEventListener \* **eventListener** = nullptr
- QHash< QPersistentModelIndex, QList< QWidget \* > > **usedWidgets**
- QHash< QWidget \*, QPersistentModelIndex > **widgetInIndex**

## 6.532 Digikam::DWItemDelegatePrivate Class Reference

Inheritance diagram for Digikam::DWItemDelegatePrivate:



### Public Slots

- void **initializeModel** (const QModelIndex &parent=QModelIndex())
- void **slotDWDataChanged** (const QModelIndex &topLeft, const QModelIndex &bottomRight)
- void **slotDWLayoutChanged** ()
- void **slotDWModelReset** ()
- void **slotDWRowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end)
- void **slotDWRowsInserted** (const QModelIndex &parent, int start, int end)
- void **slotDWRowsRemoved** (const QModelIndex &parent, int start, int end)
- void **slotDWSelectionChanged** (const QItemSelection &selected, const QItemSelection &deselected)



**Public Member Functions**

- **DWItemDelegatePrivate** ([DWItemDelegate](#) \*const qq, QObject \*const parent=nullptr)
- QStyleOptionViewItem **optionView** (const QModelIndex &index)
- void **updateRowRange** (const QModelIndex &parent, int start, int end, bool isRemoving)

**Public Attributes**

- QAbstractItemView \* **itemView** = nullptr
- QAbstractItemModel \* **model** = nullptr
- [DWItemDelegate](#) \* **q** = nullptr
- QItemSelectionModel \* **selectionModel** = nullptr
- [DWItemDelegatePool](#) \* **widgetPool** = nullptr

**Protected Member Functions**

- bool **eventFilter** (QObject \*watched, QEvent \*event) override

**6.533 Digikam::DWizardDlg Class Reference**

Inheritance diagram for Digikam::DWizardDlg:

**Public Member Functions**

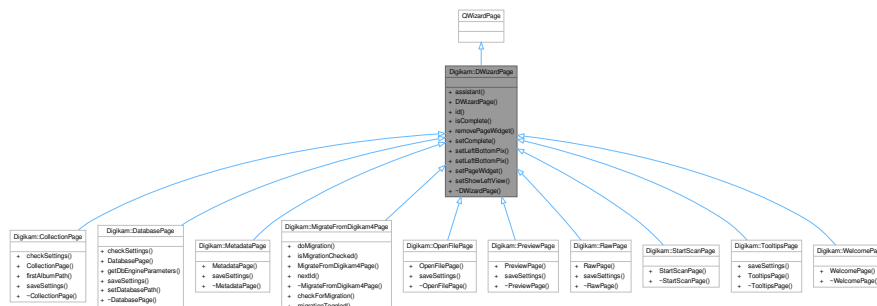
- **DWizardDlg** (QWidget \*const parent, const QString &objName)
- void **setPlugin** ([DPlugin](#) \*const tool)

## Protected Member Functions

- void **restoreDialogSize** ()
- void **saveDialogSize** ()
- void **showEvent** (QShowEvent \*) override

## 6.534 Digikam::DWizardPage Class Reference

Inheritance diagram for Digikam::DWizardPage:

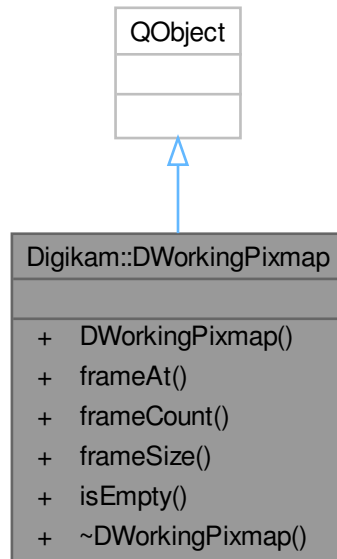


## Public Member Functions

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)
- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.535 Digikam::DWorkingPixmap Class Reference

Inheritance diagram for Digikam::DWorkingPixmap:



### Public Member Functions

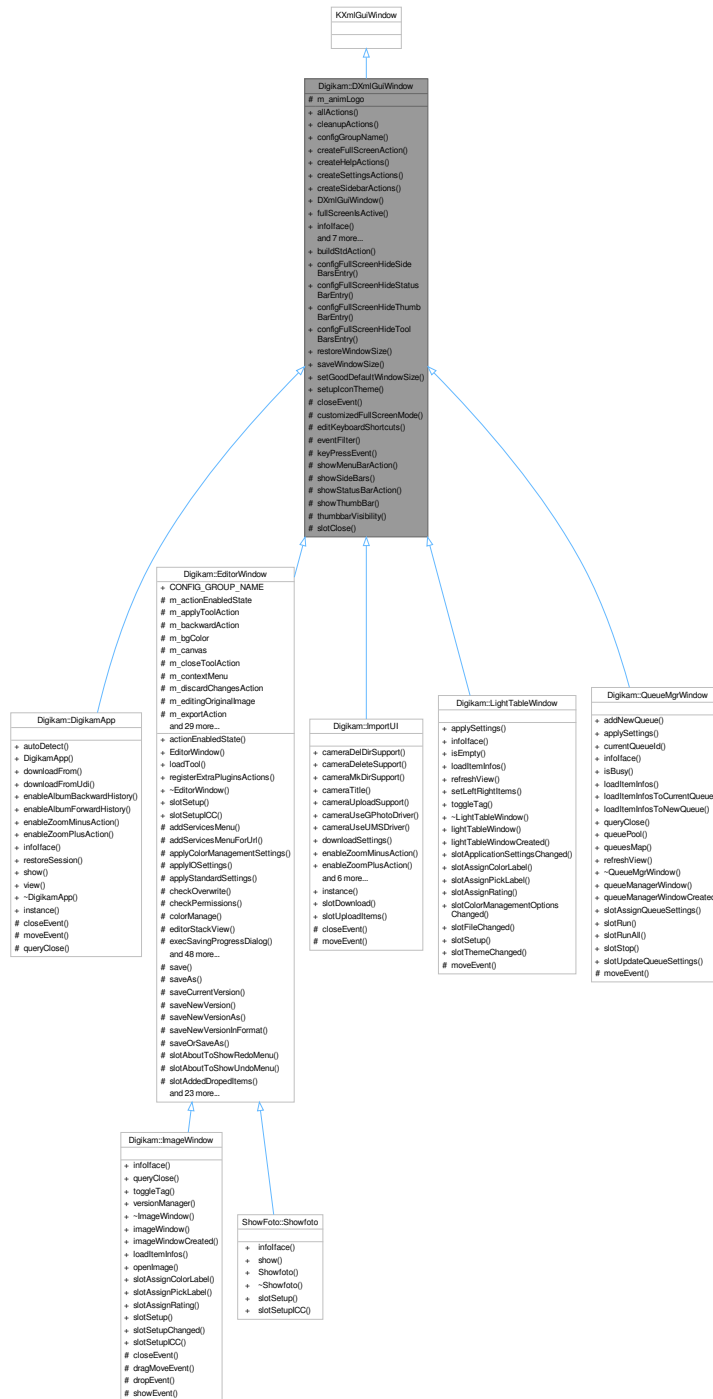
- **DWorkingPixmap** (QObject \*const parent=nullptr)
- QPixmap **frameAt** (int index) const
- int **frameCount** () const
- QSize **frameSize** () const
- bool **isEmpty** () const

### 6.535.1 Detailed Description

A widget to draw progress wheel indicator over thumbnails.

## 6.536 Digikam::DXmlGuiWindow Class Reference

Inheritance diagram for Digikam::DXmlGuiWindow:



### Classes

- class [Private](#)

## Public Member Functions

- `QList< QAction * > allActions () const`
- `void cleanupActions ()`
- `QString configGroupName () const`
- `void createFullScreenAction (const QString &name)`
- `void createHelpActions (const QString &handbookSection, bool coreOptions=true)`
- `void createSettingsActions ()`
- `void createSidebarActions ()`
- `DXmlGuiWindow (QWidget *const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())`
- `bool fullScreensActive () const`
- `virtual DInfoInterface * infoface (DPluginAction *const ac)=0`
- `void readFullScreenSettings (const KConfigGroup &group)`
- `virtual void registerExtraPluginsActions (QString &)`
- `void registerPluginsActions ()`
- `void setConfigGroupName (const QString &name)`
- `void setFullScreenOptions (int options)`
- `void unminimizeAndActivateWindow ()`

## Static Public Member Functions

- `static QAction * buildStdAction (StdActionType type, const QObject *const recvr, const char *const slot, QObject *const parent)`
- `static QString configFullScreenHideSideBarsEntry ()`
- `static QString configFullScreenHideStatusBarEntry ()`
- `static QString configFullScreenHideThumbBarEntry ()`
- `static QString configFullScreenHideToolBarsEntry ()`
- `static void restoreWindowSize (QWidget *const win, const KConfigGroup &group)`
- `static void saveWindowSize (QWidget *const win, KConfigGroup &group)`
- `static void setGoodDefaultWindowSize (QWidget *const win)`
- `static void setupIconTheme ()`

## Protected Slots

- `bool slotClose ()`

## Protected Member Functions

- `void closeEvent (QCloseEvent *e) override`
- `virtual void customizedFullScreenMode (bool set)`
- `void editKeyboardShortcuts (KActionCollection *const extraac=nullptr, const QString &actitle=QString())`
- `bool eventFilter (QObject *obj, QEvent *ev) override`
- `void keyPressEvent (QKeyEvent *e) override`
- `QAction * showMenuBarAction () const`
- `virtual void showSideBars (bool visible)`
- `QAction * showStatusBarAction () const`
- `virtual void showThumbBar (bool visible)`
- `virtual bool thumbbarVisibility () const`

## Protected Attributes

- `DLogoAction * m\_animLogo = nullptr`

## 6.536.1 Detailed Description

Generic class to use with all main window.

## 6.536.2 Member Function Documentation

### 6.536.2.1 allActions()

```
QList< QAction * > Digikam::DXmlGuiWindow::allActions ( ) const
```

Return all actions from internal collection.

### 6.536.2.2 cleanupActions()

```
void Digikam::DXmlGuiWindow::cleanupActions ( )
```

Cleanup unwanted actions from action collection.

### 6.536.2.3 configFullScreenHideToolBarsEntry()

```
static QString Digikam::DXmlGuiWindow::configFullScreenHideToolBarsEntry ( ) [inline], [static]
```

Shared with [FullScreenSettings](#).

### 6.536.2.4 createFullScreenAction()

```
void Digikam::DXmlGuiWindow::createFullScreenAction (
    const QString & name )
```

Create Full-screen action to action collection instance from managed window set through `setManagedWindow()`. This action is connected to `slotToggleFullScreen()` slot. 'name' is action name used in KDE UI rc file.

### 6.536.2.5 createHelpActions()

```
void Digikam::DXmlGuiWindow::createHelpActions (
    const QString & handbookSection,
    bool coreOptions = true )
```

Create common actions from Help menu for all digiKam main windows.

### 6.536.2.6 createSettingsActions()

```
void Digikam::DXmlGuiWindow::createSettingsActions ( )
```

Create common actions to setup all digiKam main windows.

### 6.536.2.7 createSidebarActions()

```
void Digikam::DXmlGuiWindow::createSidebarActions ( )
```

Create common actions to handle side-bar through keyboard shortcuts.

### 6.536.2.8 customizedFullScreenMode()

```
void Digikam::DXmlGuiWindow::customizedFullScreenMode (
    bool set ) [protected], [virtual]
```

Re-implement this method if you want to manage customized view visibility in full-screen mode. This method is called by `switchWindowToFullScreen()`. By default this method do nothing.

### 6.536.2.9 editKeyboardShortcuts()

```
void Digikam::DXmlGuiWindow::editKeyboardShortcuts (
    KActionCollection *const extraac = nullptr,
    const QString & actitle = QString() ) [protected]
```

Call this method from your main window to show keyboard shortcut config dialog with an extra action collection to configure. This method is called by `slotEditKeys()` which can be re-implement in child class for customization.

### 6.536.2.10 fullScreenIsActive()

```
bool Digikam::DXmlGuiWindow::fullScreenIsActive ( ) const
```

Return true if managed window is currently in Full Screen Mode.

### 6.536.2.11 infoIface()

```
virtual DInfoInterface * Digikam::DXmlGuiWindow::infoIface (
    DPluginAction *const ac ) [pure virtual]
```

Return the interface instance to access to items information.

Implemented in [Digikam::DigikamApp](#), [ShowFoto::Showfoto](#), [Digikam::ImageWindow](#), [Digikam::LightTableWindow](#), [Digikam::ImportUI](#), and [Digikam::QueueMgrWindow](#).

### 6.536.2.12 readFullScreenSettings()

```
void Digikam::DXmlGuiWindow::readFullScreenSettings (
    const KConfigGroup & group )
```

Read full-screen settings from KDE config file.

### 6.536.2.13 registerPluginsActions()

```
void Digikam::DXmlGuiWindow::registerPluginsActions ( )
```

Register all generic plugins action to this instance. Call registerExtraPluginsActions() to plug other kind of plugins in GUI.

### 6.536.2.14 setConfigGroupName()

```
void Digikam::DXmlGuiWindow::setConfigGroupName (
    const QString & name )
```

Manage config group name used by window instance to get/set settings from config file

### 6.536.2.15 setFullScreenOptions()

```
void Digikam::DXmlGuiWindow::setFullScreenOptions (
    int options )
```

Set full-screen options to managed window

### 6.536.2.16 setupIconTheme()

```
void Digikam::DXmlGuiWindow::setupIconTheme ( ) [static]
```

If we have some local breeze icon resource, prefer it.

### 6.536.2.17 showSideBars()

```
void Digikam::DXmlGuiWindow::showSideBars (
    bool visible ) [protected], [virtual]
```

Re-implement this method if you want to manage sidebars visibility in full-screen mode. By default this method do nothing.

### 6.536.2.18 showThumbBar()

```
void Digikam::DXmlGuiWindow::showThumbBar (
    bool visible ) [protected], [virtual]
```

Re-implement this method if you want to manage thumbbar visibility in full-screen mode. By default this method do nothing.

### 6.536.2.19 thumbbarVisibility()

```
bool Digikam::DXmlGuiWindow::thumbbarVisibility ( ) const [protected], [virtual]
```

Re-implement this method if managed window has a thumbbar. This must return visibility state of it.



## 6.537 Digikam::DXmlGuiWindow::Private Class Reference

### Public Attributes

- [DAboutData](#) \* **about** = nullptr
- [DLogoAction](#) \* **anim** = nullptr
- QString **configGroupName**
- QAction \* **dbStatAction** = nullptr
- bool **dirtyMainToolBar** = false
- int **fsOptions** = [FS\\_NONE](#)
- [KToggleFullScreenAction](#) \* **fullScreenAction** = nullptr
- [QToolButton](#) \* **fullScreenBtn** = nullptr
- bool **fullScreenHideSideBars** = false
- bool **fullScreenHideStatusBar** = false
- bool **fullScreenHideThumbBar** = true
- bool **fullScreenHideToolBars** = false
- [QWidget](#) \* **fullScreenParent** = nullptr
- QString **handbookSection**
- QAction \* **libsInfoAction** = nullptr
- bool **menubarVisibility** = true
- QAction \* **showMenuBarAction** = nullptr
- QAction \* **showStatusBarAction** = nullptr
- bool **statusbarVisibility** = true
- bool **thumbbarVisibility** = true
- [QMap](#)< [KToolBar](#) \*, bool > **toolbarsVisibility**

### 6.537.1 Member Data Documentation

#### 6.537.1.1 dirtyMainToolBar

```
bool Digikam::DXmlGuiWindow::Private::dirtyMainToolBar = false
```

Used by [slotToggleFullScreen\(\)](#) to manage state of full-screen button on managed window

#### 6.537.1.2 fsOptions

```
int Digikam::DXmlGuiWindow::Private::fsOptions = FS\_NONE
```

Full-Screen options. See [FullScreenOptions](#) enum and [setFullScreenOptions\(\)](#) for details.

#### 6.537.1.3 fullScreenAction

```
KToggleFullScreenAction* Digikam::DXmlGuiWindow::Private::fullScreenAction = nullptr
```

Action plug in managed window to switch fullscreen state

#### 6.537.1.4 fullScreenBtn

```
QToolButton* Digikam::DXmlGuiWindow::Private::fullScreenBtn = nullptr
```

Show only if toolbar is hidden

#### 6.537.1.5 fullScreenHideSideBars

```
bool Digikam::DXmlGuiWindow::Private::fullScreenHideSideBars = false
```

Settings taken from managed window configuration to handle toolbar visibility in full-screen mode

#### 6.537.1.6 fullScreenHideStatusBar

```
bool Digikam::DXmlGuiWindow::Private::fullScreenHideStatusBar = false
```

Settings taken from managed window configuration to handle statusbar visibility in full-screen mode

#### 6.537.1.7 fullScreenHideThumbBar

```
bool Digikam::DXmlGuiWindow::Private::fullScreenHideThumbBar = true
```

Settings taken from managed window configuration to handle thumbbar visibility in full-screen mode

#### 6.537.1.8 fullScreenHideToolBars

```
bool Digikam::DXmlGuiWindow::Private::fullScreenHideToolBars = false
```

Settings taken from managed window configuration to handle toolbar visibility in full-screen mode

#### 6.537.1.9 fullScreenParent

```
QWidget* Digikam::DXmlGuiWindow::Private::fullScreenParent = nullptr
```

Used under MacOS only. See bug #414117

#### 6.537.1.10 menubarVisibility

```
bool Digikam::DXmlGuiWindow::Private::menubarVisibility = true
```

Store previous visibility of menubar before full-screen mode.

#### 6.537.1.11 statusBarVisibility

```
bool Digikam::DXmlGuiWindow::Private::statusBarVisibility = true
```

Store previous visibility of statusbar before full-screen mode.

### 6.537.1.12 thumbbarVisibility

```
bool Digikam::DXmlGuiWindow::Private::thumbbarVisibility = true
```

Store previous visibility of thumbbar before full-screen mode.

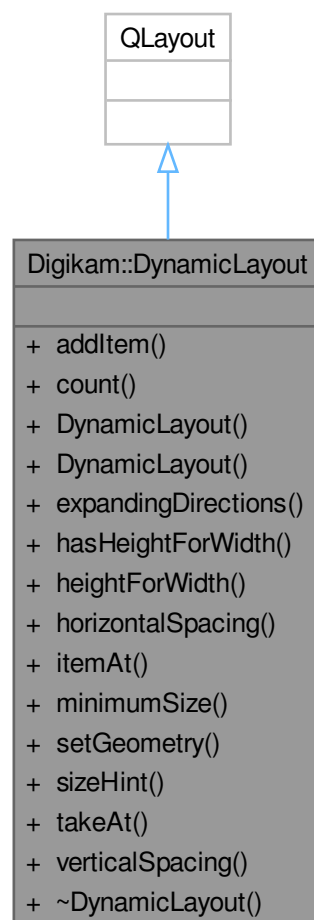
### 6.537.1.13 toolbarsVisibility

```
QMap<KToolBar*, bool> Digikam::DXmlGuiWindow::Private::toolbarsVisibility
```

Store previous visibility of toolbars before full-screen mode.

## 6.538 Digikam::DynamicLayout Class Reference

Inheritance diagram for Digikam::DynamicLayout:

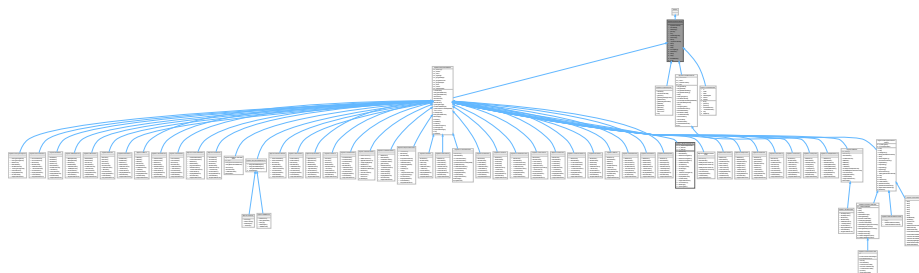


## Public Member Functions

- void **addItem** (QLayoutItem \*layItem) override
- int **count** () const override
- **DynamicLayout** (int margin=-1, int hSpacing=0, int vSpacing=0)
- **DynamicLayout** (QWidget \*const parent, int margin=-1, int hSpacing=0, int vSpacing=0)
- Qt::Orientations **expandingDirections** () const override
- bool **hasHeightForWidth** () const override
- int **heightForWidth** (int) const override
- int **horizontalSpacing** () const
- QLayoutItem \* **itemAt** (int index) const override
- QSize **minimumSize** () const override
- void **setGeometry** (const QRect &rect) override
- QSize **sizeHint** () const override
- QLayoutItem \* **takeAt** (int index) override
- int **verticalSpacing** () const

## 6.539 Digikam::DynamicThread Class Reference

Inheritance diagram for Digikam::DynamicThread:



## Public Types

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals

- void **finished** ()
- void **starting** ()

## Public Member Functions

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- virtual void **run** ()=0
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Protected Member Functions

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Friends

- class **DynamicThreadPriv**

## 6.539.1 Constructor & Destructor Documentation

### 6.539.1.1 DynamicThread()

```
Digikam::DynamicThread::DynamicThread (
    QObject *const parent = nullptr ) [explicit]
```

This class extends QRunnable, so you have to reimplement virtual void [run\(\)](#). In all aspects the class will act similar to a QThread.

### 6.539.1.2 ~DynamicThread()

```
Digikam::DynamicThread::~DynamicThread ( ) [override]
```

The destructor calls [stop\(\)](#) and [wait\(\)](#), but if you, in your destructor, delete any data that is accessed by your [run\(\)](#) method, you must call [stop\(\)](#) and [wait\(\)](#) before yourself.

## 6.539.2 Member Function Documentation

### 6.539.2.1 run()

```
virtual void Digikam::DynamicThread::run ( ) [pure virtual]
```

Implement this pure virtual function in your subclass.

Implemented in [Digikam::DImgThreadedFilter](#), [Digikam::ImageHistogram](#), [Digikam::LoadSaveThread](#), and [Digikam::ScanStateFilter](#).

### 6.539.2.2 `runningFlag()`

```
bool Digikam::DynamicThread::runningFlag ( ) const volatile [protected]
```

In you `run()` method, you shall regularly check for `runningFlag()` and cleanup and return if false.

### 6.539.2.3 `setPriority()`

```
void Digikam::DynamicThread::setPriority (
    QThread::Priority priority )
```

Sets the priority for this dynamic thread. Can be set anytime. If the thread is currently not running, the priority will be set when it is run next time. When you set `QThread::InheritPriority` (default), the priority is not changed but inherited from the thread pool.

### 6.539.2.4 `shutDown()`

```
void Digikam::DynamicThread::shutDown ( ) [protected]
```

If you are deleting data in your destructor which is accessed from the thread, do one of the following from your destructor to guarantee a safe shutdown: 1) Call this method 2) Call `stop()` and `wait()`, knowing that nothing will call `start()` anymore after this 3) Be sure the thread will never be running at destruction. Note: This irrevocably stops this object. Note: It is not sufficient that your parent class does this. Calling this method, or providing one of the above mentioned equivalent guarantees, must be done by every single last class in the hierarchy with an implemented destructor deleting data. (the base class destructor is always called after the derived class)

### 6.539.2.5 `start()`

```
void Digikam::DynamicThread::start (
    QMutexLocker< QMutex > & locker ) [protected]
```

Doing the same as `start()`, `stop()` and `wait` above, provide it with a locked `QMutexLocker` on `mutex()`. Note the `start()` will unlock and relock for scheduling once, after state change.

### 6.539.2.6 `starting`

```
void Digikam::DynamicThread::starting ( ) [signal]
```

Emitted if `emitSignals` is enabled

### 6.539.2.7 `stop`

```
void Digikam::DynamicThread::stop ( ) [slot]
```

Stop computation, sets the running flag to false.

### 6.539.2.8 threadMutex()

```
QMutex * Digikam::DynamicThread::threadMutex ( ) const [protected]
```

This is the non-recursive mutex used to protect state variables and waiting in this class. You can use it if you want to protect your memory in the same scope as calling `start`, `stop` or `wait`, then using the `QMutexLocker` variants below. Note that when you have locked this mutex, you must use these variants, as the mutex is non-recursive.

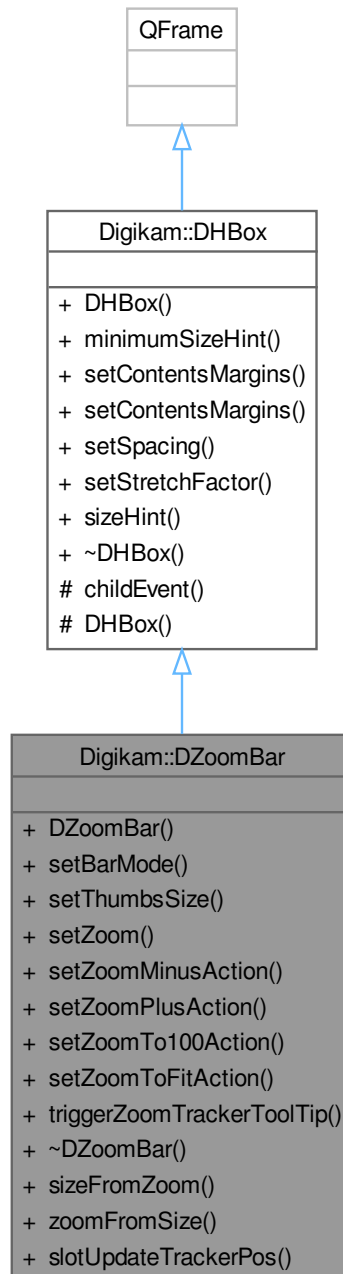
### 6.539.2.9 wait

```
void Digikam::DynamicThread::wait ( ) [slot]
```

Waits until the thread finishes. Typically, call `stop()` before.

## 6.540 Digikam::DZoomBar Class Reference

Inheritance diagram for Digikam::DZoomBar:



### Public Types

- enum `BarMode` { `PreviewZoomCtrl` = 0 , `ThumbsSizeCtrl` , `NoPreviewZoomCtrl` }



## Public Slots

- void **slotUpdateTrackerPos** ()

## Signals

- void **signalDelayedZoomSliderChanged** (int)
- void **signalZoomSliderChanged** (int)
- void **signalZoomSliderReleased** (int)
- void **signalZoomValueEdited** (double)

## Public Member Functions

- **DZoomBar** (QWidget \*const parent=nullptr)
- void **setBarMode** ([BarMode](#) mode)
- void **setThumbsSize** (int size)
- void **setZoom** (double zoom, double zmin, double zmax)
- void **setZoomMinusAction** (QAction \*const action)
- void **setZoomPlusAction** (QAction \*const action)
- void **setZoomTo100Action** (QAction \*const action)
- void **setZoomToFitAction** (QAction \*const action)
- void **triggerZoomTrackerToolTip** ()

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Static Public Member Functions

- static int **sizeFromZoom** (double zoom, double zmin, double zmax)
- static double **zoomFromSize** (int size, double zmin, double zmax)

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.540.1 Member Enumeration Documentation

### 6.540.1.1 BarMode

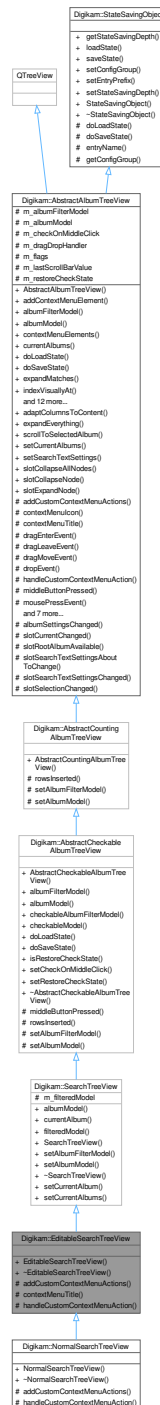
enum [Digikam::DZoomBar::BarMode](#)

## Enumerator

PreviewZoomCtrl	Preview Zoom controller.
ThumbsSizeCtrl	Thumb Size controller. Preview zoom controller still visible but disabled.
NoPreviewZoomCtrl	Thumb Size controller alone. Preview Zoom controller is hidden.

## 6.541 Digikam::EditableSearchTreeView Class Reference

Inheritance diagram for Digikam::EditableSearchTreeView:



### Public Member Functions

- [EditableSearchTreeView](#) (QWidget \*const parent, [SearchModel](#) \*const searchModel, [SearchModificationHelper](#) \*const searchModificationHelper)
- [~EditableSearchTreeView](#) () override

## Public Member Functions inherited from [Digikam::SearchTreeView](#)

- [searchModel](#) \* [albumModel](#) () const  
*Note: not filtered by search type.*
- [SAlbum](#) \* [currentAlbum](#) () const
- [SearchFilterModel](#) \* [filteredModel](#) () const  
*Contains only the searches with appropriate type - prefer to [albumModel\(\)](#)*
- [SearchTreeView](#) (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)
- void [setAlbumFilterModel](#) ([SearchFilterModel](#) \*const [filteredModel](#), [CheckableAlbumFilterModel](#) \*const model)
- void [setAlbumModel](#) ([searchModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- [CheckableAlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [albumModel](#) () const
- [CheckableAlbumFilterModel](#) \* [checkableAlbumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [checkableModel](#) () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void [addContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractSpecificAlbumModel](#) \* [albumModel](#) () const
- QList< [ContextMenuElement](#) \* > [contextMenuElements](#) () const
- template<class A >  
QList< A \* > [currentAlbums](#) ()
- bool [expandMatches](#) (const QModelIndex &index)
- QModelIndex [indexVisuallyAt](#) (const QPoint &p)
- void [removeContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > [selectedItems](#) ()  
*[selectedItems\(\)](#) -*
- void [setAlbumManagerCurrentAlbum](#) (const bool setCurrentAlbum)
- void [setContextMenuIcon](#) (const QPixmap &pixmap)
- void [setContextMenuTitle](#) (const QString &title)
- void [setEnabledContextMenu](#) (const bool enable)
- void [setExpandNewCurrentItem](#) (const bool doThat)
- void [setExpandOnSingleClick](#) (const bool doThat)
- void [setSelectAlbumOnClick](#) (const bool selectOnClick)
- void [setSelectOnContextMenu](#) (const bool select)
- bool [viewportEvent](#) (QEvent \*event) override

## Public Member Functions inherited from Digikam::StateSavingObject

- [StateSavingDepth](#) `getStateSavingDepth ()` const
- void `loadState ()`
- void `saveState ()`
- virtual void `setConfigGroup (const KConfigGroup &group)`
- virtual void `setEntryPrefix (const QString &prefix)`
- void `setStateSavingDepth (const StateSavingDepth depth)`
- [StateSavingObject](#) (`QObject *const host`)
- virtual `~StateSavingObject ()`

## Protected Member Functions

- void `addCustomContextMenuActions (ContextMenuHelper &cmh, Album *album)` override
- QString `contextMenuTitle ()` const override
- void `handleCustomContextMenuAction (QAction *action, const AlbumPointer< Album > &album)` override

## Protected Member Functions inherited from Digikam::AbstractCheckableAlbumTreeView

- void `middleButtonPressed (Album *a)` override
- void `rowsInserted (const QModelIndex &parent, int start, int end)` override
- void `setAlbumFilterModel (CheckableAlbumFilterModel *const filterModel)`
- void `setAlbumModel (AbstractCheckableAlbumModel *const model)`

## Protected Member Functions inherited from Digikam::AbstractCountingAlbumTreeView

- void `rowsInserted (const QModelIndex &parent, int start, int end)` override
- void `setAlbumFilterModel (AlbumFilterModel *const filterModel)`
- void `setAlbumModel (AbstractCountingAlbumModel *const model)`

## Protected Member Functions inherited from Digikam::AbstractAlbumTreeView

- virtual QPixmap `contextMenuIcon ()` const
  - void `dragEnterEvent (QDragEnterEvent *e)` override
  - void `dragLeaveEvent (QDragLeaveEvent *e)` override
  - void `dragMoveEvent (QDragMoveEvent *e)` override
  - void `dropEvent (QDropEvent *e)` override
  - void `mousePressEvent (QMouseEvent *e)` override
- Other helper methods.*
- virtual QPixmap  `pixmapForDrag (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)`
  - void `rowsAboutToBeRemoved (const QModelIndex &parent, int start, int end)` override
  - void `rowsInserted (const QModelIndex &index, int start, int end)` override
  - void `setAlbumFilterModel (AlbumFilterModel *const filterModel)`
  - void `setAlbumModel (AbstractSpecificAlbumModel *const model)`
  - virtual bool `showContextMenuAt (QContextMenuEvent *event, Album *albumForEvent)`
  - void `startDrag (Qt::DropActions supportedActions)` override

## Protected Member Functions inherited from Digikam::StateSavingObject

- QString `entryName (const QString &base)` const
- KConfigGroup `getConfigGroup ()` const

## Additional Inherited Members

### Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) { [CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) , [AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Public Slots inherited from [Digikam::SearchTreeView](#)

- void **setCurrentAlbum** (int searchId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)

### Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)
- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)
- void **slotCollapseAllNodes** ()  
*slotCollapseAllNodes - collapse all nodes without root node*
- void **slotCollapseNode** ()  
*slotCollapseNode - collapse recursively selected nodes*
- void **slotExpandNode** ()  
*slotExpandNode - expands recursively selected nodes*

### Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< [Album](#) \* > &selectedAlbums)

### Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void **albumSettingsChanged** ()
- void **slotCurrentChanged** ()
- virtual void [slotRootAlbumAvailable](#) ()
- void **slotSearchTextSettingsAboutToChange** (bool searched, bool willSearch)
- void **slotSearchTextSettingsChanged** (bool wasSearching, bool searching)
- void **slotSelectionChanged** ()

### Protected Attributes inherited from [Digikam::SearchTreeView](#)

- [SearchFilterModel](#) \* **m\_filteredModel** = nullptr

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* `m_albumFilterModel` = nullptr
- [AbstractSpecificAlbumModel](#) \* `m_albumModel` = nullptr
- bool `m_checkOnMiddleClick` = false
- [AlbumModelDragDropHandler](#) \* `m_dragDropHandler` = nullptr
- Flags `m_flags` = DefaultFlags
- int `m_lastScrollBarValue` = 0
- bool `m_restoreCheckState` = false

### 6.541.1 Detailed Description

This tree view for searches adds basic editing functionality via the context menu. This is in detail deleting and renaming existing searches.

Author

jwienke

### 6.541.2 Constructor & Destructor Documentation

#### 6.541.2.1 EditableSearchTreeView()

```
Digikam::EditableSearchTreeView::EditableSearchTreeView (
    QWidget *const parent,
    SearchModel *const searchModel,
    SearchModificationHelper *const searchModificationHelper )
```

Constructor.

Parameters

<i>parent</i>	qt parent
<i>searchModel</i>	the model this view should act on
<i>searchModificationHelper</i>	the modification helper object used to perform operations on the displayed searches

#### 6.541.2.2 ~EditableSearchTreeView()

```
Digikam::EditableSearchTreeView::~EditableSearchTreeView ( ) [override]
```

Destructor.

### 6.541.3 Member Function Documentation

#### 6.541.3.1 addCustomContextMenuActions()

```
void Digikam::EditableSearchTreeView::addCustomContextMenuActions (
    ContextMenuHelper & cmh,
    Album * album ) [override], [protected], [virtual]
```

Adds actions to delete or rename existing searches.

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

Reimplemented in [Digikam::NormalSearchTreeView](#).

### 6.541.3.2 contextMenuTitle()

```
QString Digikam::EditableSearchTreeView::contextMenuTitle ( ) const [override], [protected],  
[virtual]
```

implemented hook methods for context menus.

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

### 6.541.3.3 handleCustomContextMenuAction()

```
void Digikam::EditableSearchTreeView::handleCustomContextMenuAction (   
    QAction * action,  
    const AlbumPointer< Album > & album ) [override], [protected], [virtual]
```

Handles deletion and renaming actions.

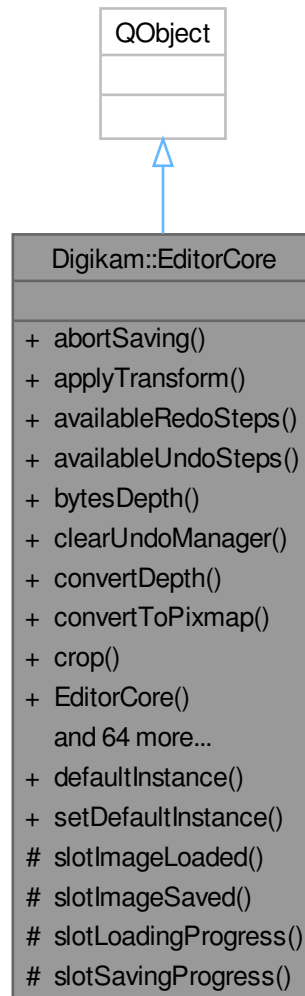
Reimplemented from [Digikam::AbstractAlbumTreeView](#).

Reimplemented in [Digikam::NormalSearchTreeView](#).



## 6.542 Digikam::EditorCore Class Reference

Inheritance diagram for Digikam::EditorCore:



### Classes

- class [Private](#)

### Signals

- void **signalFileOriginChanged** (const QString &filePath)
- void **signalImageLoaded** (const QString &filePath, bool success)
- void **signalImageSaved** (const QString &filePath, bool success)
- void **signalLoadingProgress** (const QString &filePath, float progress)
- void **signalLoadingStarted** (const QString &filename)

- void **signalModified** ()
- void **signalSavingProgress** (const QString &filePath, float progress)
- void **signalSavingStarted** (const QString &filename)
- void **signalUndoStateChanged** ()

## Public Member Functions

- void **abortSaving** ()
- void **applyTransform** (const [IccTransform](#) &transform)
- int **availableRedoSteps** () const
- int **availableUndoSteps** () const
- int **bytesDepth** () const
- void **clearUndoManager** ()
- void **convertDepth** (int depth)
- QPixmap **convertToPixmap** (const [DImg](#) &img) const
- void **crop** (const QRect &rect)
- QString **ensureHasCurrentUuid** () const
- bool **exifRotated** () const
- void **flipHoriz** ()
- void **flipVert** ()
- [IccProfile](#) **getEmbeddedICC** () const
- [ExposureSettingsContainer](#) \* **getExposureSettings** () const
- [ICCSettingsContainer](#) **getICCSettings** () const
- QString **getImageFileName** () const
- QString **getImageFilePath** () const
- QString **getImageFormat** () const
- [DImageHistory](#) **getImageHistoryOfFullRedo** () const
- [DImg](#) \* **getImg** () const
- [DImg](#) **getImgSelection** () const
- [DImageHistory](#) **getInitialImageHistory** () const
- [DImageHistory](#) **getItemHistory** () const
- [MetaEngineData](#) **getMetadata** () const
- QStringList **getRedoHistory** () const
- [DImageHistory](#) **getResolvedInitialHistory** () const
- QRect **getSelectedArea** () const
- QStringList **getUndoHistory** () const
- bool **hasAlpha** () const
- int **height** () const
- void **imageUndoChanged** (const [UndoMetadataContainer](#) &c)
- bool **isReadOnly** () const
- bool **isValid** () const
- void **load** (const QString &filename, [IOFileSettings](#) \*const iofileSettings)
- QSize **loadedSize** () const
- int **origHeight** () const
- int **origWidth** () const
- void **provideCurrentUuid** (const QString &uuid)
- void **putIccProfile** (const [IccProfile](#) &profile)
- void **putImg** (const QString &caller, const [FilterAction](#) &action, const [DImg](#) &img)
- void **putImgSelection** (const QString &caller, const [FilterAction](#) &action, const [DImg](#) &img)
- void **readMetadataFromFile** (const QString &file)
- void **redo** ()
- void **resetImage** ()
- void **restore** ()
- void **rollbackToOrigin** ()

- void **rotate180** ()
- void **rotate270** ()
- void **rotate90** ()
- void **saveAs** (const QString &file, [IOFileSettings](#) \*const iofileSettings, bool setExifOrientationTag, const QString &givenMimeType, const QString &intendedFilePath)
- void **saveAs** (const QString &file, [IOFileSettings](#) \*const iofileSettings, bool setExifOrientationTag, const QString &givenMimeType, const [VersionFileOperation](#) &operation)
- void **setDisplayingWidget** (QWidget \*const widget)
- void **setExifOrient** (bool exifOrient)
- void **setExposureSettings** ([ExposureSettingsContainer](#) \*const expoSettings)
- void **setFileOriginData** (const QVariant &data)
- void **setHistoryIsBranch** (bool isBranching)
- void **setICCSettings** (const [ICCSettingsContainer](#) &cmSettings)
- void **setLastSaved** (const QString &filePath)
- void **setModified** ()
- void **setResolvedInitialHistory** (const [DImageHistory](#) &history)
- void **setSelectedArea** (const QRect &rect)
- void **setSoftProofingEnabled** (bool enabled)
- void **setUndoImg** (const [UndoMetadataContainer](#) &c, const [DImg](#) &img)
- void **setUndoManagerOrigin** ()
- bool **sixteenBit** () const
- bool **softProofingEnabled** () const
- void **switchToLastSaved** (const [DImageHistory](#) &resolvedCurrentHistory=[DImageHistory](#)())
- void **undo** ()
- [UndoState](#) **undoState** () const
- int **width** () const
- void **zoom** (double val)

### Static Public Member Functions

- static [EditorCore](#) \* **defaultInstance** ()
- static void **setDefaultInstance** ([EditorCore](#) \*const instance)

### Protected Slots

- void **slotImageLoaded** (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)
- void **slotImageSaved** (const QString &filePath, bool success)
- void **slotLoadingProgress** (const [LoadingDescription](#) &loadingDescription, float progress)
- void **slotSavingProgress** (const QString &filePath, float progress)

## 6.542.1 Member Function Documentation

### 6.542.1.1 convertToPixmap()

```
QPixmap Digikam::EditorCore::convertToPixmap (
    const DImg & img ) const
```

Convert a [DImg](#) image to a pixmap for screen using color managed view if necessary.

### 6.542.1.2 getImgSelection()

```
DImg Digikam::EditorCore::getImgSelection ( ) const
```

Image properties

### 6.542.1.3 rotate90()

```
void Digikam::EditorCore::rotate90 ( )
```

Image transforms

### 6.542.1.4 setUndoImg()

```
void Digikam::EditorCore::setUndoImg (
    const UndoMetadataContainer & c,
    const DImg & img )
```

For internal usage by [UndoManager](#)

## 6.543 Digikam::EditorCore::Private Class Reference

### Classes

- class [FileToSave](#)

### Public Member Functions

- void **applyBuiltinFilter** (const [DImgBuiltinFilter](#) &filter, [UndoAction](#) \*const action)
- void **applyReversibleBuiltinFilter** (const [DImgBuiltinFilter](#) &filter)
- QMap< QString, QVariant > **ioAttributes** ([IOFileSettings](#) \*const iofileSettings, const QString &givenMimeType) const
- void **load** (const [LoadingDescription](#) &description)
- void **loadCurrent** ()
- void **putImageData** (uchar \*const data, int w, int h, bool sixteenBit)
- void **resetValues** ()
- void **saveAs** (const QString &file, [IOFileSettings](#) \*const iofileSettings, bool setExifOrientationTag, const QString &givenMimeType, const [VersionFileOperation](#) &operation, const QString &intendedFilePath)
- void **saveNext** ()

### Public Attributes

- [ICCSettingsContainer](#) **cmSettings**
- [LoadingDescription](#) **currentDescription**
- int **currentFileToSave** = 0
- QWidget \* **displayingWidget** = nullptr
- bool **doSoftProofing** = false
- bool **exifOrient** = false
- [ExposureSettingsContainer](#) \* **expoSettings** = nullptr
- QList< [FileToSave](#) > **filesToSave**
- int **height** = 0
- [DImg](#) **image**
- QSize **loadedSize**
- int **origHeight** = 0
- int **origWidth** = 0
- [DPluginRawImport](#) \* **rawPlugin** = nullptr
- [DImageHistory](#) **resolvedInitialHistory**
- bool **rotatedOrFlipped** = false
- int **selH** = 0
- int **selW** = 0
- int **selX** = 0
- int **selY** = 0
- [SharedLoadSaveThread](#) \* **thread** = nullptr
- [UndoManager](#) \* **undoMan** = nullptr
- bool **valid** = false
- int **width** = 0
- double **zoom** = 1.0

## 6.544 Digikam::EditorCore::Private::FileToSave Class Reference

### Public Attributes

- QString **fileName**
- QString **filePath**
- int **historyStep** = -1
- [DImg](#) **image**
- QString **intendedFilePath**
- QMap< QString, QVariant > **ioAttributes**
- QString  **mimeType**
- bool **setExifOrientationTag** = false

## 6.545 Digikam::EditorStackView Class Reference

Inheritance diagram for Digikam::EditorStackView:



### Public Types

- enum **StackViewMode** { **CanvasMode** = 0 , **ToolViewMode** }

### Public Slots

- void **setZoomFactor** (double)
- void **slotZoomSliderChanged** (int)

### Signals

- void **signalToggleOffFitToWindow** ()
- void **signalZoomChanged** (bool isMax, bool isMin, double zoom)

## Public Member Functions

- [Canvas](#) \* **canvas** () const
- void **decreaseZoom** ()
- **EditorStackView** (QWidget \*const parent=nullptr)
- void **fitToSelect** ()
- void **increaseZoom** ()
- bool **isZoomablePreview** () const
- void **setCanvas** ([Canvas](#) \*const canvas)
- void **setToolView** (QWidget \*const view)
- void **setViewMode** (int mode)
- void **toggleFitToWindow** ()
- QWidget \* **toolView** () const
- int **viewMode** () const
- double **zoomMax** () const
- double **zoomMin** () const
- void **zoomTo100Percent** ()

## 6.546 Digikam::EditorTool Class Reference

Inheritance diagram for Digikam::EditorTool:



### Public Slots

- virtual void **slotApplyTool** ()
- virtual void **slotCloseTool** ()
- void **slotPreviewModeChanged** ()
- void **slotUpdateSpotInfo** (const [Digikam::DColor](#) &col, const QPoint &point)



## Signals

- void **cancelClicked** ()
- void **okClicked** ()

## Public Member Functions

- **EditorTool** (QObject \*const parent)
- virtual void **exposureSettingsChanged** ()
- virtual void **ICCSettingsChanged** ()
- void **init** ()
- **DPlugin** \* **plugin** () const
- virtual void **setBackgroundcolor** (const QColor &bg)
- void **setInitPreview** (bool b)
- void **setPlugin** (DPlugin \*const plugin)
- **FilterAction::Category** **toolCategory** () const
- QString **toolHelp** () const
- QIcon **toolIcon** () const
- QString **toolName** () const
- **EditorToolSettings** \* **toolSettings** () const
- int **toolVersion** () const
- QWidget \* **toolView** () const

## Protected Slots

- virtual void **slotCancel** ()
- virtual void **slotInit** ()
- virtual void **slotLoadSettings** ()
- virtual void **slotOk** ()
- virtual void **slotResetSettings** ()
- void **slotTimer** ()

## Protected Member Functions

- virtual void **finalRendering** ()
- virtual void **readSettings** ()
- virtual void **setBusy** (bool)
- void **setPreviewModeMask** (int mask)
- void **setToolCategory** (const **FilterAction::Category** category)
- void **setToolHelp** (const QString &anchor)
- void **setToolIcon** (const QIcon &icon)
- void **setToolInfoMessage** (const QString &txt)
- void **setToolName** (const QString &name)
- virtual void **setToolSettings** (**EditorToolSettings** \*const settings)
- void **setToolVersion** (const int version)
- virtual void **setToolView** (QWidget \*const view)
- virtual void **slotChannelChanged** ()
- virtual void **slotPreview** ()
- virtual void **slotSaveAsSettings** ()
- virtual void **slotScaleChanged** ()
- virtual void **writeSettings** ()

## 6.546.1 Member Function Documentation

### 6.546.1.1 `init()`

```
void Digikam::EditorTool::init ( )
```

Called by editor tool interface to initialize tool when all is ready, through `slotInit()`.

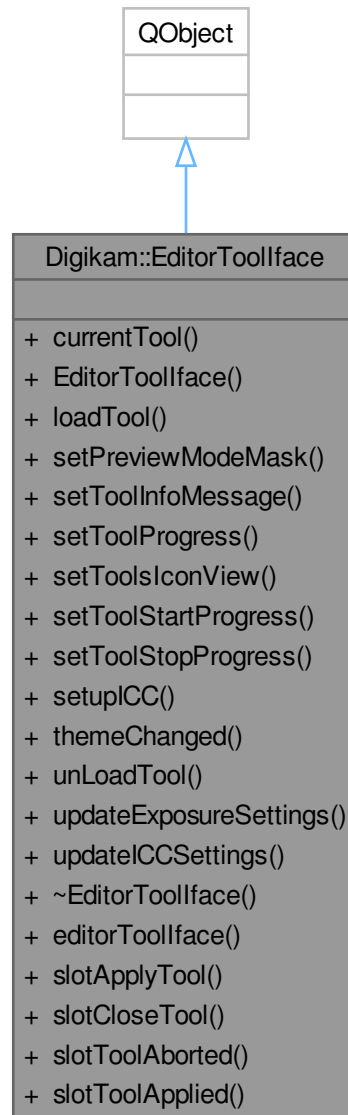
### 6.546.1.2 `setInitPreview()`

```
void Digikam::EditorTool::setInitPreview (
    bool b )
```

Set this option to on if you want to call `slotPreview()` in `slotInit()` at tool startup.

## 6.547 Digikam::EditorToolface Class Reference

Inheritance diagram for Digikam::EditorToolface:



### Public Slots

- void **slotApplyTool** ()
- void **slotCloseTool** ()
- void **slotToolAborted** ()
- void **slotToolApplied** ()

### Signals

- void **signalPreviewModeChanged** ()

**Public Member Functions**

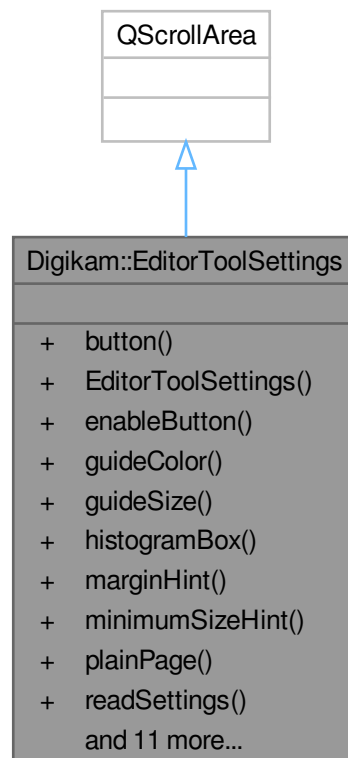
- [EditorTool](#) \* **currentTool** () const
- **EditorToolface** ([EditorWindow](#) \*const editor)
- void **loadTool** ([EditorTool](#) \*const tool)
- void **setPreviewModeMask** (int mask)
- void **setToolInfoMessage** (const QString &txt)
- void **setToolProgress** (int progress)
- void **setToolsIconView** ([DCategorizedView](#) \*const view)
- void **setToolStartProgress** (const QString &toolName)
- void **setToolStopProgress** ()
- void **setupICC** ()
- void **themeChanged** ()
- void **unLoadTool** ()
- void **updateExposureSettings** ()
- void **updateICCSettings** ()

**Static Public Member Functions**

- static [EditorToolface](#) \* **editorToolface** ()

**6.548 Digikam::EditorToolSettings Class Reference**

Inheritance diagram for Digikam::EditorToolSettings:



## Public Types

- enum **ButtonCode** {  
    **Default** = 0x00000001 , **Try** = 0x00000002 , **Ok** = 0x00000004 , **Cancel** = 0x00000008 ,  
    **SaveAs** = 0x00000010 , **Load** = 0x00000020 }
- enum **ToolCode** { **NoTool** = 0x00000000 , **ColorGuide** = 0x00000001 , **Histogram** = 0x00000002 }

## Signals

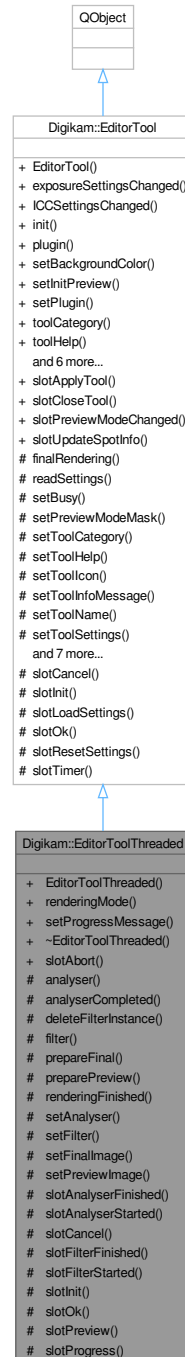
- void **signalCancelClicked** ()
- void **signalChannelChanged** ()
- void **signalColorGuideChanged** ()
- void **signalDefaultClicked** ()
- void **signalLoadClicked** ()
- void **signalOkClicked** ()
- void **signalSaveAsClicked** ()
- void **signalScaleChanged** ()
- void **signalTryClicked** ()

## Public Member Functions

- QPushButton \* **button** (int buttonCode) const
- **EditorToolSettings** (QWidget \*const parent)
- void **enableButton** (int buttonCode, bool state)
- QColor **guideColor** () const
- int **guideSize** () const
- [HistogramBox](#) \* **histogramBox** () const
- int **marginHint** ()
- QSize **minimumSizeHint** () const override
- QWidget \* **plainPage** () const
- virtual void **readSettings** ()
- virtual void **resetSettings** ()
- virtual void **setBusy** (bool)
- void **setButtons** (Buttons buttonMask)
- void **setGuideColor** (const QColor &color)
- void **setGuideSize** (int size)
- void **setHistogramType** (HistogramBoxType type)
- void **setTool** ([EditorTool](#) \*const tool)
- void **setTools** (Tools toolMask)
- int **spacingHint** ()
- virtual void **writeSettings** ()

## 6.549 Digikam::EditorToolThreaded Class Reference

Inheritance diagram for Digikam::EditorToolThreaded:



### Public Types

- enum **RenderingMode** { **NoneRendering** = 0 , **PreviewRendering** , **FinalRendering** }

## Public Slots

- virtual void **slotAbort** ()

## Public Slots inherited from [Digikam::EditorTool](#)

- virtual void **slotApplyTool** ()
- virtual void **slotCloseTool** ()
- void **slotPreviewModeChanged** ()
- void **slotUpdateSpotInfo** (const [Digikam::DColor](#) &col, const QPoint &point)

## Public Member Functions

- **EditorToolThreaded** (QObject \*const parent)
- RenderingMode **renderingMode** () const
- void **setProgressMessage** (const QString &mess)

## Public Member Functions inherited from [Digikam::EditorTool](#)

- **EditorTool** (QObject \*const parent)
- virtual void **exposureSettingsChanged** ()
- virtual void **ICCSettingsChanged** ()
- void **init** ()
- [DPlugin](#) \* **plugin** () const
- virtual void **setBackgroundColor** (const QColor &bg)
- void **setInitPreview** (bool b)
- void **setPlugin** ([DPlugin](#) \*const plugin)
- [FilterAction::Category](#) **toolCategory** () const
- QString **toolHelp** () const
- QIcon **toolIcon** () const
- QString **toolName** () const
- [EditorToolSettings](#) \* **toolSettings** () const
- int **toolVersion** () const
- QWidget \* **toolView** () const

## Protected Slots

- void **slotAnalyserFinished** (bool success)
- void **slotAnalyserStarted** ()
- void **slotCancel** () override
- void **slotFilterFinished** (bool success)
- void **slotFilterStarted** ()
- void **slotInit** () override
- void **slotOk** () override
- void **slotPreview** () override
- void **slotProgress** (int progress)

## Protected Slots inherited from [Digikam::EditorTool](#)

- virtual void **slotCancel** ()
- virtual void **slotInit** ()
- virtual void **slotLoadSettings** ()
- virtual void **slotOk** ()
- virtual void **slotResetSettings** ()
- void **slotTimer** ()

## Protected Member Functions

- [DImgThreadedAnalyser](#) \* **analyser** () const
- virtual void **analyserCompleted** ()
- void **deleteFilterInstance** (bool b=true)
- [DImgThreadedFilter](#) \* **filter** () const
- virtual void **prepareFinal** ()
- virtual void **preparePreview** ()
- virtual void **renderingFinished** ()
- void **setAnalyser** ([DImgThreadedAnalyser](#) \*const **analyser**)
- void **setFilter** ([DImgThreadedFilter](#) \*const **filter**)
- virtual void **setFinalImage** ()
- virtual void **setPreviewImage** ()

## Protected Member Functions inherited from [Digikam::EditorTool](#)

- virtual void **finalRendering** ()
- virtual void **readSettings** ()
- virtual void **setBusy** (bool)
- void **setPreviewModeMask** (int mask)
- void **setToolCategory** (const [FilterAction::Category](#) category)
- void **setToolHelp** (const QString &anchor)
- void **setToolIcon** (const QIcon &icon)
- void **setToolInfoMessage** (const QString &txt)
- void **setToolName** (const QString &name)
- virtual void **setToolSettings** ([EditorToolSettings](#) \*const settings)
- void **setToolVersion** (const int version)
- virtual void **setToolView** (QWidget \*const view)
- virtual void **slotChannelChanged** ()
- virtual void **slotPreview** ()
- virtual void **slotSaveAsSettings** ()
- virtual void **slotScaleChanged** ()
- virtual void **writeSettings** ()

## Additional Inherited Members

## Signals inherited from [Digikam::EditorTool](#)

- void **cancelClicked** ()
- void **okClicked** ()



## 6.549.1 Member Function Documentation

### 6.549.1.1 analyser()

```
DImgThreadedAnalyser * Digikam::EditorToolThreaded::analyser ( ) const [protected]
```

Manage analyser instance plugged in tool interface

### 6.549.1.2 deleteFilterInstance()

```
void Digikam::EditorToolThreaded::deleteFilterInstance (
    bool b = true ) [protected]
```

If true, delete filter instance when preview or final rendering is processed. If false, filter instance will be managed outside for ex. with ContentAwareResizing tool.

### 6.549.1.3 filter()

```
DImgThreadedFilter * Digikam::EditorToolThreaded::filter ( ) const [protected]
```

Manage filter instance plugged in tool interface

### 6.549.1.4 renderingMode()

```
EditorToolThreaded::RenderingMode Digikam::EditorToolThreaded::renderingMode ( ) const
```

return the current tool rendering mode.

### 6.549.1.5 setProgressMessage()

```
void Digikam::EditorToolThreaded::setProgressMessage (
    const QString & mess )
```

Set the small text to show in editor status progress bar during tool computation. If it's not set, tool name is used instead.

### 6.549.1.6 slotAnalyserStarted

```
void Digikam::EditorToolThreaded::slotAnalyserStarted ( ) [protected], [slot]
```

Manage start and end events from analyser

### 6.549.1.7 slotFilterStarted

```
void Digikam::EditorToolThreaded::slotFilterStarted ( ) [protected], [slot]
```

Manage start and end events from filter

### 6.549.1.8 slotProgress

```
void Digikam::EditorToolThreaded::slotProgress (
    int progress ) [protected], [slot]
```

Dispatch progress event from filter and analyser

## 6.550 Digikam::EditorWindow Class Reference

Inheritance diagram for Digikam::EditorWindow:



## Classes

- class [Private](#)

## Public Types

- enum **TransformType** { **RotateLeft** , **RotateRight** , **FlipHorizontal** , **FlipVertical** }

## Public Slots

- void **slotSetup** () override=0
- virtual void **slotSetupICC** ()=0

## Signals

- void **signalNoCurrentItem** ()
- void **signalPreviewModeChanged** (int)
- void **signalSelectionChanged** (const QRect &)
- void **signalToolApplied** ()

## Public Member Functions

- bool **actionEnabledState** () const
- **EditorWindow** (const QString &name, QWidget \*const parent=nullptr)
- void **loadTool** ([EditorTool](#) \*const tool)
- void [registerExtraPluginsActions](#) (QString &dom) override

## Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- QList< QAction \* > [allActions](#) () const
- void [cleanupActions](#) ()
- QString **configGroupName** () const
- void [createFullScreenAction](#) (const QString &name)
- void [createHelpActions](#) (const QString &handbookSection, bool coreOptions=true)
- void [createSettingsActions](#) ()
- void [createSidebarActions](#) ()
- **DXmlGuiWindow** (QWidget \*const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())
- bool [fullScreensIsActive](#) () const
- virtual [DInfoInterface](#) \* [infoface](#) ([DPluginAction](#) \*const ac)=0
- void [readFullScreenSettings](#) (const KConfigGroup &group)
- void [registerPluginsActions](#) ()
- void [setConfigGroupName](#) (const QString &name)
- void [setFullScreenOptions](#) (int options)
- void **unminimizeAndActivateWindow** ()

## Static Public Attributes

- static const QString **CONFIG\_GROUP\_NAME**

## Protected Types

- enum **SaveAskMode** {  
**AskIfNeeded** , **OverwriteWithoutAsking** , **AlwaysSaveAs** , **SaveVersionWithoutAsking** = Overwrite↔  
WithoutAsking ,  
**AlwaysNewVersion** = AlwaysSaveAs }

## Protected Slots

- virtual bool **save** ()=0
- virtual bool **saveAs** ()=0
- virtual bool **saveCurrentVersion** ()=0
- virtual bool **saveNewVersion** ()=0
- virtual bool **saveNewVersionAs** ()=0
- virtual bool **saveNewVersionInFormat** (const QString &)=0
- virtual bool **saveOrSaveAs** ()
- void **slotAboutToShowRedoMenu** ()
- void **slotAboutToShowUndoMenu** ()
- virtual void **slotAddedDroppedItems** (QDropEvent \*e)=0
- virtual void **slotBackward** ()=0
- virtual void **slotChanged** ()=0
- void **slotComponentsInfo** () override
- virtual void **slotContextMenu** ()=0
- virtual void **slotDeleteCurrentItem** ()=0
- virtual void **slotDiscardChanges** ()
- virtual void **slotFileOriginChanged** (const QString &filePath)
- virtual void **slotFileWithDefaultApplication** ()=0
- virtual void **slotFirst** ()=0
- virtual void **slotForward** ()=0
- virtual void **slotLast** ()=0
- virtual void **slotLoadingFinished** (const QString &filename, bool success)
- void **slotLoadingProgress** (const QString &filePath, float progress)
- virtual void **slotLoadingStarted** (const QString &filename)
- void **slotNameLabelCancelButtonPressed** ()
- virtual void **slotOpenOriginal** ()
- virtual void **slotOpenWith** (QAction \*action=nullptr)=0
- virtual void **slotPrepareToLoad** ()
- virtual void **slotRevert** ()=0
- void **slotSavingProgress** (const QString &filePath, float progress)
- virtual void **slotSavingStarted** (const QString &filename)
- void **slotSelected** (bool)
- virtual void **slotUpdateItemInfo** ()=0

## Protected Slots inherited from [Digikam::DXmlGuiWindow](#)

- bool **slotClose** ()

## Protected Member Functions

- virtual void **addServicesMenu** ()=0
- void **addServicesMenuForUrl** (const QUrl &url)
- void **applyColorManagementSettings** ()
- void **applyIOSettings** ()
- void **applyStandardSettings** ()
- bool **checkOverwrite** (const QUrl &url)
- bool **checkPermissions** (const QUrl &url)
- void **colorManage** ()
- [EditorStackView](#) \* **editorStackView** () const
- void **execSavingProgressDialog** ()
- [ExposureSettingsContainer](#) \* **exposureSettings** () const
- virtual void **finishSaving** (bool success)
- virtual bool **hasOriginalToRestore** ()
- virtual void **moveFile** ()
- bool **moveLocalFile** (const QString &src, const QString &dest)
- void **movingSaveFileFinished** (bool successful)
- void **openWith** (const QUrl &url, QAction \*action)
- bool **promptForOverWrite** ()
- bool **promptUserDelete** (const QUrl &url)
- bool **promptUserSave** (const QUrl &url, SaveAskMode mode=AskIfNeeded, bool allowCancel=true)
- virtual void **readSettings** ()
- void **readStandardSettings** ()
- void **resetOrigin** ()
- void **resetOriginSwitchFile** ()
- virtual [DImageHistory](#) **resolvedImageHistory** (const [DImageHistory](#) &history)
- virtual [Sidebar](#) \* **rightSideBar** () const =0
- virtual void **saveAsIsComplete** ()=0
- [VersionFileOperation](#) **saveAsVersionFileOperation** (const QUrl &url, const QUrl &saveLocation, const QString &format)
- virtual QUrl **saveDestinationUrl** ()=0
- [VersionFileOperation](#) **saveInFormatVersionFileOperation** (const QUrl &url, const QString &format)
- virtual void **savelsComplete** ()=0
- virtual void **saveSettings** ()
- void **saveStandardSettings** ()
- [VersionFileOperation](#) **saveVersionFileOperation** (const QUrl &url, bool fork)
- virtual void **saveVersionIsComplete** ()=0
- virtual void **setupActions** ()=0
- virtual void **setupConnections** ()=0
- void **setupContextMenu** ()
- void **setupSelectToolsAction** ()
- void **setupStandardActions** ()
- void **setupStandardConnections** ()
- void **setupStatusBar** ()
- virtual void **setupUserArea** ()=0
- [SidebarSplitter](#) \* **sidebarSplitter** () const
- void **startingSave** (const QUrl &url)
- bool **startingSaveAs** (const QUrl &url)
- bool **startingSaveCurrentVersion** (const QUrl &url)
- bool **startingSaveNewVersion** (const QUrl &url)
- bool **startingSaveNewVersionAs** (const QUrl &url)
- bool **startingSaveNewVersionInFormat** (const QUrl &url, const QString &format)
- virtual [ThumbBarDock](#) \* **thumbBar** () const =0
- virtual void **toggleActions** (bool val)

- void **toggleNonDestructiveActions** ()
- void **toggleStandardActions** (bool val)
- void **toggleToolActions** ([EditorTool](#) \*tool=nullptr)
- void **toggleZoomActions** (bool val)
- virtual [VersionManager](#) \* **versionManager** () const
- bool **waitForSavingToComplete** ()

### Protected Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- void **closeEvent** (QCloseEvent \*e) override
- void **editKeyboardShortcuts** (KActionCollection \*const extraac=nullptr, const QString &actitle=QString())
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override
- void **keyPressEvent** (QKeyEvent \*e) override
- QAction \* **showMenuBarAction** () const
- QAction \* **showStatusBarAction** () const

### Protected Attributes

- bool **m\_actionEnabledState** = false
- QAction \* **m\_applyToolAction** = nullptr
- QAction \* **m\_backwardAction** = nullptr
- QColor **m\_bgColor**
- [Canvas](#) \* **m\_canvas** = nullptr
- QAction \* **m\_closeToolAction** = nullptr
- QMenu \* **m\_contextMenu** = nullptr
- QAction \* **m\_discardChangesAction** = nullptr
- bool **m\_editingOriginalImage** = true
- QAction \* **m\_exportAction** = nullptr
- QAction \* **m\_fileDeleteAction** = nullptr
- QAction \* **m\_firstAction** = nullptr
- QString **m\_formatForRAWVersioning**
- QString **m\_formatForSubversions**
- QAction \* **m\_forwardAction** = nullptr
- [IOFileSettings](#) \* **m\_IOFileSettings** = nullptr
- QAction \* **m\_lastAction** = nullptr
- [StatusProgressBar](#) \* **m\_nameLabel** = nullptr
- bool **m\_nonDestructive** = true
- QAction \* **m\_openVersionAction** = nullptr
- KToolBarPopupAction \* **m\_redoAction** = nullptr
- [DAdjustableLabel](#) \* **m\_resLabel** = nullptr
- QAction \* **m\_revertAction** = nullptr
- QAction \* **m\_saveAction** = nullptr
- QAction \* **m\_saveAsAction** = nullptr
- QAction \* **m\_saveCurrentVersionAction** = nullptr
- KToolBarPopupAction \* **m\_saveNewVersionAction** = nullptr
- QAction \* **m\_saveNewVersionAsAction** = nullptr
- QMenu \* **m\_saveNewVersionInFormatAction** = nullptr
- [SavingContext](#) **m\_savingContext**
- QPointer< QProgressDialog > **m\_savingProgressDialog** = nullptr
- QAction \* **m\_serviceAction** = nullptr
- QMenu \* **m\_servicesMenu** = nullptr
- bool **m\_setExifOrientationTag** = true
- QAction \* **m\_showBarAction** = nullptr
- [SidebarSplitter](#) \* **m\_splitter** = nullptr
- [EditorStackView](#) \* **m\_stackView** = nullptr
- QVector< TransformType > **m\_transformQue**  
*NOTE: using QVector to store transforms.*
- KToolBarPopupAction \* **m\_undoAction** = nullptr

## Protected Attributes inherited from [Digikam::DXmlGuiWindow](#)

- [DLogoAction](#) \* `m_animLogo` = nullptr

## Friends

- class `EditorTooliface`

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- static QAction \* `buildStdAction` (StdActionType type, const QObject \*const recvr, const char \*const slot, QObject \*const parent)
- static QString `configFullScreenHideSideBarsEntry` ()
- static QString `configFullScreenHideStatusBarEntry` ()
- static QString `configFullScreenHideThumbBarEntry` ()
- static QString `configFullScreenHideToolBarsEntry` ()
- static void `restoreWindowSize` (QWindow \*const win, const KConfigGroup &group)
- static void `saveWindowSize` (QWindow \*const win, KConfigGroup &group)
- static void `setGoodDefaultWindowSize` (QWindow \*const win)
- static void `setupIconTheme` ()

## 6.550.1 Member Function Documentation

### 6.550.1.1 registerExtraPluginsActions()

```
void Digikam::EditorWindow::registerExtraPluginsActions (
    QString & dom ) [override], [virtual]
```

Reimplemented from [Digikam::DXmlGuiWindow](#).

### 6.550.1.2 saveDestinationUrl()

```
virtual QUrl Digikam::EditorWindow::saveDestinationUrl ( ) [protected], [pure virtual]
```

Hook method that subclasses must implement to return the destination url of the image to save. This may also be a remote url.

This method will only be called while saving.

#### Returns

destination for the file that is currently being saved.

### 6.550.1.3 toggleZoomActions()

```
void Digikam::EditorWindow::toggleZoomActions (
    bool val ) [protected]
```

Method used by Editor Tools. Only tools based on imageregionwidget support zooming. TODO: Fix this behavior when editor tool preview widgets will be factored.

## 6.551 Digikam::EditorWindow::Private Class Reference

### Public Member Functions

- void **plugNewVersionInFormatAction** ([EditorWindow](#) \*const q, [QMenu](#) \*const menuAction, const [QString](#) &text, const [QString](#) &format)

### Public Attributes

- [QToolButton](#) \* **cmViewIndicator** = nullptr
- const [QString](#) **configAutoZoomEntry** = [QLatin1String](#)("AutoZoom")
- const [QString](#) **configAvifCompressionEntry** = [QLatin1String](#)("AVIFCompression")
- const [QString](#) **configAvifLossLessEntry** = [QLatin1String](#)("AVIFLossLess")
- const [QString](#) **configBackgroundColorEntry** = [QLatin1String](#)("BackgroundColor")
- const [QString](#) **configExpoIndicatorModeEntry** = [QLatin1String](#)("ExpoIndicatorMode")
- const [QString](#) **configHeifCompressionEntry** = [QLatin1String](#)("HEIFCompression")
- const [QString](#) **configHeifLossLessEntry** = [QLatin1String](#)("HEIFLossLess")
- const [QString](#) **configJpeg2000CompressionEntry** = [QLatin1String](#)("JPEG2000Compression")
- const [QString](#) **configJpeg2000LossLessEntry** = [QLatin1String](#)("JPEG2000LossLess")
- const [QString](#) **configJpegCompressionEntry** = [QLatin1String](#)("JPEGCompression")
- const [QString](#) **configJpegSubSamplingEntry** = [QLatin1String](#)("JPEGSubSampling")
- const [QString](#) **configJxlCompressionEntry** = [QLatin1String](#)("JXLCompression")
- const [QString](#) **configJxlLossLessEntry** = [QLatin1String](#)("JXLLossLess")
- const [QString](#) **configOverExposureColorEntry** = [QLatin1String](#)("OverExposureColor")
- const [QString](#) **configOverExposureIndicatorEntry** = [QLatin1String](#)("OverExposureIndicator")
- const [QString](#) **configOverExposurePercentsEntry** = [QLatin1String](#)("OverExposurePercentsEntry")
- const [QString](#) **configPgfCompressionEntry** = [QLatin1String](#)("PGFCompression")
- const [QString](#) **configPgfLossLessEntry** = [QLatin1String](#)("PGFLossLess")
- const [QString](#) **configPngCompressionEntry** = [QLatin1String](#)("PNGCompression")
- const [QString](#) **configRawImportToolIdEntry** = [QLatin1String](#)("RawImportToolId")
- const [QString](#) **configTiffCompressionEntry** = [QLatin1String](#)("TIFFCompression")
- const [QString](#) **configUnderExposureColorEntry** = [QLatin1String](#)("UnderExposureColor")
- const [QString](#) **configUnderExposureIndicatorEntry** = [QLatin1String](#)("UnderExposureIndicator")
- const [QString](#) **configUnderExposurePercentsEntry** = [QLatin1String](#)("UnderExposurePercentsEntry")
- const [QString](#) **configUseRawImportToolEntry** = [QLatin1String](#)("UseRawImportTool")
- const [QString](#) **configUseThemeBackgroundColorEntry** = [QLatin1String](#)("UseThemeBackgroundColor")
- const [QString](#) **configVerticalSplitterStateEntry** = [QLatin1String](#)("Vertical Splitter State")
- const [QString](#) **configWebpCompressionEntry** = [QLatin1String](#)("WEBPCompression")
- const [QString](#) **configWebpLossLessEntry** = [QLatin1String](#)("WEBPLossLess")
- [QAction](#) \* **copyAction** = nullptr
- [QAction](#) \* **cropAction** = nullptr
- [QDialog](#) \* **currentWindowModalDialog** = nullptr
- [VersionManager](#) **defaultVersionManager**
- [ExposureSettingsContainer](#) \* **exposureSettings** = nullptr



- QAction \* **flipHorizAction** = nullptr
- QAction \* **flipVertAction** = nullptr
- QList< int > **fullscreenSizeBackup**
- ICCSettingsContainer \* **ICCSettings** = nullptr
- DAdjustableLabel \* **infoLabel** = nullptr
- QMap< QString, DServiceInfo > **newServicesMap**
- QAction \* **openWithAction** = nullptr
- QToolButton \* **overExposureIndicator** = nullptr
- PreviewToolBar \* **previewToolBar** = nullptr
- QAction \* **rotateLeftAction** = nullptr
- QAction \* **rotateRightAction** = nullptr
- QAction \* **selectAllAction** = nullptr
- QAction \* **selectNoneAction** = nullptr
- ActionCategorizedView \* **selectToolsActionView** = nullptr
- QMap< QString, KService::Ptr > **servicesMap**
- QAction \* **softProofOptionsAction** = nullptr
- EditorToolface \* **toolface** = nullptr
- QToolButton \* **underExposureIndicator** = nullptr
- QAction \* **viewCMViewAction** = nullptr
- QAction \* **viewOverExpoAction** = nullptr
- QAction \* **viewSoftProofAction** = nullptr
- QAction \* **viewUnderExpoAction** = nullptr
- QEventLoop \* **waitingLoop** = nullptr
- DZoomBar \* **zoomBar** = nullptr
- QAction \* **zoomFitToSelectAction** = nullptr
- QAction \* **zoomFitToWindowAction** = nullptr
- QAction \* **zoomMinusAction** = nullptr
- QAction \* **zoomPlusAction** = nullptr
- QAction \* **zoomTo100percents** = nullptr

## 6.552 Digikam::EffectMngr Class Reference

### Classes

- class [Private](#)

### Public Types

- enum [EffectType](#) {  
[None](#) = 0 , [KenBurnsZoomIn](#) , [KenBurnsZoomOut](#) , [KenBurnsPanLR](#) ,  
[KenBurnsPanRL](#) , [KenBurnsPanTB](#) , [KenBurnsPanBT](#) , [Random](#) }

### Public Member Functions

- QImage **currentFrame** (int &tmout)
- void **setEffect** ([EffectType](#) eff)
- void **setFrames** (int ifrms)
- void **setImage** (const QImage &img)
- void **setOutputSize** (const QSize &size)

### Static Public Member Functions

- static QMap< [EffectType](#), QString > **effectNames** ()

## 6.552.1 Member Enumeration Documentation

### 6.552.1.1 EffectType

enum [Digikam::EffectMngr::EffectType](#)

See KEn Burns effect description: [https://en.wikipedia.org/wiki/Ken\\_Burns\\_effect](https://en.wikipedia.org/wiki/Ken_Burns_effect)

Enumerator

None	Static camera.
------	----------------

## 6.553 Digikam::EffectMngr::Private Class Reference

### Public Types

- typedef int(EffectMngr::Private::\* **EffectMethod**) (bool)

### Public Member Functions

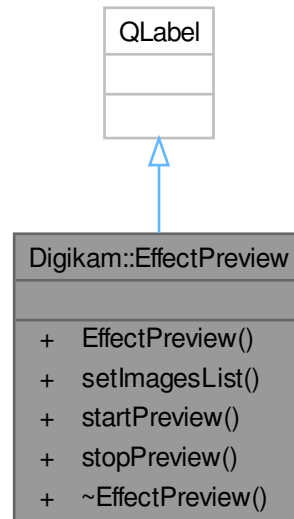
- [EffectMngr::EffectType](#) **getRandomEffect** () const
- void **registerEffects** ()

### Public Attributes

- [EffectMngr::EffectType](#) **eff\_curEffect** = [EffectMngr::None](#)
- QImage **eff\_curFrame**
- QMap< [EffectMngr::EffectType](#), EffectMethod > **eff\_effectList**
- QImage **eff\_image**
- int **eff\_imgFrames** = 125
- bool **eff\_isRunning** = false
- QSize **eff\_outSize**
- int **eff\_step** = 0

## 6.554 Digikam::EffectPreview Class Reference

Inheritance diagram for Digikam::EffectPreview:



### Public Member Functions

- **EffectPreview** (`QWidget *const parent=nullptr`)
- void **setImagesList** (`const QList< QUrl > &images`)
- void **startPreview** (`EffectMgr::EffectType eff`)
- void **stopPreview** ()

## 6.555 Digikam::Ellipsoid Class Reference

### Public Member Functions

- double **eccentricity** () const
- double **inverseFlattening** () const
- bool **isIsvfDefinitive** () const
- bool **isSphere** () const
- double **orthodromicDistance** (`double x1, double y1, double x2, double y2`)
- double **radiusOfCurvature** (`double latitude`)
- double **semiMajorAxis** () const
- double **semiMinorAxis** () const

### Static Public Member Functions

- static [Ellipsoid CLARKE\\_1866](#) ()
- static [Ellipsoid createEllipsoid](#) (const QString &name, double [semiMajorAxis](#), double [semiMinorAxis](#))
- static [Ellipsoid createFlattenedSphere](#) (const QString &name, double [semiMajorAxis](#), double [inverseFlattening](#))
- static [Ellipsoid GRS80](#) ()
- static [Ellipsoid INTERNATIONAL\\_1924](#) ()
- static [Ellipsoid SPHERE](#) ()
- static [Ellipsoid WGS84](#) ()

### Protected Member Functions

- **Ellipsoid** (const QString &name, double radius, bool ivfDefinitive)
- [Ellipsoid](#) (const QString &name, double [semiMajorAxis](#), double [semiMinorAxis](#), double [inverseFlattening](#), bool ivfDefinitive)

### Protected Attributes

- double [m\\_inverseFlattening](#) = 0.0
- bool [m\\_isSphere](#) = false
- bool [m\\_ivfDefinitive](#) = false
- double [m\\_semiMajorAxis](#) = 0.0
- double [m\\_semiMinorAxis](#) = 0.0
- QString [name](#)

## 6.555.1 Detailed Description

Geometric figure that can be used to describe the approximate shape of the earth. In mathematical terms, it is a surface formed by the rotation of an ellipse about its minor axis. An ellipsoid requires two defining parameters:

- semi-major axis and inverse flattening, or
- semi-major axis and semi-minor axis.

## 6.555.2 Constructor & Destructor Documentation

### 6.555.2.1 Ellipsoid()

```
Digikam::Ellipsoid::Ellipsoid (
    const QString & name,
    double semiMajorAxis,
    double semiMinorAxis,
    double inverseFlattening,
    bool ivfDefinitive ) [protected]
```

Constructs a new ellipsoid using the specified axis length. The properties map is given unchanged to the Abstract← IdentifiedObjectAbstractIdentifiedObject(Map) super-class constructor.

## Parameters

<i>name</i>	The ellipsoid name.
<i>semiMajorAxis</i>	The equatorial radius.
<i>semiMinorAxis</i>	The polar radius.
<i>inverseFlattening</i>	The inverse of the flattening value.
<i>ivfDefinitive</i>	<code>true</code> if the inverse flattening is definitive.

## See also

[createEllipsoid](#)

[createFlattenedSphere](#)

## 6.555.3 Member Function Documentation

### 6.555.3.1 CLARKE\_1866()

`Ellipsoid` Digikam::Ellipsoid::CLARKE\_1866 ( ) [static]

Clarke 1866 ellipsoid with axis in metres.

### 6.555.3.2 createEllipsoid()

`Ellipsoid` Digikam::Ellipsoid::createEllipsoid (   
     const QString & *name*,   
     double *semiMajorAxis*,   
     double *semiMinorAxis* ) [static]

Constructs a new ellipsoid using the specified axis length.

## Parameters

<i>name</i>	The ellipsoid name.
<i>semiMajorAxis</i>	The equatorial radius.
<i>semiMinorAxis</i>	The polar radius.

### 6.555.3.3 createFlattenedSphere()

`Ellipsoid` Digikam::Ellipsoid::createFlattenedSphere (   
     const QString & *name*,   
     double *semiMajorAxis*,   
     double *inverseFlattening* ) [static]

Constructs a new ellipsoid using the specified axis length and inverse flattening value.

## Parameters

<i>name</i>	The ellipsoid name.
-------------	---------------------

## Parameters

<i>semiMajorAxis</i>	The equatorial radius.
<i>inverseFlattening</i>	The inverse flattening value. values.

**6.555.3.4 eccentricity()**

```
double Digikam::Ellipsoid::eccentricity ( ) const
```

The ratio of the distance between the center and a focus of the ellipse to the length of its semimajor axis. The eccentricity can alternately be computed from the equation:  $e = \sqrt{2f - f^2}$ .

**6.555.3.5 GRS80()**

```
Ellipsoid Digikam::Ellipsoid::GRS80 ( ) [static]
```

GRS 80 ellipsoid with axis in metres.

**6.555.3.6 INTERNATIONAL\_1924()**

```
Ellipsoid Digikam::Ellipsoid::INTERNATIONAL_1924 ( ) [static]
```

International 1924 ellipsoid with axis in metres.

**6.555.3.7 inverseFlattening()**

```
double Digikam::Ellipsoid::inverseFlattening ( ) const
```

Returns the value of the inverse of the flattening constant. Flattening is a value used to indicate how closely an ellipsoid approaches a spherical shape. The inverse flattening is related to the equatorial/polar radius by the formula

$ivf = r_e / (r_e - r_p)$ .

For perfect spheres (i.e. if `isSphere` returns `true`), the `DoublePOSITIVE_INFINITY` value is used.

## Returns

The inverse flattening value.

**6.555.3.8 isIvfDefinitive()**

```
bool Digikam::Ellipsoid::isIvfDefinitive ( ) const
```

Indicates if the inverse flattening is definitive for this ellipsoid. Some ellipsoids use the IVF as the defining value, and calculate the polar radius whenever asked. Other ellipsoids use the polar radius to calculate the IVF whenever asked. This distinction can be important to avoid floating-point rounding errors.

## Returns

`true` if the inverse flattening is definitive, or `false` if the polar radius is definitive.

### 6.555.3.9 isSphere()

```
bool Digikam::Ellipsoid::isSphere ( ) const
```

`true` if the ellipsoid is degenerate and is actually a sphere. The sphere is completely defined by the semi-major axis, which is the radius of the sphere.

#### Returns

`true` if the ellipsoid is degenerate and is actually a sphere.

### 6.555.3.10 orthodromicDistance()

```
double Digikam::Ellipsoid::orthodromicDistance (
    double x1,
    double y1,
    double x2,
    double y2 )
```

Returns the orthodromic distance between two geographic coordinates. The orthodromic distance is the shortest distance between two points on a sphere's surface. The orthodromic path is always on a great circle. This is different from the loxodromic distance, which is a longer distance on a path with a constant direction on the compass.

#### Parameters

<code>x1</code>	Longitude of first point (in decimal degrees).
<code>y1</code>	Latitude of first point (in decimal degrees).
<code>x2</code>	Longitude of second point (in decimal degrees).
<code>y2</code>	Latitude of second point (in decimal degrees).

#### Returns

The orthodromic distance (in the units of this ellipsoid's axis).

### 6.555.3.11 radiusOfCurvature()

```
double Digikam::Ellipsoid::radiusOfCurvature (
    double latitude )
```

Returns the Radius Of Curvature for the given latitude, using the geometric mean of two radii of curvature for all azimuths.

#### Parameters

<code>latitude</code>	in degrees
-----------------------	------------

### 6.555.3.12 semiMajorAxis()

```
double Digikam::Ellipsoid::semiMajorAxis ( ) const
```

Length of the semi-major axis of the ellipsoid. This is the equatorial radius in axis linear unit.

#### Returns

Length of semi-major axis.

### 6.555.3.13 semiMinorAxis()

```
double Digikam::Ellipsoid::semiMinorAxis ( ) const
```

Length of the semi-minor axis of the ellipsoid. This is the polar radius in axis linear unit.

#### Returns

Length of semi-minor axis.

### 6.555.3.14 SPHERE()

```
Ellipsoid Digikam::Ellipsoid::SPHERE ( ) [static]
```

A sphere with a radius of 6371000 metres. Spheres use a simpler algorithm for orthodromic distance computation, which may be faster and more robust.

### 6.555.3.15 WGS84()

```
Ellipsoid Digikam::Ellipsoid::WGS84 ( ) [static]
```

WGS 1984 ellipsoid with axis in metres. This ellipsoid is used in GPS systems and is the default for most `org.kde.geotools` packages.

## 6.555.4 Member Data Documentation

### 6.555.4.1 m\_inverseFlattening

```
double Digikam::Ellipsoid::m_inverseFlattening = 0.0 [protected]
```

The inverse of the flattening value, or `DBL_MAX` if the ellipsoid is a sphere.

#### See also

`getInverseFlattening`



#### 6.555.4.2 m\_ivfDefinitive

```
bool Digikam::Ellipsoid::m_ivfDefinitive = false [protected]
```

Tells if the Inverse Flattening definitive for this ellipsoid.

See also

[isIvfDefinitive](#)

#### 6.555.4.3 m\_semiMajorAxis

```
double Digikam::Ellipsoid::m_semiMajorAxis = 0.0 [protected]
```

The equatorial radius.

See also

[getSemiMajorAxis](#)

#### 6.555.4.4 m\_semiMinorAxis

```
double Digikam::Ellipsoid::m_semiMinorAxis = 0.0 [protected]
```

The polar radius.

See also

[getSemiMinorAxis](#)

## 6.556 Digikam::EmbossFilter Class Reference

Inheritance diagram for Digikam::EmbossFilter:



**Public Member Functions**

- **EmbossFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, int depth=30)
- **EmbossFilter** (QObject \*const parent=nullptr)
- [FilterAction](#) filterAction () override
- QString filterIdentifier () const override
- void readParameters (const [FilterAction](#) &action) override

**Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)**

- virtual void cancelFilter ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & filterName ()
- int filterVersion () const
- [DImg](#) getTargetImage ()
- QList< int > multithreadedSteps (int stop, int start=0) const
- virtual bool parametersSuccessfullyRead () const
- virtual QString readParametersError (const [FilterAction](#) &actionThatFailed) const
- void setFilterName (const QString &name)
- void setFilterVersion (int version)
- void setOriginalImage (const [DImg](#) &orgImage)
- void setupAndStartDirectly (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void setupFilter (const [DImg](#) &orgImage)
- virtual void startFilter ()
- virtual void startFilterDirectly ()
- virtual QList< int > supportedVersions () const

**Public Member Functions inherited from [Digikam::DynamicThread](#)**

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool isFinished () const
- bool isRunning () const
- QThread::Priority priority () const
- void setEmitSignals (bool emitThem)
- void setPriority (QThread::Priority priority)
- State state () const
- ~[DynamicThread](#) () override

**Static Public Member Functions**

- static int CurrentVersion ()
- static QString DisplayableName ()
- static QString FilterIdentifier ()
- static QList< int > SupportedVersions ()

**Additional Inherited Members****Public Types inherited from [Digikam::DynamicThread](#)**

- enum State { Inactive , Scheduled , Running , Deactivating }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.556.1 Member Function Documentation

### 6.556.1.1 filterAction()

`FilterAction` Digikam::EmbossFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.556.1.2 filterIdentifier()

`QString` Digikam::EmbossFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

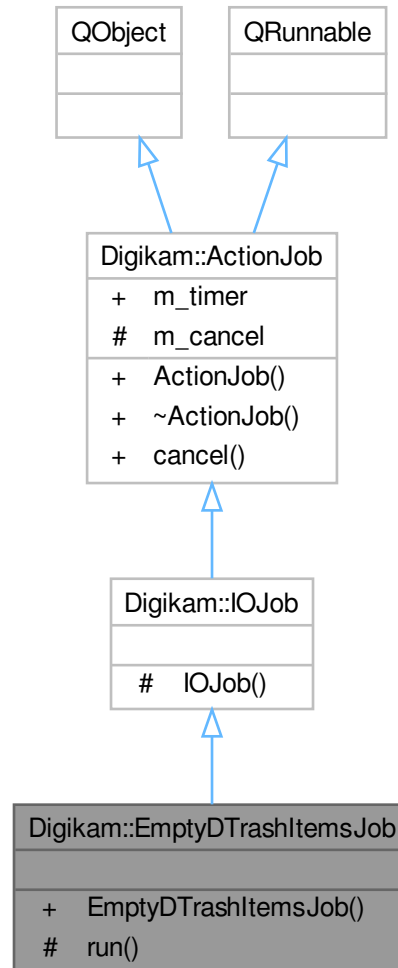
### 6.556.1.3 readParameters()

```
void Digikam::EmbossFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.557 Digikam::EmptyDTrashItemsJob Class Reference

Inheritance diagram for Digikam::EmptyDTrashItemsJob:



### Public Member Functions

- `EmptyDTrashItemsJob (IOJobData *const data)`

### Public Member Functions inherited from Digikam::ActionJob

- `ActionJob (QObject *const parent=nullptr)`
- `~ActionJob ()` override

### Protected Member Functions

- `void run ()` override

### Additional Inherited Members

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Signals inherited from [Digikam::IOJob](#)

- void **signalError** (const QString &errMsg)
- void **signalOneProcessed** (const QUrl &url)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Attributes inherited from [Digikam::ActionJob](#)

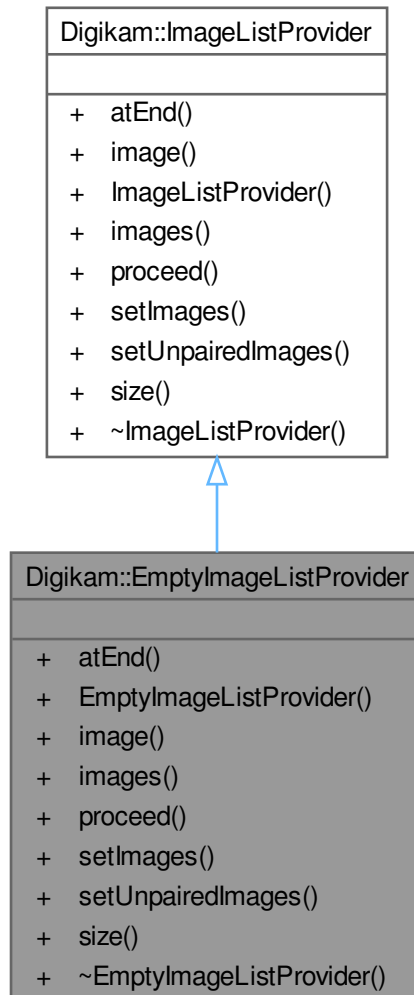
- QElapsedTimer [m\\_timer](#)

### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.558 Digikam::EmptyImageListProvider Class Reference

Inheritance diagram for Digikam::EmptyImageListProvider:



### Public Member Functions

- bool `atEnd` () const override
- `QPair< QImage *, QString >` `image` () override
- `QList< QPair< QImage *, QString > >` `images` () override
- void `proceed` (int steps=1) override
- void `setImages` (const `QList< QPair< QImage *, QString > >` &) override
- void `setUnpairedImages` (const `QList< QImage * >` &) override
- int `size` () const override



## 6.558.1 Member Function Documentation

### 6.558.1.1 atEnd()

```
bool Digikam::EmptyImageListProvider::atEnd ( ) const [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

### 6.558.1.2 image()

```
QPair< QImage *, QString > Digikam::EmptyImageListProvider::image ( ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

### 6.558.1.3 images()

```
QList< QPair< QImage *, QString > > Digikam::EmptyImageListProvider::images ( ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

### 6.558.1.4 proceed()

```
void Digikam::EmptyImageListProvider::proceed (
    int steps = 1 ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

### 6.558.1.5 setImages()

```
void Digikam::EmptyImageListProvider::setImages (
    const QList< QPair< QImage *, QString > > & ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

### 6.558.1.6 setUnpairedImages()

```
void Digikam::EmptyImageListProvider::setUnpairedImages (
    const QList< QImage * > & ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

### 6.558.1.7 size()

```
int Digikam::EmptyImageListProvider::size ( ) const [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

## 6.559 Digikam::EqualizeFilter Class Reference

Inheritance diagram for Digikam::EqualizeFilter:



### Public Member Functions

- **EqualizeFilter** (`DImg *const orgImage, const DImg *const reflImage, QObject *const parent=nullptr`)
- **EqualizeFilter** (`QObject *const parent=nullptr`)

- [FilterAction filterAction](#) () override
- [QString filterIdentifier](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.559.1 Member Function Documentation

### 6.559.1.1 filterAction()

`FilterAction` Digikam::EqualizeFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.559.1.2 filterIdentifier()

`QString` Digikam::EqualizeFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

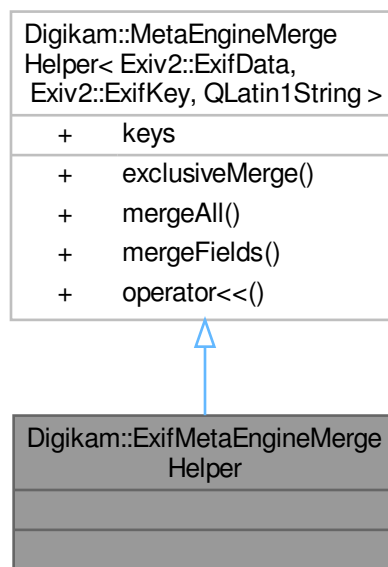
### 6.559.1.3 readParameters()

```
void Digikam::EqualizeFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.560 Digikam::ExifMetaEngineMergeHelper Class Reference

Inheritance diagram for Digikam::ExifMetaEngineMergeHelper:



### Additional Inherited Members

#### Public Member Functions inherited from

[Digikam::MetaEngineMergeHelper](#)< [Exiv2::ExifData](#), [Exiv2::ExifKey](#), [QLatin1String](#) >

- void [exclusiveMerge](#) (const [Exiv2::ExifData](#) &src, [Exiv2::ExifData](#) &dest)
- void [mergeAll](#) (const [Exiv2::ExifData](#) &src, [Exiv2::ExifData](#) &dest)
- void [mergeFields](#) (const [Exiv2::ExifData](#) &src, [Exiv2::ExifData](#) &dest)
- [MetaEngineMergeHelper](#) & **operator**<< (const [QLatin1String](#) &key)

#### Public Attributes inherited from

[Digikam::MetaEngineMergeHelper](#)< [Exiv2::ExifData](#), [Exiv2::ExifKey](#), [QLatin1String](#) >

- [QList](#)< [QLatin1String](#) > **keys**

## 6.561 Digikam::ExifToolBinary Class Reference

Inheritance diagram for Digikam::ExifToolBinary:



### Public Member Functions

- **ExifToolBinary** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DBinaryIface](#)

- virtual QString **baseName** () const
- virtual bool **checkDir** ()
- virtual bool **checkDirForPath** (const QString &path)
- **DBinaryIface** (const QString &binaryName, const QString &minimalVersion, const QString &header, const int headerLine, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- **DBinaryIface** (const QString &binaryName, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- const QString & **description** () const
- bool **developmentVersion** () const
- virtual QString **directory** () const
- bool **hasError** () const
- bool **isFound** () const
- bool **isValid** () const
- virtual QString **minimalVersion** () const
- virtual QString **path** () const
- virtual QString **path** (const QString &dir) const
- virtual QString **projectName** () const
- virtual bool **recheckDirectories** ()
- virtual void **setup** (const QString &prev=QString())
- virtual QUrl **url** () const
- const QString & **version** () const
- bool **versionsRight** () const
- bool **versionsRight** (const float) const

## Additional Inherited Members

## Public Slots inherited from [Digikam::DBinaryIface](#)

- virtual void **slotAddPossibleSearchDirectory** (const QString &dir)
- virtual void **slotAddSearchDirectory** (const QString &dir)
- virtual void **slotNavigateAndCheck** ()

## Signals inherited from [Digikam::DBinaryIface](#)

- void **signalBinaryValid** ()
- void **signalSearchDirectoryAdded** (const QString &dir)

## Static Public Member Functions inherited from [Digikam::DBinaryIface](#)

- static QString **goodBaseName** (const QString &b)

## Protected Member Functions inherited from [Digikam::DBinaryIface](#)

- QString **findHeader** (const QStringList &output, const QString &header) const
- virtual bool **parseHeader** (const QString &output)
- virtual QString **readConfig** ()
- void **setVersion** (QString &version)
- virtual void **writeConfig** ()

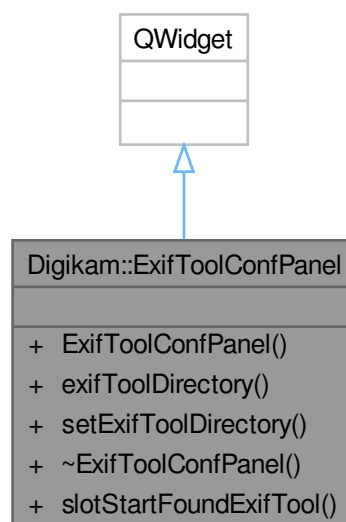


## Protected Attributes inherited from [Digikam::DBinaryIface](#)

- const QStringList **m\_binaryArguments**
- const QString **m\_binaryBaseName**
- QLabel \* **m\_binaryLabel** = nullptr
- const bool **m\_checkVersion**
- const QString **m\_configGroup**
- QString **m\_description**
- bool **m\_developmentVersion** = false
- QLabel \* **m\_downloadButton** = nullptr
- bool **m\_hasError** = false
- const int **m\_headerLine**
- const QString **m\_headerStarts**
- bool **m\_isFound** = false
- QLineEdit \* **m\_lineEdit** = nullptr
- const QString **m\_minimalVersion**
- QPushButton \* **m\_pathButton** = nullptr
- QString **m\_pathDir** = QLatin1String("")
- QFrame \* **m\_pathWidget** = nullptr
- const QString **m\_projectName**
- QSet< QString > **m\_searchPaths**
- QLabel \* **m\_statusIcon** = nullptr
- const QUrl **m\_url**
- QString **m\_version** = QLatin1String("")
- QLabel \* **m\_versionLabel** = nullptr

## 6.562 Digikam::ExifToolConfPanel Class Reference

Inheritance diagram for Digikam::ExifToolConfPanel:



**Public Slots**

- void **slotStartFoundExifTool** ()

**Signals**

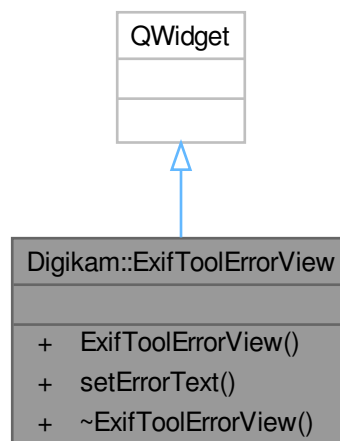
- void **signalExifToolSettingsChanged** (bool available)

**Public Member Functions**

- **ExifToolConfPanel** (QWidget \*const parent=nullptr)
- QString **exifToolDirectory** () const
- void **setExifToolDirectory** (const QString &dir)

**6.563 Digikam::ExifToolErrorView Class Reference**

Inheritance diagram for Digikam::ExifToolErrorView:

**Signals**

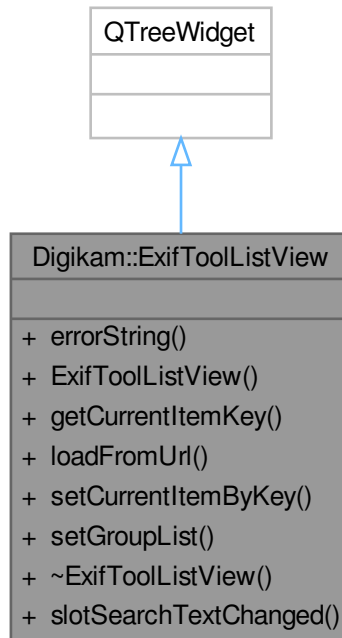
- void **signalSetupExifTool** ()

**Public Member Functions**

- **ExifToolErrorView** (QWidget \*const parent)
- void **setErrorText** (const QString &err)

## 6.564 Digikam::ExifToolListView Class Reference

Inheritance diagram for Digikam::ExifToolListView:



### Public Slots

- void **slotSearchTextChanged** (const [SearchTextSettings](#) &)

### Signals

- void **signalLoadingResult** (bool ok)
- void **signalTextFilterMatch** (bool)

### Public Member Functions

- QString **errorString** () const
- **ExifToolListView** (QWidget \*const parent)
- QString **getCurrentItemKey** () const
- void **loadFromUrl** (const QUrl &url)
- void **setCurrentItemByKey** (const QString &itemKey)
- void **setGroupList** (const QStringList &tagsFilter, const QStringList &keysFilter=QStringList())

## 6.564.1 Member Function Documentation

### 6.564.1.1 setGroupList()

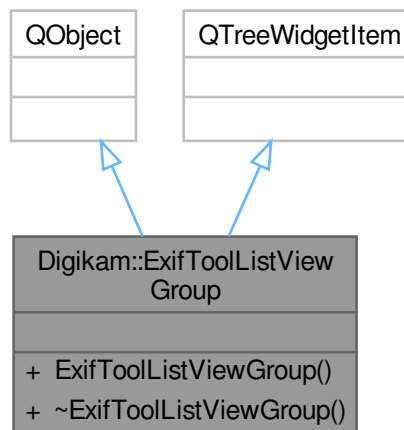
```
void Digikam::ExifToolListView::setGroupList (
    const QStringList & tagsFilter,
    const QStringList & keysFilter = QStringList() )
```

Key is formatted like this:

EXIF.ExifIFD.Image.ExposureCompensation File.File.Other.FileType Composite.Composite.Time.SubSecModify↔  
 Date File.System.Time.FileNodeChangeDate File.System.Other.FileSize EXIF.GPS.Location.GPSLongitude ICC↔  
 \_Profile.ICC-header.Image.ProfileCreator EXIF.IFD1.Image.ThumbnailOffset JFIF.JFIF.Image.YResolution ICC\_↔  
 Profile.ICC\_Profile.Image.GreenMatrixColumn

## 6.565 Digikam::ExifToolListViewGroup Class Reference

Inheritance diagram for Digikam::ExifToolListViewGroup:

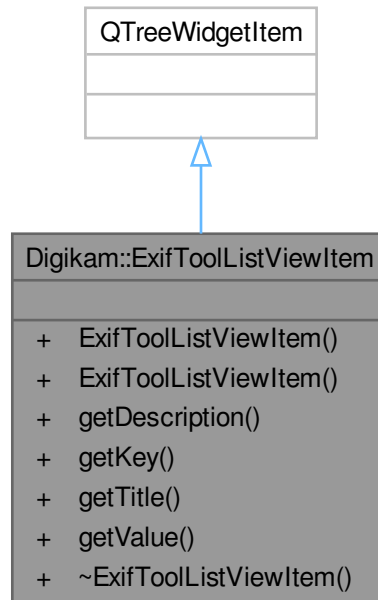


### Public Member Functions

- **ExifToolListViewGroup** (QTreeWidgetItem \*const parent, const QString &group)

## 6.566 Digikam::ExifToolListViewItem Class Reference

Inheritance diagram for Digikam::ExifToolListViewItem:

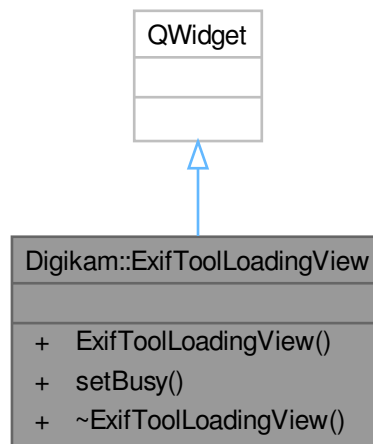


### Public Member Functions

- **ExifToolListViewItem** ([ExifToolListViewGroup](#) \*const parent, const QString &key)
- **ExifToolListViewItem** ([ExifToolListViewGroup](#) \*const parent, const QString &key, const QString &value, const QString &desc)
- QString **getDescription** () const
- QString **getKey** () const
- QString **getTitle** () const
- QString **getValue** () const

## 6.567 Digikam::ExifToolLoadingView Class Reference

Inheritance diagram for Digikam::ExifToolLoadingView:

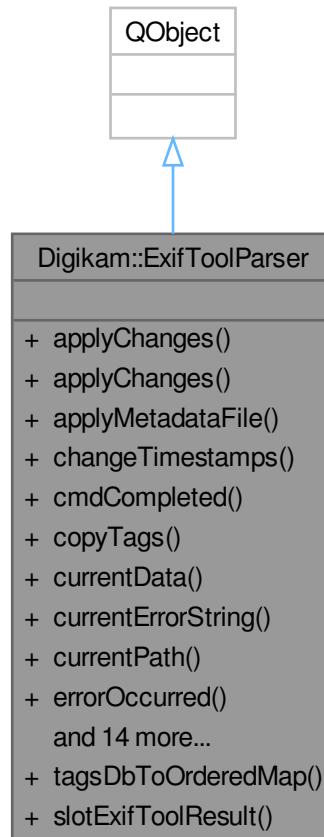


### Public Member Functions

- **ExifToolLoadingView** (`QWidget *const parent`)
- void **setBusy** (`bool b`)

## 6.568 Digikam::ExifToolParser Class Reference

Inheritance diagram for Digikam::ExifToolParser:



### Classes

- class [Private](#)

### Public Types

- typedef `QHash< QString, QVariantList >` [ExifToolData](#)

### Public Slots

- void **slotExifToolResult** (int cmdId)

### Signals

- void **signalExifToolAsyncData** (const [ExifToolParser::ExifToolData](#) &map)
- void **signalExifToolDataAvailable** ()

## Public Member Functions

- bool [applyChanges](#) (const QString &path, const [ExifToolData](#) &newTags)
- bool [applyChanges](#) (const QString &path, const QString &exvTempFile, bool hasExif=true, bool hasXmp=true, bool hasCSet=false)
- bool [applyMetadataFile](#) (const QString &path, const QString &meta)
- bool [changeTimestamps](#) (const QString &path, const QDateTime &dateTime)
- void **cmdCompleted** (const [ExifToolProcess::Result](#) &result)
 

*ExifTool Output Management Methods. See [exiftoolparser\\_output.cpp](#) for details.*
- bool [copyTags](#) (const QString &src, const QString &dst, unsigned char copyOps, unsigned char writeModes=[ExifToolProcess::ALL\\_MODES](#))
- [ExifToolData](#) **currentData** () const
- QString **currentErrorString** () const
- QString **currentPath** () const
- void **errorOccurred** (const [ExifToolProcess::Result](#) &result, QProcess::ProcessError error, const QString &description)
- bool [exifToolAvailable](#) () const
- **ExifToolParser** (QObject \*const parent, bool async=false)
 

*Constructor, Destructor, and Configuration Accessors. See [exiftoolparser.cpp](#) for details.*
- void **finished** ()
- bool [load](#) (const QString &path)
 

*ExifTool Command Methods. See [exiftoolparser\\_command.cpp](#) for details.*
- bool [loadChunk](#) (const QString &path, bool copyToAll=false)
- bool [readableFormats](#) ()
- void **setExifToolProgram** (const QString &path)
- void [setOutputStream](#) (int cmdAction, const QByteArray &cmdOutputChannel, const QByteArray &cmdErrorChannel)
- bool [tagsDatabase](#) ()
- bool [translateTags](#) (const QString &path, unsigned char transOps)
- bool [translationsList](#) ()
- bool [version](#) ()
- bool [writableFormats](#) ()

## Static Public Member Functions

- static [MetaEngine::TagsMap](#) [tagsDbToOrderedMap](#) (const [ExifToolData](#) &tagsDb)

## 6.568.1 Member Typedef Documentation

### 6.568.1.1 ExifToolData

```
typedef QHash<QString, QVariantList> DigiKam::ExifToolParser::ExifToolData
```

A map used to store ExifTool data shared with [ExifToolProcess](#) class:

With [load\(\)](#) method, the container is used to get a map of ExifTool tag name as key and tags properties as values: key = ExifTool Tag name (QString - ExifTool Group 0.1.2.4.6) See -G Exiftool option ( [https://exiftool.org/exiftool\\_pod.html#Input-output-text-formatting](https://exiftool.org/exiftool_pod.html#Input-output-text-formatting)). values = ExifTool Tag value (QString). ExifTool Tag type (QString). ExifTool Tag description (QString). ExifTool Tag numerical value (QString) - available if any .

With [loadChunk\(\)](#) method, the container is used to get a EXV chunk as value: key = "EXV" (QString). value = the Exiv2 metadata container (QByteArray).



With [applyChanges\(\)](#) method, the container is used as argument to store tuple of ExifTool tag name as key and tag value: key = ExifTool tag name (QString). value = ExifTool Tag value (QString).

With [readableFormats\(\)](#) method, the container is used to get a list of upper-case file format extensions supported by ExifTool for reading. key = "READ\_FORMAT" (QString). value = list of pairs (ext,desc) (QStringList)

With [writableFormats\(\)](#) method, the container is used to get a list of upper-case file format extensions supported by ExifTool for writing. key = "WRITE\_FORMAT" (QString). value = list of pairs (ext,desc) (QStringList).

With [translationsList\(\)](#) method, the container is used to get a list of ExifTool languages available for translations. key = "TRANSLATIONS\_LIST" (QString). value = list of languages as strings (aka fr, en, de, es, etc.) (QStringList).

With [tagsDatabase\(\)](#) method, the container is used as argument to store tuple of ExifTool tag name as key and tag description: key = ExifTool tag name (QString). values = ExifTool Tag description (QString). ExifTool Tag type (QString). ExifTool Tag writable (QString).

## 6.568.2 Member Function Documentation

### 6.568.2.1 applyChanges() [1/2]

```
bool Digikam::ExifToolParser::applyChanges (
    const QString & path,
    const ExifToolData & newTags )
```

Apply tag changes to a target file using ExifTool with a list of tag properties. Tags can already exists in target file or new ones can be created. To remove a tag, pass an empty string as value.

#### Parameters

<i>path</i>	is the target files to change.
<i>newTags</i>	is the list of tag properties.

### 6.568.2.2 applyChanges() [2/2]

```
bool Digikam::ExifToolParser::applyChanges (
    const QString & path,
    const QString & exvTempFile,
    bool hasExif = true,
    bool hasXmp = true,
    bool hasCSet = false )
```

Apply tag changes to a target file using ExifTool with a EXV container. Tags can already exists in target file or new ones can be created.

#### Parameters

<i>path</i>	is the target files to change.
<i>exvTempFile</i>	is the list of changes embedded in EXV container.
<i>hasExif</i>	if the EXV container has Exif metadata restore MarkerNotes.
<i>hasXmp</i>	if the EXV container has Xmp metadata.
<i>hasCSet</i>	if the EXV container has characters set information.

### 6.568.2.3 applyMetadataFile()

```
bool Digikam::ExifToolParser::applyMetadataFile (
    const QString & path,
    const QString & meta )
```

Apply a file with metadata to the target file.

#### Parameters

<i>path</i>	is the target file to change.
<i>meta</i>	is the metadata file.

### 6.568.2.4 changeTimestamps()

```
bool Digikam::ExifToolParser::changeTimestamps (
    const QString & path,
    const QDateTime & dateTime )
```

Change all timestamps of the target file using ExifTool.

#### Parameters

<i>path</i>	is the target file to change.
<i>dateTime</i>	is the date/time.

### 6.568.2.5 copyTags()

```
bool Digikam::ExifToolParser::copyTags (
    const QString & src,
    const QString & dst,
    unsigned char copyOps,
    unsigned char writeModes = ExifToolProcess::ALL_MODES )
```

Copy group of tags from one source file to a destination file, following copy operations defined by 'copyOps'.

#### Parameters

<i>src</i>	must be a readable file format supported by ExifTool.
<i>dst</i>	must be a writable file format supported by ExifTool.
<i>copyOps</i>	is a OR combination of <a href="#">ExifToolProcess::CopyTagsSource</a> values.
<i>writeModes</i>	is a OR combination of <a href="#">ExifToolProcess::WritingTagsMode</a> values.

### 6.568.2.6 exifToolAvailable()

```
bool Digikam::ExifToolParser::exifToolAvailable ( ) const
```

Check the ExifTool program availability.

**6.568.2.7 load()**

```
bool Digikam::ExifToolParser::load (
    const QString & path )
```

Load all metadata with ExifTool from a file. Use `currentData()` to get the ExifTool map.

**6.568.2.8 loadChunk()**

```
bool Digikam::ExifToolParser::loadChunk (
    const QString & path,
    bool copyToAll = false )
```

Load Exif, Iptc, and Xmp chunk as Exiv2 EXV byte-array from a file. Use `currentData()` to get the container.

**6.568.2.9 readableFormats()**

```
bool Digikam::ExifToolParser::readableFormats ( )
```

Return a list of readable file format extensions. Use `currentData()` to get the container as `QStringList`.

**6.568.2.10 setOutputStream()**

```
void Digikam::ExifToolParser::setOutputStream (
    int cmdAction,
    const QByteArray & cmdOutputChannel,
    const QByteArray & cmdErrorChannel )
```

Unit-test method to check ExifTool stream parsing.

**6.568.2.11 tagsDatabase()**

```
bool Digikam::ExifToolParser::tagsDatabase ( )
```

Return a list of all tags from ExifTool database. Use `currentData()` to get the container.

**Warning**

: This method get whole ExifTool database in XML format and take age.

**6.568.2.12 tagsDbToOrderedMap()**

```
MetaEngine::TagsMap Digikam::ExifToolParser::tagsDbToOrderedMap (
    const ExifToolData & tagsDb ) [static]
```

Helper conversion method to translate unordered tags database hash-table to ordered map. Tag are formatted like this:

```
EXIF.IFD0.Image.XResolution EXIF.IFD0.Image.YCbCrCoefficients EXIF.IFD0.Image.YCbCrPositioning EXIF.IFD0.Image.YCbCrSubSampling EXIF.IFD0.Image.YClipPathUnits EXIF.IFD0.Image.YPosition EXIF.IFD0.Image.YResolution FITS.FITS.Image.Author FITS.FITS.Image.Background FITS.FITS.Image.CreateDate FITS.FITS.Image.Instrument FITS.FITS.Image.Object FITS.FITS.Image.ObservationDate
```

**6.568.2.13 translateTags()**

```
bool Digikam::ExifToolParser::translateTags (
    const QString & path,
    unsigned char transOps )
```

Translate group of tags in file.

## Parameters

<i>path</i>	must be a readable file format supported by ExifTool.
<i>transOps</i>	is a OR combination of <a href="#">ExifToolProcess::TranslateTagsOps</a> values.

**6.568.2.14 translationsList()**

```
bool Digikam::ExifToolParser::translationsList ( )
```

Return a list of available translations. Use `currentData()` to get the container as `QStringList`.

**6.568.2.15 version()**

```
bool Digikam::ExifToolParser::version ( )
```

Return the current version of ExifTool. Use `currentData()` to get the container as `QString`.

**6.568.2.16 writableFormats()**

```
bool Digikam::ExifToolParser::writableFormats ( )
```

Return a list of writable file format extensions. Use `currentData()` to get the container as `QStringList`.

**6.569 Digikam::ExifToolParser::Private Class Reference****Public Member Functions**

- `QString` [actionString](#) (int cmdAction) const
- `QByteArray` [filePathEncoding](#) (const `QFileInfo` &fi) const
- void [jumpToResultCommand](#) (const [ExifToolProcess::Result](#) &result, int cmdId)
- void [prepareFileAndSidecar](#) (`QByteArrayList` &cmdArgs, const `QFileInfo` &fi)
- void [prepareProcess](#) ()
- **Private** ([ExifToolParser](#) \*const q)
- bool [startProcess](#) (const `QByteArrayList` &cmdArgs, [ExifToolProcess::Action](#) cmdAction)

**Public Attributes**

- `QTemporaryFile` [argsFile](#)  
*Temporary file to store Exiftool arg config file.*
- bool [async](#) = false
- `QList< int >` [asyncRunning](#)
- `QString` [currentPath](#)  
*Current file path processed by ExifTool.*
- `QString` [errorString](#)  
*Current error string from the last started ExifTool process.*
- [ExifToolData](#) [exifToolData](#)  
*Current ExifTool data (input or output depending of the called method.*
- `QMutex` [mutex](#)
- [ExifToolParser](#) \* [pp](#) = nullptr
- [ExifToolProcess](#) \* [proc](#) = nullptr  
*ExifTool process instance.*

## 6.569.1 Member Function Documentation

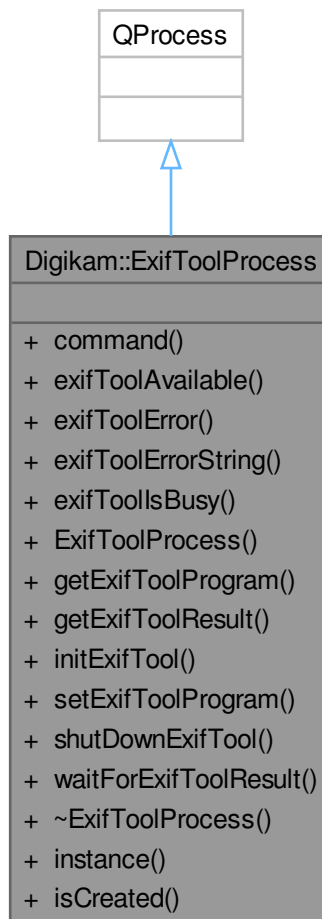
### 6.569.1.1 actionString()

```
QString Digikam::ExifToolParser::Private::actionString (
    int cmdAction ) const
```

Returns a string for an action.

## 6.570 Digikam::ExifToolProcess Class Reference

Inheritance diagram for Digikam::ExifToolProcess:



### Classes

- class [Private](#)
- class [Result](#)

## Public Types

- enum [Action](#) {  
[LOAD\\_METADATA](#) = 0 , [LOAD\\_CHUNKS](#) , [APPLY\\_CHANGES](#) , [APPLY\\_CHANGES\\_EXV](#) ,  
[APPLY\\_METADATA\\_FILE](#) , [CHANGE\\_TIMESTAMPS](#) , [READ\\_FORMATS](#) , [WRITE\\_FORMATS](#) ,  
[TRANSLATIONS\\_LIST](#) , [TAGS\\_DATABASE](#) , [VERSION\\_STRING](#) , [COPY\\_TAGS](#) ,  
[TRANS\\_TAGS](#) , [NO\\_ACTION](#) }
- enum [CopyTagsSource](#) {  
[COPY\\_EXIF](#) = 0x01 , [COPY\\_MAKERNOTES](#) = 0x02 , [RESTORE\\_PREVIEW](#) = 0x04 , [COPY\\_IPTC](#) = 0x08 ,  
[COPY\\_XMP](#) = 0x10 , [COPY\\_ICC](#) = 0x20 , [COPY\\_ALL](#) = 0x40 , [COPY\\_NONE](#) = 0x80 }
- enum [ResultStatus](#) { [COMMAND\\_RESULT](#) = 0 , [FINISH\\_RESULT](#) , [ERROR\\_RESULT](#) }
- enum [TranslateTagsOps](#) { [TRANS\\_ALL\\_XMP](#) = 0x01 , [TRANS\\_ALL\\_IPTC](#) = 0x02 , [TRANS\\_ALL\\_EXIF](#) =  
0x04 }
- enum [WritingTagsMode](#) { [WRITE\\_EXISTING\\_TAGS](#) = 0x01 , [CREATE\\_NEW\\_TAGS](#) = 0x02 , [CREATE\\_NEW\\_GROUPS](#)  
= 0x04 , [ALL\\_MODES](#) }

## Signals

- void [signalChangeProgram](#) (const QString &etExePath)
- void [signalExecNextCmd](#) ()
- void [signalExifToolResult](#) (int cmdId)

## Public Member Functions

- int [command](#) (const QByteArrayList &args, [Action](#) ac)
- bool [exifToolAvailable](#) () const
- QProcess::ProcessError [exifToolError](#) () const
- QString [exifToolErrorString](#) () const
- bool [exifToolsBusy](#) () const
- [ExifToolProcess](#) ()
- QString [getExifToolProgram](#) () const
- [ExifToolProcess::Result](#) [getExifToolResult](#) (int cmdId) const
- void [initExifTool](#) ()
- void [setExifToolProgram](#) (const QString &etExePath)
- void [shutDownExifTool](#) ()
- [ExifToolProcess::Result](#) [waitForExifToolResult](#) (int cmdId) const
- [~ExifToolProcess](#) ()

## Static Public Member Functions

- static [ExifToolProcess](#) \* [instance](#) ()  
*Q\_GLOBAL\_STATIC implementation.*
- static bool [isCreated](#) ()

## 6.570.1 Member Enumeration Documentation

### 6.570.1.1 Action

enum [Digikam::ExifToolProcess::Action](#)

ExifTool actions to process.

## Enumerator

LOAD_METADATA	Load all metadata from a file with ExifTool.
LOAD_CHUNKS	Load Exif, Iptc, and Xmp chunks from a file as byte-array for <a href="#">MetaEngine</a> .
APPLY_CHANGES	Apply tag changes in a file with ExifTool.
APPLY_CHANGES_EXV	Apply tag changes in a file with ExifTool using an EXV container.
APPLY_METADATA_FILE	Apply a metadata file to a file with ExifTool.
CHANGE_TIMESTAMPS	Change all timestamps in a file with ExifTool.
READ_FORMATS	Return the list of readable ExifTool file formats.
WRITE_FORMATS	Return the list of writable ExifTool file formats.
TRANSLATIONS_LIST	List of ExifTool languages available for translations.
TAGS_DATABASE	List of ExifTool tags from database.
VERSION_STRING	Return the ExifTool version as string.
COPY_TAGS	Copy tags from one file to another one. See CopyTagsSource enum for details.
TRANS_TAGS	Translate tags in file. See TranslateTagsOps enum for details.
NO_ACTION	Last value from this list. Do nothing.

## 6.570.1.2 CopyTagsSource

enum [Digikam::ExifToolProcess::CopyTagsSource](#)

Possible copying tags operations to OR combine with COPY\_TAGS action.

## Enumerator

COPY_EXIF	Copy all Exif Tags from source file.
COPY_MAKERNOTES	Copy all Makernotes tags from source file.
RESTORE_PREVIEW	Restore preview image from source file.
COPY_IPTC	Copy all Iptc tags from source file.
COPY_XMP	Copy all Xmp tags from source file.
COPY_ICC	Copy ICC profile from source file.
COPY_ALL	Copy all tags from source file.
COPY_NONE	No copy operation.

## 6.570.1.3 ResultStatus

enum [Digikam::ExifToolProcess::ResultStatus](#)

Command result state.

## 6.570.1.4 TranslateTagsOps

enum [Digikam::ExifToolProcess::TranslateTagsOps](#)

Possible translating tags operations to OR combine with COPY\_TAGS action.

## Enumerator

TRANS_ALL_XMP	Translate all existing Tags from source file to Xmp.
TRANS_ALL_IPTC	Translate all existing Tags from source file to Iptc.
TRANS_ALL_EXIF	Translate all existing Tags from source file to Exif.

**6.570.1.5 WritingTagsMode**

```
enum Digikam::ExifToolProcess::WritingTagsMode
```

Possible writing tags mode to OR combine with COPY\_TAGS action.

## Enumerator

WRITE_EXISTING_TAGS	Overwrite existing tags.
CREATE_NEW_TAGS	Create new tags.
CREATE_NEW_GROUPS	Create new groups if necessary.

**6.570.2 Constructor & Destructor Documentation****6.570.2.1 ExifToolProcess()**

```
Digikam::ExifToolProcess::ExifToolProcess ( )
```

Constructs a [ExifToolProcess](#).

**6.570.2.2 ~ExifToolProcess()**

```
Digikam::ExifToolProcess::~~ExifToolProcess ( )
```

Destructs the [ExifToolProcess](#) object, i.e., killing the process. Note that this function will not return until the process is terminated.

**6.570.3 Member Function Documentation****6.570.3.1 command()**

```
int Digikam::ExifToolProcess::command (
    const QByteArrayList & args,
    Action ac )
```

Send a command to exiftool process. This function can be called from another thread. Return 0: ExifTool not running, write channel is closed or args is empty.



### 6.570.3.2 `exifToolAvailable()`

```
bool Digikam::ExifToolProcess::exifToolAvailable ( ) const
```

Returns true if [ExifToolProcess](#) is available (process state == Running)

### 6.570.3.3 `exifToolError()`

```
QProcess::ProcessError Digikam::ExifToolProcess::exifToolError ( ) const
```

Returns the type of error that occurred last.

### 6.570.3.4 `exifToolErrorString()`

```
QString Digikam::ExifToolProcess::exifToolErrorString ( ) const
```

Returns an error message.

### 6.570.3.5 `exifToolsBusy()`

```
bool Digikam::ExifToolProcess::exifToolIsBusy ( ) const
```

Returns true if a command is running.

### 6.570.3.6 `getExifToolResult()`

```
ExifToolProcess::Result Digikam::ExifToolProcess::getExifToolResult (
    int cmdId ) const
```

Returns the [ExifToolProcess](#) result.

### 6.570.3.7 `initExifTool()`

```
void Digikam::ExifToolProcess::initExifTool ( )
```

[Setup](#) connections, apply Settings and start ExifTool process. This function cannot be called from another thread.

### 6.570.3.8 `setExifToolProgram()`

```
void Digikam::ExifToolProcess::setExifToolProgram (
    const QString & etExePath )
```

Change the ExifTool path configuration. This function can be called from another thread.

### 6.570.3.9 shutDownExifTool()

```
void Digikam::ExifToolProcess::shutDownExifTool ( )
```

Attempts to shut down the ExifTool process. This function cannot be called from another thread.

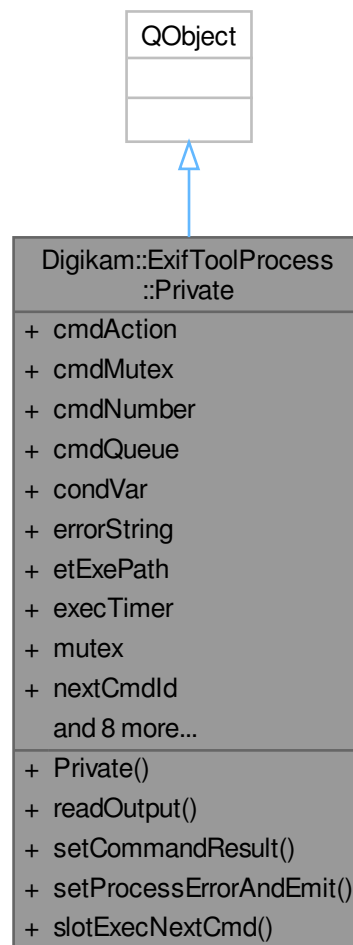
### 6.570.3.10 waitForExifToolResult()

```
ExifToolProcess::Result Digikam::ExifToolProcess::waitForExifToolResult (
    int cmdId ) const
```

WaitCondition for the [ExifToolParser](#) class. Returns the [ExifToolProcess](#) result.

## 6.571 Digikam::ExifToolProcess::Private Class Reference

Inheritance diagram for Digikam::ExifToolProcess::Private:



## Classes

- class [Command](#)

## Public Slots

- void **slotExecNextCmd** ()

## Public Member Functions

- **Private** ([ExifToolProcess](#) \*const q)
- void **readOutput** (const QProcess::ProcessChannel channel)
- void **setCommandResult** (int cmdStatus)
- void **setProcessErrorAndEmit** (QProcess::ProcessError error, const QString &description)

## Public Attributes

- [ExifToolProcess::Action](#) **cmdAction** = [ExifToolProcess::NO\\_ACTION](#)
- QMutex **cmdMutex**
- int **cmdNumber** = 0
- QList< [Command](#) > **cmdQueue**
- QWaitCondition **condVar**
- QString **errorString**
- QString **etExePath**
- QElapsedTimer **execTimer**
- QMutex **mutex**
- int **nextCmdId** = CMD\_ID\_MIN  
*Unique identifier, even in a multi-instances or multi-thread environment.*
- int **outAwait** [2]  
*[0] StandardOutput | [1] ErrorOutput*
- QByteArray **outBuff** [2]  
*[0] StandardOutput | [1] ErrorOutput*
- bool **outReady** [2]  
*[0] StandardOutput | [1] ErrorOutput*
- QString **perlExePath**
- [ExifToolProcess](#) \* **pp** = nullptr
- QProcess::ProcessError **processError** = QProcess::UnknownError
- QMap< int, [ExifToolProcess::Result](#) > **resultMap**
- bool **writeChannelsClosed** = true

## 6.572 Digikam::ExifToolProcess::Private::Command Class Reference

### Public Attributes

- [ExifToolProcess::Action](#) **ac** = [ExifToolProcess::NO\\_ACTION](#)
- QByteArray **argsStr**
- int **id** = 0

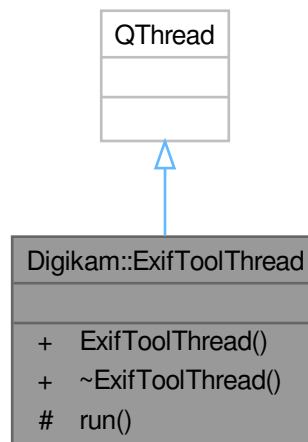
## 6.573 Digikam::ExifToolProcess::Result Class Reference

### Public Attributes

- int **cmdAction** = [ExifToolProcess::NO\\_ACTION](#)
- int **cmdNumber** = 0
- int **cmdStatus** = [ExifToolProcess::COMMAND\\_RESULT](#)
- int **elapsed** = 0
- QByteArray **output**
- bool **waitError** = false

## 6.574 Digikam::ExifToolThread Class Reference

Inheritance diagram for Digikam::ExifToolThread:



### Signals

- void **exifToolProcessStarted** ()

### Public Member Functions

- **ExifToolThread** (QObject \*const parent)

### Protected Member Functions

- void **run** () override

## 6.574.1 Member Function Documentation

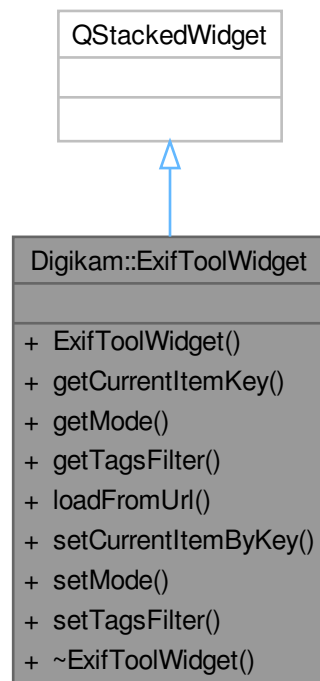
### 6.574.1.1 run()

```
void Digikam::ExifToolThread::run ( ) [override], [protected]
```

Main thread loop.

## 6.575 Digikam::ExifToolWidget Class Reference

Inheritance diagram for Digikam::ExifToolWidget:



### Public Types

- enum `TagFilters` { `NONE = 0` , `PHOTO` , `CUSTOM` }

### Signals

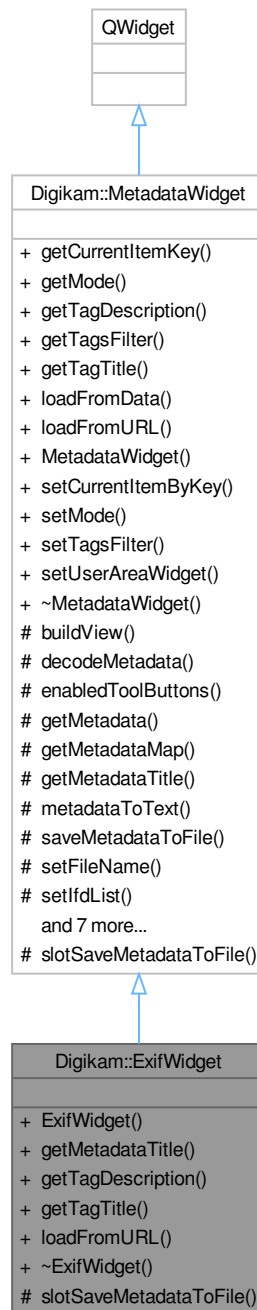
- void `signalSetupExifTool ()`
- void `signalSetupMetadataFilters ()`

**Public Member Functions**

- **ExifToolWidget** (QWidget \*const parent)
- QString **getCurrentItemKey** () const
- int **getMode** () const
- QStringList **getTagsFilter** () const
- void **loadFromUrl** (const QUrl &url)
- void **setCurrentItemByKey** (const QString &itemKey)
- void **setMode** (int mode)
- void **setTagsFilter** (const QStringList &list)

## 6.576 Digikam::ExifWidget Class Reference

Inheritance diagram for Digikam::ExifWidget:



### Public Member Functions

- **ExifWidget** (QWidget \*const parent, const QString &name=QString())
- QString [getMetadataTitle](#) () const override

- QString [getTagDescription](#) (const QString &key) override
- QString [getTagTitle](#) (const QString &key) override
- bool [loadFromURL](#) (const QUrl &url) override

### Public Member Functions inherited from [Digikam::MetadataWidget](#)

- QString [getCurrentItemKey](#) () const
- int [getMode](#) () const
- QStringList [getTagsFilter](#) () const
- virtual bool [loadFromData](#) (const QString &fileName, const [DMetadata](#) &data=[DMetadata](#)())
- [MetadataWidget](#) (QWidget \*const parent, const QString &name=QString())
- void [setCurrentItemByKey](#) (const QString &itemKey)
- void [setMode](#) (int mode)
- void [setTagsFilter](#) (const QStringList &list)
- void [setUserAreaWidget](#) (QWidget \*const w)

### Protected Slots

- void [slotSaveMetadataToFile](#) () override

### Protected Slots inherited from [Digikam::MetadataWidget](#)

- virtual void [slotSaveMetadataToFile](#) ()=0

### Additional Inherited Members

### Public Types inherited from [Digikam::MetadataWidget](#)

- enum [TagFilters](#) { NONE = 0 , PHOTO , CUSTOM }

### Signals inherited from [Digikam::MetadataWidget](#)

- void [signalSetupMetadataFilters](#) ()

### Protected Member Functions inherited from [Digikam::MetadataWidget](#)

- void [enabledToolButtons](#) (bool)
- [DMetadata](#) \* [getMetadata](#) () const
- const [DMetadata::MetaDataMap](#) & [getMetadataMap](#) ()
- QString [metadataToText](#) () const
- QUrl [saveMetadataToFile](#) (const QString &caption, const QString &fileFilter)
- void [setFileName](#) (const QString &fileName)
- void [setIfdList](#) (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &keysFilter, const QStringList &tagsFilter)
- void [setIfdList](#) (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &tagsFilter=QStringList())
- bool [setMetadata](#) (const [DMetadata](#) &data=[DMetadata](#)())
- virtual void [setMetadataEmpty](#) ()
- void [setMetadataMap](#) (const [DMetadata::MetaDataMap](#) &data=[DMetadata::MetaDataMap](#)())
- void [setup](#) ()
- bool [storeMetadataToFile](#) (const QUrl &url, const QByteArray &metaData)
- [MetadataListView](#) \* [view](#) () const



## 6.576.1 Member Function Documentation

### 6.576.1.1 getMetadataTitle()

```
QString Digikam::ExifWidget::getMetadataTitle ( ) const [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

### 6.576.1.2 getTagDescription()

```
QString Digikam::ExifWidget::getTagDescription (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

### 6.576.1.3 getTagTitle()

```
QString Digikam::ExifWidget::getTagTitle (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

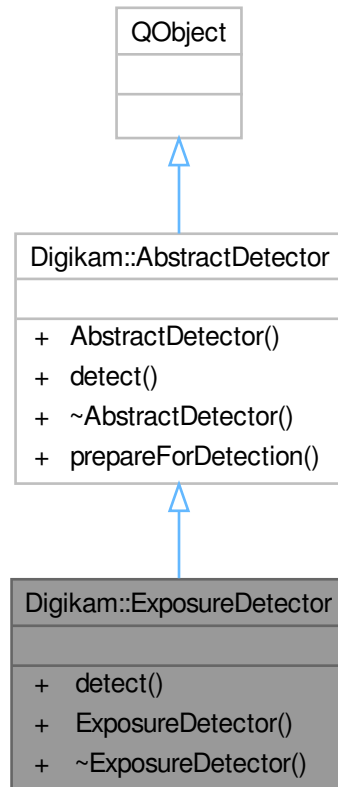
### 6.576.1.4 loadFromURL()

```
bool Digikam::ExifWidget::loadFromURL (
    const QUrl & url ) [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

## 6.577 Digikam::ExposureDetector Class Reference

Inheritance diagram for Digikam::ExposureDetector:



### Public Member Functions

- float [detect](#) (const cv::Mat &image) const override

### Public Member Functions inherited from [Digikam::AbstractDetector](#)

- **AbstractDetector** (QObject \*const parent=nullptr)

### Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::AbstractDetector](#)

- static cv::Mat [prepareForDetection](#) (const [DImg](#) &inputImage)

## 6.577.1 Member Function Documentation

### 6.577.1.1 detect()

```
float Digikam::ExposureDetector::detect (
    const cv::Mat & image ) const [override], [virtual]
```

Implements [Digikam::AbstractDetector](#).

## 6.578 Digikam::ExposureSettingsContainer Class Reference

### Public Attributes

- bool [exposureIndicatorMode](#) = true
- QColor [overExposureColor](#) = Qt::black
- bool [overExposureIndicator](#) = false
- float [overExposurePercent](#) = 1.0
- QColor [underExposureColor](#) = Qt::white
- bool [underExposureIndicator](#) = false
- float [underExposurePercent](#) = 1.0

## 6.578.1 Member Data Documentation

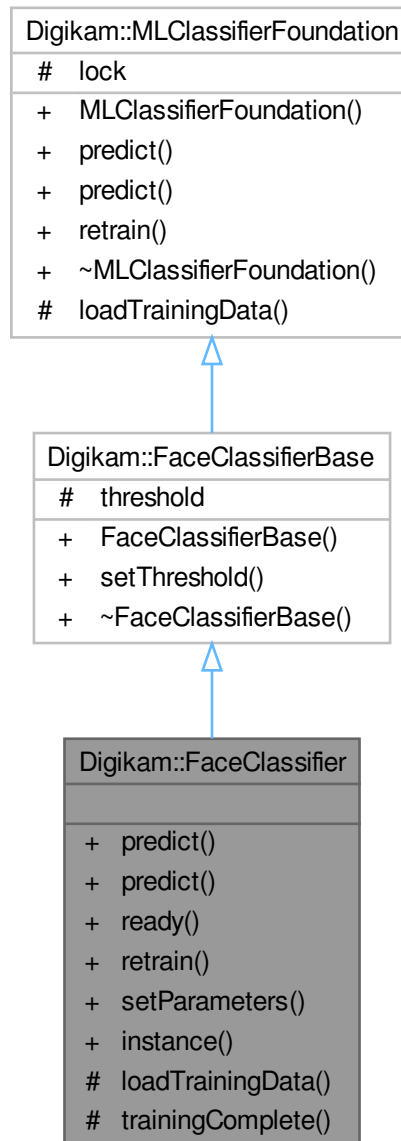
### 6.578.1.1 exposureIndicatorMode

```
bool Digikam::ExposureSettingsContainer::exposureIndicatorMode = true
```

If this option is true, over and under exposure indicators will be displayed only when pure white and pure black color matches, as all color components match the condition in the same time. Else indicators are turn on when one of color components match the condition.

## 6.579 Digikam::FaceClassifier Class Reference

Inheritance diagram for Digikam::FaceClassifier:



### Public Member Functions

- int [predict](#) (const cv::Mat &target) const override
- int [predict](#) (const cv::UMat &target) const override
- bool **ready** () const
- bool [retrain](#) () override
- void **setParameters** (const [FaceScanSettings](#) &parameters)

## Public Member Functions inherited from [Digikam::FaceClassifierBase](#)

- void **setThreshold** (float *\_threshold*)

## Static Public Member Functions

- static [FaceClassifier](#) \* **instance** ()

## Protected Member Functions

- bool **loadTrainingData** () override
- void **trainingComplete** ()

## Friends

- class **FaceClassifierCreator**

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::FaceClassifierBase](#)

- float **threshold** = 0.0F

## Protected Attributes inherited from [Digikam::MLClassifierFoundation](#)

- QReadWriteLock **lock**

## 6.579.1 Member Function Documentation

### 6.579.1.1 loadTrainingData()

```
bool Digikam::FaceClassifier::loadTrainingData ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLClassifierFoundation](#).

### 6.579.1.2 predict() [1/2]

```
int Digikam::FaceClassifier::predict (
    const cv::Mat & target ) const [override], [virtual]
```

Implements [Digikam::MLClassifierFoundation](#).

### 6.579.1.3 predict() [2/2]

```
int Digikam::FaceClassifier::predict (
    const cv::UMat & target ) const [override], [virtual]
```

Implements [Digikam::MLClassifierFoundation](#).

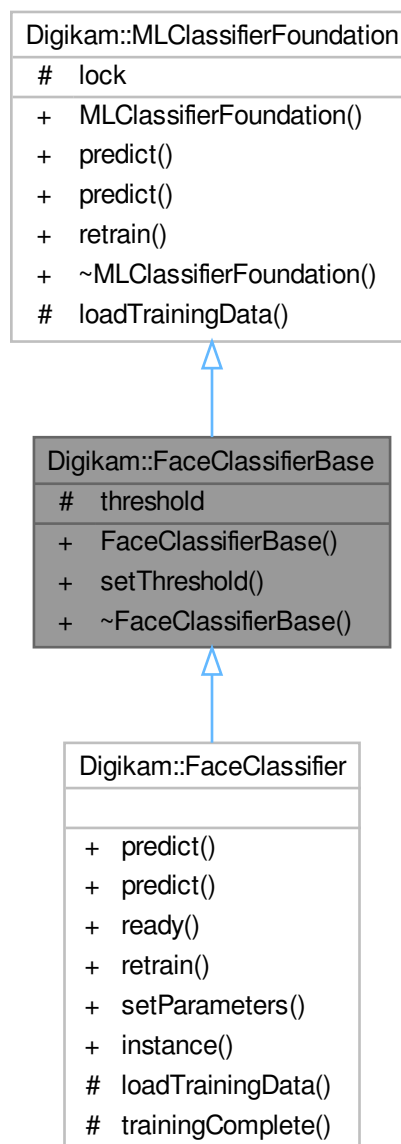
**6.579.1.4 retrain()**

```
bool Digikam::FaceClassifier::retrain ( ) [override], [virtual]
```

Implements [Digikam::MLClassifierFoundation](#).

**6.580 Digikam::FaceClassifierBase Class Reference**

Inheritance diagram for Digikam::FaceClassifierBase:



**Public Member Functions**

- void **setThreshold** (float \_threshold)

**Public Member Functions inherited from [Digikam::MLClassifierFoundation](#)**

- virtual int **predict** (const cv::Mat &target) const =0
- virtual int **predict** (const cv::UMat &target) const =0
- virtual bool **retrain** ()=0

**Protected Attributes**

- float **threshold** = 0.0F

**Protected Attributes inherited from [Digikam::MLClassifierFoundation](#)**

- QReadWriteLock **lock**

**Additional Inherited Members****Protected Member Functions inherited from [Digikam::MLClassifierFoundation](#)**

- virtual bool **loadTrainingData** ()=0

## 6.581 Digikam::FaceDb Class Reference

**Classes**

- class [Private](#)

**Public Member Functions**

- int **addIdentity** () const
- void **clearDNNTraining** ()  
*clearDNNTraining: clear all trained data in the database.*
- void **clearDNNTraining** (const QList< int > &identities)  
*clearDNNTraining: clear*
- void **clearIdentities** ()
- void **deleteIdentity** (const QString &uuid)
- void **deleteIdentity** (int id)
- **FaceDb** ([FaceDbBackend](#) \*const db, [FaceScanSettings::FaceRecognitionModel](#) recModel)
- int **getNumberOfIdentities** () const
- QList< [Identity](#) > **identities** () const
- [Identity](#) **identity** (int id) const
- QList< int > **identityIds** () const
- int **insertFaceVector** (const cv::Mat &faceEmbedding, const int label, const QString &hash) const  
*insertFaceVector: insert a new face embedding to database.*

- bool `integrityCheck ()`
- bool `removeFaceVector (const int id) const`  
*removeFaceVector: remove a face embedding from the database.*
- bool `removeFaceVector (const QString &hash) const`  
*removeFaceVector: remove a face embedding from the database.*
- `BdEngineBackend::QueryState` `setSetting (const QString &keyword, const QString &value)`
- QString `setting (const QString &keyword) const`
- cv::Ptr< cv::ml::TrainData > `trainData () const`  
*trainData: extract train data from database.*
- void `updateIdentity (const Identity &p)`
- void `vacuum ()`

## 6.581.1 Member Function Documentation

### 6.581.1.1 clearDNNTraining()

```
void Digikam::FaceDb::clearDNNTraining (
    const QList< int > & identities )
```

#### Parameters

<i>identities</i>	in the database.
-------------------	------------------

### 6.581.1.2 insertFaceVector()

```
int Digikam::FaceDb::insertFaceVector (
    const cv::Mat & faceEmbedding,
    const int label,
    const QString & hash ) const
```

#### Parameters

<i>faceEmbedding</i>	
<i>label</i>	
<i>hash</i>	

#### Returns

id of newly inserted entry.

### 6.581.1.3 integrityCheck()

```
bool Digikam::FaceDb::integrityCheck ( )
```

Returns true if the integrity of the database is preserved.



**6.581.1.4 removeFaceVector()** [1/2]

```
bool Digikam::FaceDb::removeFaceVector (
    const int id ) const
```

**Parameters**

<i>id</i>	the nodeId (row id) to remove.
-----------	--------------------------------

**Returns**

bool

**6.581.1.5 removeFaceVector() [2/2]**

```
bool Digikam::FaceDb::removeFaceVector (
    const QString & hash ) const
```

**Parameters**

<i>hash</i>	the removeHash (removeHash) to remove.
-------------	--

**Returns**

bool

**6.581.1.6 trainData()**

```
cv::Ptr< cv::ml::TrainData > Digikam::FaceDb::trainData ( ) const
```

**Returns**

the train data instance.

**6.581.1.7 vacuum()**

```
void Digikam::FaceDb::vacuum ( )
```

Shrinks the database.

## 6.582 Digikam::FaceDb::Private Class Reference

**Public Attributes**

- [FaceDbBackend](#) \* **db** = nullptr
- [FaceScanSettings::FaceRecognitionModel](#) **recognizeModel** = [FaceScanSettings::FaceRecognitionModel::OpenFace](#)

## 6.583 Digikam::FaceDbAccess Class Reference

### Public Member Functions

- [FaceDbBackend](#) \* **backend** () const
- [FaceDb](#) \* **db** () const
- [FaceDbAccess](#) ()
- const QString & **lastError** () const
- void **setLastError** (const QString &error)

### Static Public Member Functions

- static bool **checkReadyForUse** ([InitializationObserver](#) \*const observer)
- static void **cleanUpDatabase** ()
- static void **initDbEngineErrorHandler** ([DbEngineErrorHandler](#) \*const errorhandler)
- static bool **isInitialized** ()
- static [DbEngineParameters](#) **parameters** ()
- static void **setParameters** (const [DbEngineParameters](#) &parameters, [FaceScanSettings::FaceRecognitionModel](#) recognizeModel)

### Friends

- class [FaceDbAccessUnlock](#)

## 6.583.1 Constructor & Destructor Documentation

### 6.583.1.1 FaceDbAccess()

```
Digikam::FaceDbAccess::FaceDbAccess ( )
```

This class is written in analogy to [CoreDbAccess](#) (some features stripped off). For documentation, see [coredbaccess.h](#)

## 6.583.2 Member Function Documentation

### 6.583.2.1 setLastError()

```
void Digikam::FaceDbAccess::setLastError (
    const QString & error )
```

Set the "last error" message. This method is not for public use.

## 6.584 Digikam::FaceDbAccessUnlock Class Reference

### Public Member Functions

- [FaceDbAccessUnlock](#) ()
- [FaceDbAccessUnlock](#) ([FaceDbAccess](#) \*const access)

## 6.584.1 Constructor & Destructor Documentation

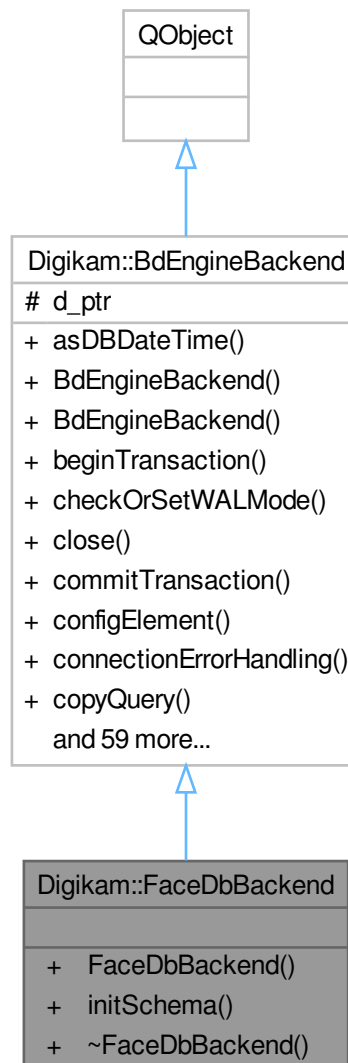
### 6.584.1.1 FaceDbAccessUnlock()

```
Digikam::FaceDbAccessUnlock::FaceDbAccessUnlock ( )
```

Acquire an object of this class if you want to assure that the [FaceDbAccess](#) is *not* held during the lifetime of the object. At creation, the lock is obtained shortly, then all locks are released. At destruction, all locks are acquired again. If you need to access any locked structures during lifetime, acquire a new [FaceDbAccess](#).

## 6.585 Digikam::FaceDbBackend Class Reference

Inheritance diagram for Digikam::FaceDbBackend:



## Public Member Functions

- **FaceDbBackend** ([DbEngineLocking](#) \*const locking, const [QString](#) &backendName=[QLatin1String](#)("face← Database-"))
- bool [initSchema](#) ([FaceDbSchemaUpdater](#) \*const updater)

## Public Member Functions inherited from [Digikam::BdEngineBackend](#)

- [QDateTime](#) [asDBDateTime](#) (const [QDateTime](#) &dateTime) const
- [BdEngineBackend](#) (const [QString](#) &backendName, [DbEngineLocking](#) \*const locking)
- **BdEngineBackend** (const [QString](#) &backendName, [DbEngineLocking](#) \*const locking, [BdEngineBackendPrivate](#) &dd)
- [BdEngineBackend::QueryState](#) [beginTransaction](#) ()
- bool [checkOrSetWALMode](#) ()
- void [close](#) ()
- [BdEngineBackend::QueryState](#) [commitTransaction](#) ()
- [DbEngineConfigSettings](#) [configElement](#) () const
- bool [connectionErrorHandling](#) (int retries)
- [DbEngineSqlQuery](#) [copyQuery](#) (const [DbEngineSqlQuery](#) &old)
- [DbType](#) [databaseType](#) () const
- bool [exec](#) ([DbEngineSqlQuery](#) &query)
- bool [execBatch](#) ([DbEngineSqlQuery](#) &query)
- [QueryState](#) [execDBAction](#) (const [DbEngineAction](#) &action, const [QMap](#)< [QString](#), [QVariant](#) > &bindingMap, [QList](#)< [QVariant](#) > \*const values=nullptr, [QVariant](#) \*const lastInsertId=nullptr)
- [QueryState](#) [execDBAction](#) (const [DbEngineAction](#) &action, [QList](#)< [QVariant](#) > \*const values=nullptr, [QVariant](#) \*const lastInsertId=nullptr)
- [QueryState](#) [execDBAction](#) (const [QString](#) &action, const [QMap](#)< [QString](#), [QVariant](#) > &bindingMap, [QList](#)< [QVariant](#) > \*const values=nullptr, [QVariant](#) \*const lastInsertId=nullptr)
- [QueryState](#) [execDBAction](#) (const [QString](#) &action, [QList](#)< [QVariant](#) > \*const values=nullptr, [QVariant](#) \*const lastInsertId=nullptr)
- [QStringQuery](#) [execDBActionQuery](#) (const [DbEngineAction](#) &action, const [QMap](#)< [QString](#), [QVariant](#) > &bindingMap)
- [QStringQuery](#) [execDBActionQuery](#) (const [QString](#) &action, const [QMap](#)< [QString](#), [QVariant](#) > &bindingMap)
- [QueryState](#) [execDirectSql](#) (const [QString](#) &query)
- [QueryState](#) [execDirectSqlWithResult](#) (const [QString](#) &query, [QList](#)< [QVariant](#) > \*const values=nullptr, [QVariant](#) \*const lastInsertId=nullptr)
- [DbEngineSqlQuery](#) [execQuery](#) (const [QString](#) &sql)
- [DbEngineSqlQuery](#) [execQuery](#) (const [QString](#) &sql, const [QList](#)< [QVariant](#) > &boundValues)
- [DbEngineSqlQuery](#) [execQuery](#) (const [QString](#) &sql, const [QMap](#)< [QString](#), [QVariant](#) > &bindingMap)
- [DbEngineSqlQuery](#) [execQuery](#) (const [QString](#) &sql, const [QVariant](#) &boundValue1)
- [DbEngineSqlQuery](#) [execQuery](#) (const [QString](#) &sql, const [QVariant](#) &boundValue1, const [QVariant](#) &boundValue2)
- [DbEngineSqlQuery](#) [execQuery](#) (const [QString](#) &sql, const [QVariant](#) &boundValue1, const [QVariant](#) &boundValue2, const [QVariant](#) &boundValue3)
- [DbEngineSqlQuery](#) [execQuery](#) (const [QString](#) &sql, const [QVariant](#) &boundValue1, const [QVariant](#) &boundValue2, const [QVariant](#) &boundValue3, const [QVariant](#) &boundValue4)
- void [execQuery](#) ([DbEngineSqlQuery](#) &preparedQuery, const [QList](#)< [QVariant](#) > &boundValues)
- void [execQuery](#) ([DbEngineSqlQuery](#) &preparedQuery, const [QVariant](#) &boundValue1)
- void [execQuery](#) ([DbEngineSqlQuery](#) &preparedQuery, const [QVariant](#) &boundValue1, const [QVariant](#) &boundValue2)
- void [execQuery](#) ([DbEngineSqlQuery](#) &preparedQuery, const [QVariant](#) &boundValue1, const [QVariant](#) &boundValue2, const [QVariant](#) &boundValue3)
- void [execQuery](#) ([DbEngineSqlQuery](#) &preparedQuery, const [QVariant](#) &boundValue1, const [QVariant](#) &boundValue2, const [QVariant](#) &boundValue3, const [QVariant](#) &boundValue4)

- [QueryState execSql](#) (const QString &sql, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) (const QString &sql, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) ([DbEngineSqlQuery](#) &preparedQuery, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execSql](#) ([DbEngineSqlQuery](#) &preparedQuery, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState execUpsertDBAction](#) (const [DbEngineAction](#) &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [QueryState execUpsertDBAction](#) (const QString &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [DbEngineAction getDBAction](#) (const QString &actionName) const
- [DbEngineSqlQuery getQuery](#) ()
- [QueryState handleQueryResult](#) ([DbEngineSqlQuery](#) &query, QList< QVariant > \*const values, QVariant \*const lastInsertId)
- bool [isCompatible](#) (const [DbEngineParameters](#) &parameters)
- bool [isInTransaction](#) () const
- bool [isOpen](#) () const
- bool [isReady](#) () const
- QString [lastError](#) ()
- QSqlError [lastSQLError](#) ()
- int [maximumBoundValues](#) () const
- bool [open](#) (const [DbEngineParameters](#) &parameters)
- [DbEngineSqlQuery prepareQuery](#) (const QString &sql)
- bool [queryErrorHandling](#) ([DbEngineSqlQuery](#) &query, int retries)
- QList< QVariant > [readToList](#) ([DbEngineSqlQuery](#) &query)
- void [rollbackTransaction](#) ()
- void [setDbEngineErrorHandler](#) ([DbEngineErrorHandler](#) \*const handler)
- void [setForeignKeyChecks](#) (bool check)
- [Status status](#) () const
- QStringList [tables](#) ()
- bool [transactionErrorHandling](#) (const QSqlError &[lastError](#), int retries)

## Additional Inherited Members

### Public Types inherited from [Digikam::BdEngineBackend](#)

- enum [DbType](#) { [SQLite](#) , [MySQL](#) }
- enum [QueryOperationStatus](#) { [ExecuteNormal](#) , [Wait](#) , [AbortQueries](#) }
- enum [QueryStateEnum](#) { [NoErrors](#) , [SQLException](#) , [ConnectionError](#) }
- enum [Status](#) { [Unavailable](#) , [Open](#) , [OpenSchemaChecked](#) }

### Protected Attributes inherited from [Digikam::BdEngineBackend](#)

- [BdEngineBackendPrivate](#) \*const [d\\_ptr](#) = nullptr

## 6.585.1 Member Function Documentation

### 6.585.1.1 [initSchema\(\)](#)

```
bool Digikam::FaceDbBackend::initSchema (
    FaceDbSchemaUpdater *const updater )
```

Initialize the database schema to the current version, carry out upgrades if necessary. Shall only be called from the thread that called [open\(\)](#).

## 6.586 Digikam::FaceDbOperationGroup Class Reference

### Public Member Functions

- void [allowLift](#) ()
- [FaceDbOperationGroup](#) ()
- [FaceDbOperationGroup](#) ([FaceDbAccess](#) \*const dbAccess)
- void [lift](#) ()
- void [resetTime](#) ()
- void [setMaximumTime](#) (int msec)

### 6.586.1 Detailed Description

When you intend to execute a number of write operations to the database, group them while holding a [FaceDbOperationGroup](#). For some database systems (SQLite), keeping a transaction across write operations occurring in short time results in enormous speedup (800x). For system that do not need this optimization, this class is a no-op.

### 6.586.2 Constructor & Destructor Documentation

#### 6.586.2.1 [FaceDbOperationGroup\(\)](#) [1/2]

```
Digikam::FaceDbOperationGroup::FaceDbOperationGroup ( )
```

Retrieve a [FaceDbAccess](#) object each time when constructing and destructing.

### 6.586.2.2 FaceDbOperationGroup() [2/2]

```
Digikam::FaceDbOperationGroup::FaceDbOperationGroup (
    FaceDbAccess *const dbAccess ) [explicit]
```

Use an existing [FaceDbAccess](#) object, which must live as long as this object exists.

## 6.586.3 Member Function Documentation

### 6.586.3.1 allowLift()

```
void Digikam::FaceDbOperationGroup::allowLift ( )
```

Allows to [lift\(\)](#). The transaction will be lifted if the time set by [setMaximumTime\(\)](#) has expired.

### 6.586.3.2 lift()

```
void Digikam::FaceDbOperationGroup::lift ( )
```

This will - if a transaction is held - commit the transaction and acquire a new one. This may improve concurrent access.

### 6.586.3.3 resetTime()

```
void Digikam::FaceDbOperationGroup::resetTime ( )
```

Resets to 0 the time used by [allowLift\(\)](#).

## 6.587 Digikam::FaceDbSchemaUpdater Class Reference

### Public Member Functions

- **FaceDbSchemaUpdater** ([FaceDbAccess](#) \*const dbAccess)
- void **setObserver** ([InitializationObserver](#) \*const observer)
- bool **update** ()

### Static Public Member Functions

- static int **schemaVersion** ()



## 6.588 Digikam::FaceDetector Class Reference

### Public Member Functions

- QString **backendIdentifier** () const
- QList< QRectF > **detectFaces** (const [DImg](#) &image, const QSize &originalSize=QSize())
- QList< QRectF > **detectFaces** (const QImage &image, const QSize &originalSize=QSize())
- QList< QRectF > **detectFaces** (const QString &imagePath)
- [FaceDetector](#) ()
- **FaceDetector** (const [FaceDetector](#) &other)
- [FaceDetector](#) & **operator=** (const [FaceDetector](#) &other)
- QVariantMap **parameters** () const
- int **recommendedImageSize** (const QSize &availableSize=QSize()) const
- void **setParameter** (const QString &parameter, const QVariant &value)
- void **setParameters** (const QVariantMap &parameters)

### Static Public Member Functions

- static QRect **toAbsoluteRect** (const QRectF &relativeRect, const QSize &size)
- static QList< QRect > **toAbsoluteRects** (const QList< QRectF > &relativeRects, const QSize &size)
- static QRectF **toRelativeRect** (const QRect &absoluteRect, const QSize &size)
- static QList< QRectF > **toRelativeRects** (const QList< QRect > &absoluteRects, const QSize &size)

## 6.588.1 Constructor & Destructor Documentation

### 6.588.1.1 FaceDetector()

```
Digikam::FaceDetector::FaceDetector ( )
```

Provides face detection, that means the process of selecting those regions of a full image which contain face.

This class provides shallow copying The class is fully reentrant (a single object and its copies are not thread-safe). Deferred creation is guaranteed, that means creation of a [FaceDetector](#) object is cheap, the expensive creation of the detection backend is performed when `detectFaces` is called for the first time.

## 6.588.2 Member Function Documentation

### 6.588.2.1 detectFaces() [1/2]

```
QList< QRectF > Digikam::FaceDetector::detectFaces (
    const DImg & image,
    const QSize & originalSize = QSize() )
```

Scan an image for faces. Return a list with regions possibly containing faces. If the image has been downscaled anywhere in the process, provide the original size of the image as this may be of importance in the detection process.

Found faces are returned in relative coordinates.

### 6.588.2.2 detectFaces() [2/2]

```
QList< QRectF > Digikam::FaceDetector::detectFaces (
    const QImage & image,
    const QSize & originalSize = QSize() )
```

Scan an image for faces. Return a list with regions possibly containing faces. If the image has been downscaled anywhere in the process, provide the original size of the image as this may be of importance in the detection process.

Found faces are returned in relative coordinates.

### 6.588.2.3 recommendedImageSize()

```
int Digikam::FaceDetector::recommendedImageSize (
    const QSize & availableSize = QSize() ) const
```

Returns the recommended size if you want to scale images for detection. Larger images can be passed, but may be downscaled.

### 6.588.2.4 setParameter()

```
void Digikam::FaceDetector::setParameter (
    const QString & parameter,
    const QVariant & value )
```

Tunes backend parameters. Available parameters:

"speed" vs. "accuracy", 0..1, float "sensitivity" vs. "specificity", 0..1, float.

For both pairs: a = 1-b, you can set either. The first pair changes the ROC curve in a trade for computing time. The second pair moves on a given ROC curve towards more false positives, or more missed faces.

## 6.589 Digikam::FaceGroup Class Reference

Inheritance diagram for Digikam::FaceGroup:



### Classes

- class [Private](#)

## Public Slots

- void [aboutToSetInfo](#) (const [ItemInfo](#) &info)
- void [addFace](#) ()
- void [markAllAsIgnored](#) ()
- void [rejectAll](#) ()
- void [setInfo](#) (const [ItemInfo](#) &info)
- void [setVisible](#) (bool visible)
- void [setVisibleItem](#) ([RegionFrameItem](#) \*item)

## Public Member Functions

- bool [acceptsMouseClicked](#) (const QPointF &scenePos)
- bool [autoSuggest](#) () const
- [RegionFrameItem](#) \* [closestItem](#) (const QPointF &p, qreal \*const manhattanLength=nullptr) const
- void [enterEvent](#) (QEvent \*)
- [FaceGroup](#) ([GraphicsDImgView](#) \*const view)
- bool [hasUnconfirmed](#) ()
- bool [hasVisibleItems](#) () const
- [ItemInfo](#) [info](#) () const
- bool [isVisible](#) () const
- void [itemHoverEnterEvent](#) (QGraphicsSceneHoverEvent \*event)
- void [itemHoverLeaveEvent](#) (QGraphicsSceneHoverEvent \*event)
- void [itemHoverMoveEvent](#) (QGraphicsSceneHoverEvent \*event)
- QList< [RegionFrameItem](#) \* > [items](#) () const
- void [leaveEvent](#) (QEvent \*)
- void [setAutoSuggest](#) (bool doAutoSuggest)
- void [setShowOnHover](#) (bool show)
- bool [showOnHover](#) () const

## Protected Slots

- void [itemStateChanged](#) (int)
- void [slotAddItemFinished](#) (const QRectF &rect)
- void [slotAddItemMoving](#) (const QRectF &rect)
- void [slotAddItemStarted](#) (const QPointF &pos)
- void [slotAlbumRenamed](#) ([Album](#) \*album)
- void [slotAlbumsUpdated](#) (int type)
- void [slotAssigned](#) (const [TaggingAction](#) &action, const [ItemInfo](#) &info, const QVariant &faceIdentifier)
- void [slotCancelAddItem](#) ()
- void [slotFocusRandomFace](#) ()
- void [slotIgnored](#) (const [ItemInfo](#) &info, const QVariant &faceIdentifier)
- void [slotIgnoredClicked](#) (const [ItemInfo](#) &info, const QVariant &faceIdentifier)
- void [slotLabelClicked](#) (const [ItemInfo](#) &info, const QVariant &faceIdentifier)
- void [slotRejected](#) (const [ItemInfo](#) &info, const QVariant &faceIdentifier)
- void [startAutoSuggest](#) ()

## Protected Member Functions

- void [applyItemGeometryChanges](#) ()
- void [clear](#) ()
- void [load](#) ()

## Properties

- bool **visible**

## 6.589.1 Constructor & Destructor Documentation

### 6.589.1.1 FaceGroup()

```
Digikam::FaceGroup::FaceGroup (  
    GraphicsDImgView *const view ) [explicit]
```

Constructs a new face group, managing RegionFrameItems for faces of a particular image, displayed on a [GraphicsDImgView](#).

## 6.589.2 Member Function Documentation

### 6.589.2.1 aboutToSetInfo

```
void Digikam::FaceGroup::aboutToSetInfo (  
    const ItemInfo & info ) [slot]
```

Prepares to load a new info. Closes the face group for editing. Pass a null info if about to close.

### 6.589.2.2 addFace

```
void Digikam::FaceGroup::addFace ( ) [slot]
```

Enters a special state where by click + drag a new face can be created.

### 6.589.2.3 closestItem()

```
RegionFrameItem * Digikam::FaceGroup::closestItem (  
    const QPointF & p,  
    qreal *const manhattanLength = nullptr ) const
```

Returns the item in this group closest to scene position p. If given, manhattanLength is set to the manhattan length between p and the closest point of the returned item's bounding rectangle. In particular, if p is inside the item's rectangle, manhattanLength is 0.

### 6.589.2.4 hasUnconfirmed()

```
bool Digikam::FaceGroup::hasUnconfirmed ( )
```

Returns a boolean whether there is at least one unconfirmed face in the group or not.

### 6.589.2.5 markAllAsIgnored

```
void Digikam::FaceGroup::markAllAsIgnored ( ) [slot]
```

Mark all unconfirmed faces as ignored on the current image.

### 6.589.2.6 rejectAll

```
void Digikam::FaceGroup::rejectAll ( ) [slot]
```

Rejects (clears) all faces on the current image.

### 6.589.2.7 setAutoSuggest()

```
void Digikam::FaceGroup::setAutoSuggest (
    bool doAutoSuggest )
```

Auto suggest applies if an image has not been scanned, or an unknown face is registered. In this case, a new scan will be triggered.

### 6.589.2.8 setInfo

```
void Digikam::FaceGroup::setInfo (
    const ItemInfo & info ) [slot]
```

Sets the current [ItemInfo](#).

### 6.589.2.9 setShowOnHover()

```
void Digikam::FaceGroup::setShowOnHover (
    bool show )
```

Even if `visible()` is false, show the item under the mouse cursor.

### 6.589.2.10 setVisible

```
void Digikam::FaceGroup::setVisible (
    bool visible ) [slot]
```

Shows or hides the frames.

## 6.590 Digikam::FaceGroup::Private Class Reference

### Public Member Functions

- [Faceltem](#) \* **addItem** (const [FaceTagsIface](#) &face)
- void **applyVisible** ()
- AssignNameWidget::Mode **assignWidgetMode** (FaceTagsIface::Type type)
- void **checkModels** ()
- [AssignNameWidget](#) \* **createAssignNameWidget** (const [FaceTagsIface](#) &face, const QVariant &identifier)
- [Faceltem](#) \* **createItem** (const [FaceTagsIface](#) &face)
- QList< [QGraphicsItem](#) \* > **hotItems** (const QPointF &scenePos)
- **Private** ([FaceGroup](#) \*const qq)

### Public Attributes

- bool **autoSuggest** = false
- bool **exifRotate** = true
- [TagPropertiesFilterModel](#) \* **filteredModel** = nullptr
- [CheckableAlbumFilterModel](#) \* **filterModel** = nullptr
- [ItemInfo](#) **info**
- QList< [Faceltem](#) \* > **items**
- [Faceltem](#) \* **manuallyAddedItem** = nullptr
- [ClickDragReleaseItem](#) \* **manuallyAddWrapItem** = nullptr
- const int [MaxFaceListSize](#) = 5
- const qreal [MaxMouseDistance](#) = 25.0
- [FacePipelineEdit](#) \* **newEditPipeline** = nullptr
- [FaceGroup](#) \*const **q** = nullptr
- bool **showOnHover** = false
- FaceGroupState **state** = NoFaces
- [TagModel](#) \* **tagModel** = nullptr
- [GraphicsDlmgView](#) \* **view** = nullptr
- [ItemVisibilityController](#) \* **visibilityController** = nullptr

### 6.590.1 Member Data Documentation

#### 6.590.1.1 MaxFaceListSize

```
const int Digikam::FaceGroup::Private::MaxFaceListSize = 5
```

Maximum size of faces in [FaceTagsIface](#) container while processing before to clean up and create again.

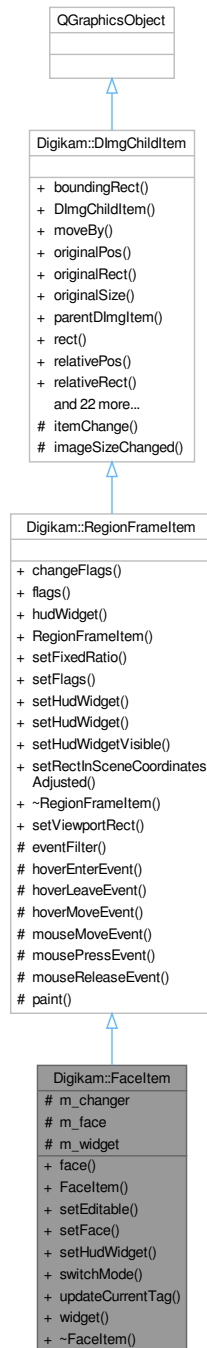
#### 6.590.1.2 MaxMouseDistance

```
const qreal Digikam::FaceGroup::Private::MaxMouseDistance = 25.0
```

The maximum distance of the mouse coordinate, when a face overlay widget is shown or hidden in the image, or whether the input widget moves up or down.

## 6.591 Digikam::Faceltem Class Reference

Inheritance diagram for Digikam::Faceltem:



### Public Member Functions

- [FaceTagsIface](#) **face** () const
- **Faceltem** (QGraphicsItem \*const parent)



- void **setEditable** (bool allowEdit)
- void **setFace** (const [FaceTagsIface](#) &face)
- void **setHudWidget** ([AssignNameWidget](#) \*const widget)
- void **switchMode** ([AssignNameWidget::Mode](#) mode)
- void **updateCurrentTag** ()
- [AssignNameWidget](#) \* **widget** () const

### Public Member Functions inherited from [Digikam::RegionFrameItem](#)

- void **changeFlags** (Flags flags, bool addOrRemove)
- Flags **flags** () const
- [QGraphicsWidget](#) \* **hudWidget** () const
- [RegionFrameItem](#) ([QGraphicsItem](#) \*const parent)
- void **setFixedRatio** (double ratio)
- void **setFlags** (Flags flags)
- void **setHudWidget** ([QGraphicsWidget](#) \*const hudWidget)
- void **setHudWidget** ([QWidget](#) \*const widget, [Qt::WindowFlags](#) wFlags=[Qt::WindowFlags](#)())
- void **setHudWidgetVisible** (bool visible)
- void **setRectInSceneCoordinatesAdjusted** (const [QRectF](#) &[rect](#))

### Public Member Functions inherited from [Digikam::DImgChildItem](#)

- [QRectF](#) **boundingRect** () const override
- [DImgChildItem](#) ([QGraphicsItem](#) \*const parent=nullptr)
- void **moveBy** (qreal dx, qreal dy)
- [QPoint](#) **originalPos** () const
- [QRect](#) **originalRect** () const
- [QSize](#) **originalSize** () const
- [GraphicsDImgItem](#) \* **parentDImgItem** () const
- [QRectF](#) **rect** () const
- [QPointF](#) **relativePos** () const
- [QRectF](#) **relativeRect** () const
- [QSizeF](#) **relativeSize** () const
- void **setOriginalPos** (const [QPointF](#) &posInOriginal)
- void **setOriginalPos** (qreal x, qreal y)
- void **setOriginalRect** (const [QRectF](#) &[rect](#))
- void **setOriginalRect** (qreal x, qreal y, qreal width, qreal height)
- void **setOriginalSize** (const [QSizeF](#) &sizeInOriginal)
- void **setOriginalSize** (qreal width, qreal height)
- void **setPos** (const [QPointF](#) &zoomedPos)
- void **setPos** (qreal x, qreal y)
- void **setRect** (const [QRectF](#) &[rect](#))
- void **setRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRectInSceneCoordinates** (const [QRectF](#) &[rect](#))
- void **setRelativePos** (const [QPointF](#) &relativePosition)
- void **setRelativePos** (qreal x, qreal y)
- void **setRelativeRect** (const [QRectF](#) &[rect](#))
- void **setRelativeRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRelativeSize** (const [QSizeF](#) &relativeSize)
- void **setRelativeSize** (qreal width, qreal height)
- void **setSize** (const [QSizeF](#) &zoomedSize)
- void **setSize** (qreal width, qreal height)
- [QSizeF](#) **size** () const

**Protected Attributes**

- [HidingStateChanger](#) \* **m\_changer** = nullptr
- [FaceTagsIface](#) **m\_face**
- [AssignNameWidget](#) \* **m\_widget** = nullptr

**Additional Inherited Members****Public Types inherited from [Digikam::RegionFrameItem](#)**

- enum **Flag** { **NoFlags** = 0 , **ShowResizeHandles** = 1 << 0 , **MoveByDrag** = 1 << 1 , **GeometryEditable** = ShowResizeHandles | MoveByDrag }

**Public Slots inherited from [Digikam::RegionFrameItem](#)**

- void [setViewportRect](#) (const QRectF &rect)

**Signals inherited from [Digikam::RegionFrameItem](#)**

- void [geometryEdited](#) ()

**Signals inherited from [Digikam::DImgChildItem](#)**

- void [geometryChanged](#) ()
- void [geometryOnImageChanged](#) ()
- void [positionChanged](#) ()
- void [positionOnImageChanged](#) ()
- void [sizeChanged](#) ()
- void [sizeOnImageChanged](#) ()

**Protected Slots inherited from [Digikam::DImgChildItem](#)**

- void [imageSizeChanged](#) (const QSizeF &)

**Protected Member Functions inherited from [Digikam::RegionFrameItem](#)**

- bool [eventFilter](#) (QObject \*watched, QEvent \*event) override
- void [hoverEnterEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [hoverLeaveEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [hoverMoveEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [mouseMoveEvent](#) (QGraphicsSceneMouseEvent \*) override
- void [mousePressEvent](#) (QGraphicsSceneMouseEvent \*) override
- void [mouseReleaseEvent](#) (QGraphicsSceneMouseEvent \*) override
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget=nullptr) override

**Protected Member Functions inherited from [Digikam::DImgChildItem](#)**

- QVariant [itemChange](#) (GraphicsItemChange change, const QVariant &value) override

## 6.592 Digikam::FaceltemRetriever Class Reference

### Public Member Functions

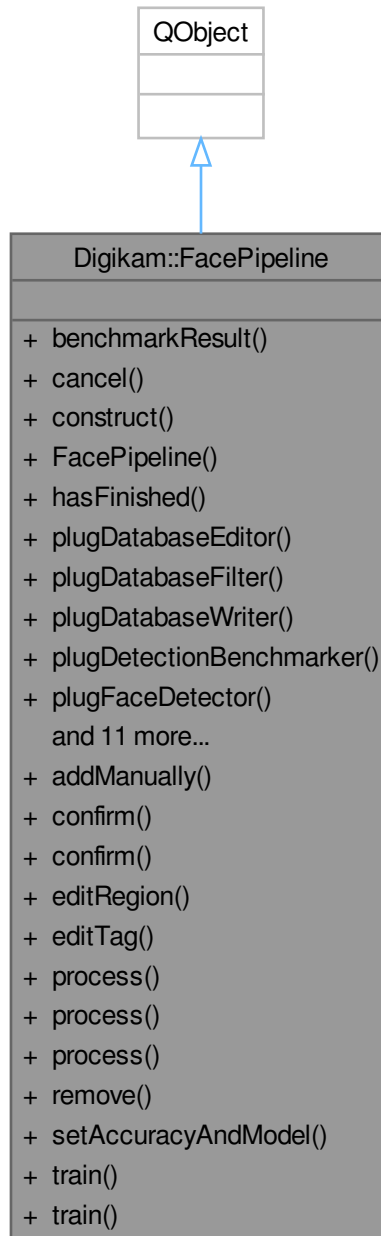
- void **cancel** ()
- **FaceltemRetriever** ([FacePipeline::Private](#) \*const d)
- [QList< QImage \\* >](#) **getDetails** (const [DImg](#) &src, const [QList< FaceTagsIface >](#) &faces) const
- [QList< QImage \\* >](#) **getDetails** (const [DImg](#) &src, const [QList< QRectF >](#) &rects) const
- [QList< QImage \\* >](#) **getThumbnails** (const [QString](#) &filePath, const [QList< FaceTagsIface >](#) &faces) const

### Protected Attributes

- [ThumbnailImageCatcher](#) \* **catcher** = nullptr

## 6.593 Digikam::FacePipeline Class Reference

Inheritance diagram for Digikam::FacePipeline:



### Classes

- class [Private](#)

## Public Types

- enum `FilterMode` { `ScanAll` , `SkipAlreadyScanned` , `ReadUnconfirmedFaces` , `ReadFacesForTraining` , `ReadConfirmedFaces` }
- enum `WriteMode` { `NormalWrite` , `OverwriteAllFaces` , `OverwriteUnconfirmed` }

## Public Slots

- `FaceTagsIface` `addManually` (const `ItemInfo` &info, const `DImage` &image, const `TagRegion` &assignedRegion)
- `FaceTagsIface` `confirm` (const `ItemInfo` &info, const `FaceTagsIface` &face, const `DImage` &image, int assignedTagId=0, const `TagRegion` &assignedRegion=`TagRegion`())
- `FaceTagsIface` `confirm` (const `ItemInfo` &info, const `FaceTagsIface` &face, int assignedTagId=0, const `TagRegion` &assignedRegion=`TagRegion`())
- `FaceTagsIface` `editRegion` (const `ItemInfo` &info, const `DImage` &image, const `FaceTagsIface` &databaseFace, const `TagRegion` &newRegion)
- `FaceTagsIface` `editTag` (const `ItemInfo` &info, const `FaceTagsIface` &databaseFace, int newTagId)
- bool `process` (const `ItemInfo` &info)
- bool `process` (const `ItemInfo` &info, const `DImage` &image)
- void `process` (const `QList`< `ItemInfo` > &infos)
- void `remove` (const `ItemInfo` &info, const `FaceTagsIface` &face)
- void `setAccuracyAndModel` (int detectAccuracy, `FaceScanSettings::FaceDetectionModel` detectModel, `FaceScanSettings::FaceDetectionSize` detectSize, int recognizeAccuracy, `FaceScanSettings::FaceRecognitionModel` recognizeModel)
- void `train` (const `ItemInfo` &info, const `QList`< `FaceTagsIface` > &faces)
- void `train` (const `ItemInfo` &info, const `QList`< `FaceTagsIface` > &faces, const `DImage` &image)

## Signals

- void `finished` ()  
*Emitted when the last package has finished processing.*
- void `processed` (const `FacePipelinePackage` &package)  
*Emitted when one package has finished processing.*
- void `processing` (const `FacePipelinePackage` &package)  
*Emitted when one package begins processing.*
- void `progressValueChanged` (float progress)
- void `scheduled` ()  
*Emitted when processing is scheduled.*
- void `skipped` (const `QList`< `ItemInfo` > &skippedInfos)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void `started` (const `QString` &message)  
*Emitted when processing has started.*

## Public Member Functions

- QString **benchmarkResult** () const
- void **cancel** ()
- void **construct** ()
- bool **hasFinished** () const
- void **plugDatabaseEditor** ()
- void **plugDatabaseFilter** ([FilterMode](#) mode)
- void **plugDatabaseWriter** ([WriteMode](#) mode)
- void **plugDetectionBenchmark** ()
- void **plugFaceDetector** ()
- void **plugFacePreviewLoader** ()
- void **plugFaceRecognizer** ()
- void **plugParallelFaceDetectors** ()
- void **plugRecognitionBenchmark** ()
- void **plugRerecognizingDatabaseFilter** ()
- void **plugRetrainingDatabaseFilter** ()
- void **plugTrainer** ()
- QThread::Priority **priority** () const
- void **setPriority** (QThread::Priority priority)
- void **shutDown** ()

## Friends

- class **Private**

## 6.593.1 Member Enumeration Documentation

### 6.593.1.1 FilterMode

enum [Digikam::FacePipeline::FilterMode](#)

#### Enumerator

ScanAll	Will read any given image.
SkipAlreadyScanned	Will skip any image that is already marked as scanned.
ReadUnconfirmedFaces	Will read unconfirmed faces for recognition.
ReadFacesForTraining	Will read faces marked for training.
ReadConfirmedFaces	Will read faces which are confirmed.

### 6.593.1.2 WriteMode

enum [Digikam::FacePipeline::WriteMode](#)

#### Enumerator

NormalWrite	Write results. Merge with existing entries.
OverwriteAllFaces	Add new results. Previous all results will be cleared.
OverwriteUnconfirmed	Add new results. Previous unconfirmed results will be cleared.

## 6.593.2 Member Function Documentation

### 6.593.2.1 addManually

```
FaceTagsIface Digikam::FacePipeline::addManually (
    const ItemInfo & info,
    const DImg & image,
    const TagRegion & assignedRegion ) [slot]
```

Add an entry manually.

### 6.593.2.2 cancel()

```
void Digikam::FacePipeline::cancel ( )
```

Cancels all processing.

### 6.593.2.3 confirm

```
FaceTagsIface Digikam::FacePipeline::confirm (
    const ItemInfo & info,
    const FaceTagsIface & face,
    int assignedTagId = 0,
    const TagRegion & assignedRegion = TagRegion() ) [slot]
```

Confirm the face. Pass the original face, and additionally tag id or region if they changed. Returns the confirmed face entry immediately purely for convenience, it is not yet in the database (connect to signal [processed\(\)](#) to react when the processing finished). If a trainer is plugged, the face will be trained.

### 6.593.2.4 editRegion

```
FaceTagsIface Digikam::FacePipeline::editRegion (
    const ItemInfo & info,
    const DImg & image,
    const FaceTagsIface & databaseFace,
    const TagRegion & newRegion ) [slot]
```

Change the given face's region to newRegion. Does not care for training atm.

### 6.593.2.5 editTag

```
FaceTagsIface Digikam::FacePipeline::editTag (
    const ItemInfo & info,
    const FaceTagsIface & databaseFace,
    int newTagId ) [slot]
```

Changes the given face's tagId to newTagId. Used to Reject Facial Recognition suggestions, since the tag needs to be converted from Unconfirmed to Unknown.

### 6.593.2.6 plugDatabaseFilter()

```
void Digikam::FacePipeline::plugDatabaseFilter (
    FilterMode mode )
```

You can plug these four different steps in the working pipeline. 1) Call any of the four plug...() methods. See below for supported combinations. 2) Call construct() to set up the pipeline.

- Database filter: Prepares database records and/or filters out items. See FilterMode for specification.
- Preview loader: If no preview loader is plugged, you must provide a [DImg](#) for face detection and recognition
- Face Detector: If no recognizer is plugged, all detected face are marked as the unknown person
- Face Recognizer: If no detector is plugged, only already scanned faces marked as unknown will be processed. They are implicitly read from the database.
- [DatabaseWriter](#): Writes the detection and recognition results to the database. The trainer works on a completely different storage and is not affected by the database writer.
- DatabaseEditor: Can confirm or reject faces

PlugParallel: You can call this instead of the simple plugging method. Depending on the number of processor cores of the machine and the memory cost, more than one element may be plugged and process parallelly for this part of the pipeline.

Supported combinations: (Database [Filter](#) ->) (Preview Loader ->) Detector -> Recognizer (-> [DatabaseWriter](#)) (Database [Filter](#) ->) (Preview Loader ->) Detector (-> [DatabaseWriter](#)) (Database [Filter](#) ->) (Preview Loader ->) Recognizer (-> [DatabaseWriter](#)) DatabaseEditor Trainer DatabaseEditor -> Trainer

### 6.593.2.7 process [1/2]

```
bool Digikam::FacePipeline::process (
    const ItemInfo & info ) [slot]
```

Processes the given image info. If a filter is installed, returns false if the info is skipped, or true if it is processed. If no preview loader is plugged, you must provide a [DImg](#) for detection or recognition. Any of the signals below will only be emitted if true is returned.

### 6.593.2.8 process [2/2]

```
void Digikam::FacePipeline::process (
    const QList< ItemInfo > & infos ) [slot]
```

Batch processing. If a filter is installed, the [skipped\(\)](#) signal will inform about skipped infos. Filtering is done in a thread, returns immediately. Some of the signals below will be emitted in any case.

### 6.593.2.9 remove

```
void Digikam::FacePipeline::remove (
    const ItemInfo & info,
    const FaceTagsIface & face ) [slot]
```

Remove the given face.



### 6.593.2.10 setPriority()

```
void Digikam::FacePipeline::setPriority (
    QThread::Priority priority )
```

Set the priority of the threads used by this pipeline. The default setting is QThread::LowPriority.

### 6.593.2.11 shutDown()

```
void Digikam::FacePipeline::shutDown ( )
```

Cancels and waits for the pipeline to finish.

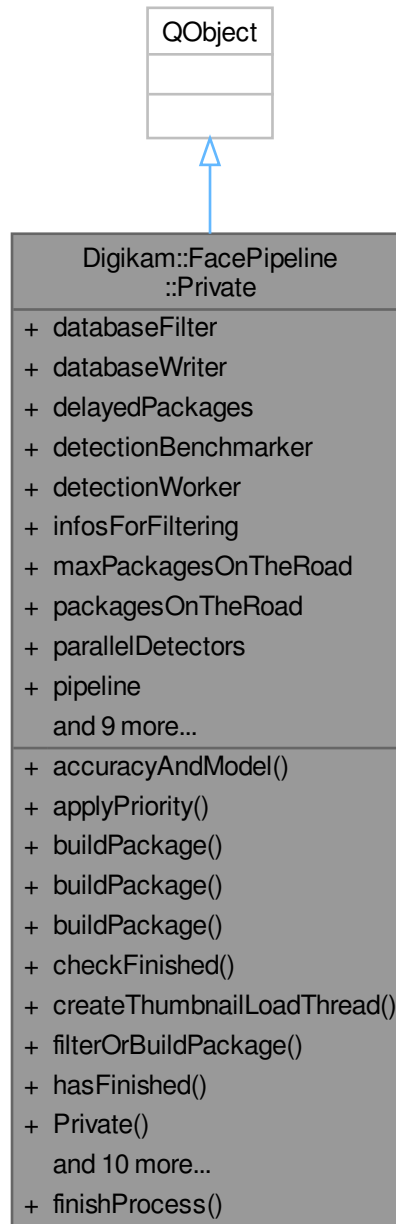
### 6.593.2.12 train

```
void Digikam::FacePipeline::train (
    const ItemInfo & info,
    const QList< FaceTagsIface > & faces ) [slot]
```

Train the given faces.

## 6.594 Digikam::FacePipeline::Private Class Reference

Inheritance diagram for Digikam::FacePipeline::Private:



### Public Slots

- void **finishProcess** (FacePipelineExtendedPackage::Ptr package)

## Public Member Functions

- void **accuracyAndModel** (int detectAccuracy, [FaceScanSettings::FaceDetectionModel](#) detectModel, [FaceScanSettings::FaceDetectionSize](#) detectSize, int recognizeAccuracy, [FaceScanSettings::FaceRecognitionModel](#) recognizeModel)
- void **applyPriority** ()
- [FacePipelineExtendedPackage::Ptr](#) **buildPackage** (const [ItemInfo](#) &info)
- [FacePipelineExtendedPackage::Ptr](#) **buildPackage** (const [ItemInfo](#) &info, const [FacePipelineFaceTagsIface](#) &, const [DImg](#) &image)
- [FacePipelineExtendedPackage::Ptr](#) **buildPackage** (const [ItemInfo](#) &info, const [FacePipelineFaceTagsIfaceList](#) &faces, const [DImg](#) &image)
- void **checkFinished** ()
- [ThumbnailLoadThread](#) \* **createThumbnailLoadThread** ()
- [FacePipelineExtendedPackage::Ptr](#) **filterOrBuildPackage** (const [ItemInfo](#) &info)
- bool **hasFinished** () const
- **Private** ([FacePipeline](#) \*const qq)
- void **processBatch** (const [QList](#)< [ItemInfo](#) > &infos)
- void **receiverFlowControl** ()
- void **send** (const [FacePipelineExtendedPackage::Ptr](#) &package)
- bool **senderFlowControl** (const [FacePipelineExtendedPackage::Ptr](#) &package)
- void **sendFromFilter** (const [QList](#)< [FacePipelineExtendedPackage::Ptr](#) > &packages)  
*called by filter.*
- void **skipFromFilter** (const [QList](#)< [ItemInfo](#) > &infosForSkipping)  
*called by filter.*
- void **start** ()
- void **startProcess** (const [FacePipelineExtendedPackage::Ptr](#) &package)
- void **stop** ()
- void **wait** ()

## Public Attributes

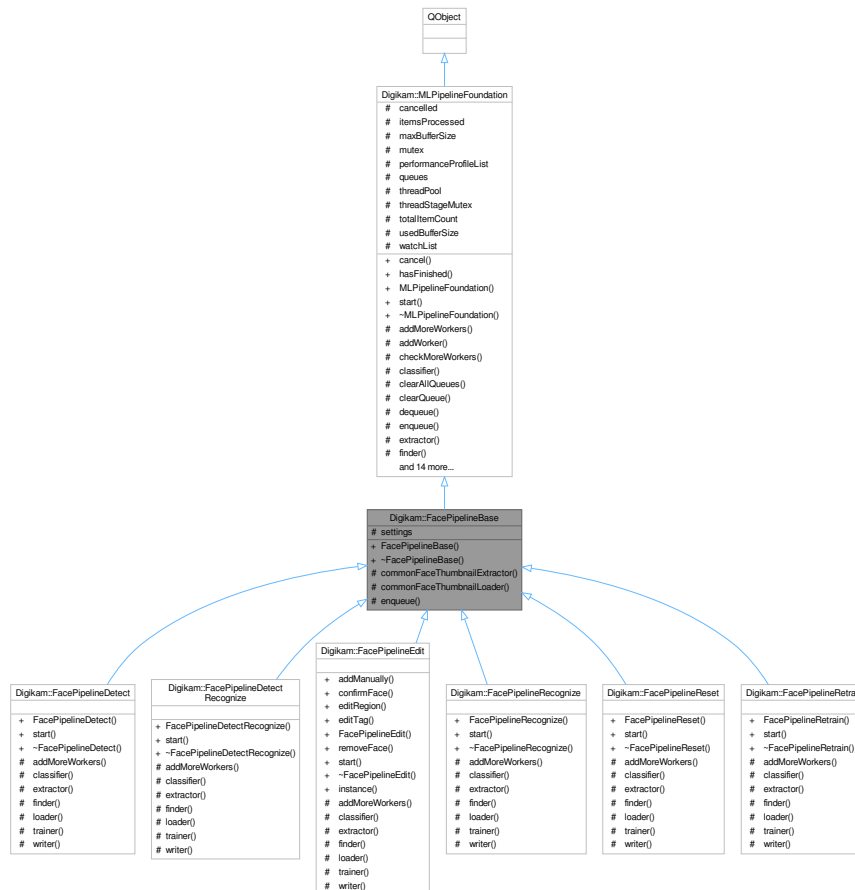
- [ScanStateFilter](#) \* **databaseFilter** = nullptr
- [DatabaseWriter](#) \* **databaseWriter** = nullptr
- [QList](#)< [FacePipelineExtendedPackage::Ptr](#) > **delayedPackages**
- [DetectionBenchmark](#) \* **detectionBenchmark** = nullptr
- [DetectionWorker](#) \* **detectionWorker** = nullptr
- int **infosForFiltering** = 0
- int **maxPackagesOnTheRoad** = 30
- int **packagesOnTheRoad** = 0
- [ParallelPipes](#) \* **parallelDetectors** = nullptr
- [QList](#)< [QObject](#) \* > **pipeline**
- [FacePreviewLoader](#) \* **previewThread** = nullptr
- [QThread::Priority](#) **priority** = [QThread::LowPriority](#)
- [RecognitionBenchmark](#) \* **recognitionBenchmark** = nullptr
- [RecognitionWorker](#) \* **recognitionWorker** = nullptr
- bool **started** = false
- [QList](#)< [ThumbnailLoadThread](#) \* > **thumbnailLoadThreads**
- int **totalPackagesAdded** = 0
- [TrainerWorker](#) \* **trainerWorker** = nullptr
- bool **waiting** = false

## Friends

- class [FacePipeline](#)

## 6.595 Digikam::FacePipelineBase Class Reference

Inheritance diagram for Digikam::FacePipelineBase:



### Public Types

- enum [FilterMode](#) { [ScanAll](#) , [ScanNew](#) , [TrainNew](#) , [TrainAll](#) , [TrainRemove](#) , [TrainReset](#) }
- enum [WriteMode](#) { [NormalWrite](#) , [OverwriteAllFaces](#) , [OverwriteUnconfirmed](#) }

### Public Types inherited from Digikam::MLPipelineFoundation

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue](#)< [MLPipelinePackageFoundation](#) \* > [MLPipelineQueue](#)
- enum [MLPipelineStage](#) { [Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) , [Trainer](#) , [Writer](#) , [None](#) }

## Public Member Functions

- **FacePipelineBase** (const [FaceScanSettings](#) &\_settings)

## Public Member Functions inherited from [Digikam::MLPipelineFoundation](#)

- virtual void [cancel](#) ()
- bool [hasFinished](#) () const
- virtual bool [start](#) ()

## Protected Member Functions

- bool [commonFaceThumbnailExtractor](#) (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool [commonFaceThumbnailLoader](#) (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool [enqueue](#) ([MLPipelineQueue](#) \*thisQueue, [MLPipelinePackageFoundation](#) \*package) override

## Protected Member Functions inherited from [Digikam::MLPipelineFoundation](#)

- virtual void [addMoreWorkers](#) ()=0
- bool [addWorker](#) (const [MLPipelineStage](#) &stage)
- bool [checkMoreWorkers](#) (int totalItemCount, int currentItemCount, bool useFullCpu)
- virtual bool [classifier](#) ()=0
- void [clearAllQueues](#) ()
- void [clearQueue](#) ([MLPipelineQueue](#) \*thisQueue)
- virtual [MLPipelinePackageFoundation](#) \* [dequeue](#) ([MLPipelineQueue](#) \*thisQueue)
- virtual bool [extractor](#) ()=0
- virtual bool [finder](#) ()=0
- virtual bool [loader](#) ()=0
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [DImg](#) &\_thumbnail)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QIcon](#) &\_thumbnail)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QImage](#) &\_thumbnail)
- void [pipelinePerformanceEnd](#) (const [MLPipelineStage](#) &stage, int totalItemCount, [QElapsedTimer](#) &timer)
- void [pipelinePerformanceEnd](#) (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- void [pipelinePerformanceStart](#) (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- [MLPipelinePackageFoundation](#) \* [queueEndSignal](#) () const
- void [showPipelinePerformance](#) () const
- void [stageEnd](#) ([MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage)
- void [stageStart](#) ([QThread::Priority](#) threadPriority, [MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage, [MLPipelineQueue](#) \*&thisQueue, [MLPipelineQueue](#) \*&nextQueue)
- virtual bool [trainer](#) ()=0
- void [waitForStart](#) ()
- virtual bool [writer](#) ()=0

## Protected Attributes

- [FaceScanSettings](#) [settings](#)

## Protected Attributes inherited from [Digikam::MLPipelineFoundation](#)

- bool **cancelled** = false
- QAtomicInteger< int > **itemsProcessed** = 0
- quint64 **maxBufferSize** = 2147483648  
*2 GB default*
- QMutex **mutex**
- QMap< [MLPipelineStage](#), [MLPipelinePerformanceProfile](#) > **performanceProfileList**
- QMap< [MLPipelineStage](#), [MLPipelineQueue](#) \* > **queues**
- QThreadPool \* **threadPool** = nullptr
- QMutex **threadStageMutex**
- QAtomicInteger< int > **totalItemCount** = 0
- quint64 **usedBufferSize** = 0
- QList< QFutureWatcher< bool > \* > **watchList**

## Additional Inherited Members

## Signals inherited from [Digikam::MLPipelineFoundation](#)

- void **finished** ()  
*Emitted when the last package has finished processing.*
- void **processed** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package has finished processing.*
- void **processing** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package begins processing.*
- void **progressValueChanged** (float progress)
- void **scheduled** ()  
*Emitted when processing is scheduled.*
- void **signalAddMoreWorkers** ()
- void **signalUpdateItemCount** (const qlonglong itemCount)
- void **skipped** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void **started** (const QString &message)  
*Emitted when processing has started.*

## 6.595.1 Member Enumeration Documentation

### 6.595.1.1 FilterMode

enum [Digikam::FacePipelineBase::FilterMode](#)

#### Enumerator

ScanAll	Will read any given image.
ScanNew	Scan new images, will skip any image that is already marked as scanned.
TrainNew	Adds new face(s) to training.
TrainAll	Retrains the face DB.
TrainRemove	Removes the face(s) from training.
TrainReset	Removes all face training, sets all images to not scanned.

### 6.595.1.2 WriteMode

```
enum Digikam::FacePipelineBase::WriteMode
```

#### Enumerator

NormalWrite	Write results. Merge with existing entries.
OverwriteAllFaces	Add new results. Previous all results will be cleared.
OverwriteUnconfirmed	Add new results. Previous unconfirmed results will be cleared.

## 6.595.2 Member Function Documentation

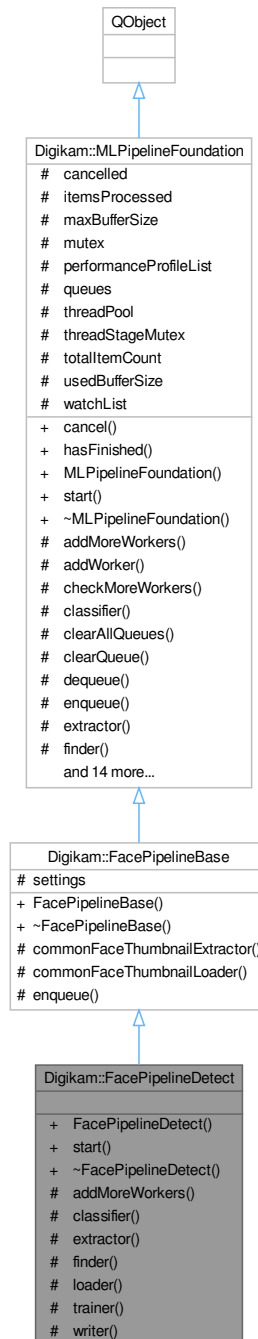
### 6.595.2.1 enqueue()

```
bool Digikam::FacePipelineBase::enqueue (  
    MLPipelineQueue * thisQueue,  
    MLPipelinePackageFoundation * package ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::MLPipelineFoundation](#).

## 6.596 Digikam::FacePipelineDetect Class Reference

Inheritance diagram for Digikam::FacePipelineDetect:



### Public Member Functions

- **FacePipelineDetect** (const [FaceScanSettings](#) &\_settings)
- bool [start](#) () override



## Public Member Functions inherited from Digikam::FacePipelineBase

- **FacePipelineBase** (const [FaceScanSettings](#) &\_settings)

## Public Member Functions inherited from Digikam::MLPipelineFoundation

- virtual void [cancel](#) ()
- bool [hasFinished](#) () const

## Protected Member Functions

- void [addMoreWorkers](#) () override
- bool [classifier](#) () override
- bool [extractor](#) () override
- bool [finder](#) () override
- bool [loader](#) () override
- bool [trainer](#) () override
- bool [writer](#) () override

## Protected Member Functions inherited from Digikam::FacePipelineBase

- bool [commonFaceThumbnailExtractor](#) (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool [commonFaceThumbnailLoader](#) (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool [enqueue](#) ([MLPipelineQueue](#) \*thisQueue, [MLPipelinePackageFoundation](#) \*package) override

## Protected Member Functions inherited from Digikam::MLPipelineFoundation

- bool [addWorker](#) (const [MLPipelineStage](#) &stage)
- bool [checkMoreWorkers](#) (int totalItemCount, int currentItemCount, bool useFullCpu)
- void [clearAllQueues](#) ()
- void [clearQueue](#) ([MLPipelineQueue](#) \*thisQueue)
- virtual [MLPipelinePackageFoundation](#) \* [dequeue](#) ([MLPipelineQueue](#) \*thisQueue)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [DImg](#) &\_thumbnail)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QIcon](#) &\_thumbnail)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QImage](#) &\_thumbnail)
- void [pipelinePerformanceEnd](#) (const [MLPipelineStage](#) &stage, int totalItemCount, [QElapsedTimer](#) &timer)
- void [pipelinePerformanceEnd](#) (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- void [pipelinePerformanceStart](#) (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- [MLPipelinePackageFoundation](#) \* [queueEndSignal](#) () const
- void [showPipelinePerformance](#) () const
- void [stageEnd](#) ([MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage)
- void [stageStart](#) ([QThread::Priority](#) threadPriority, [MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage, [MLPipelineQueue](#) \*&thisQueue, [MLPipelineQueue](#) \*&nextQueue)
- void [waitForStart](#) ()

## Additional Inherited Members

### Public Types inherited from [Digikam::FacePipelineBase](#)

- enum [FilterMode](#) { [ScanAll](#) , [ScanNew](#) , [TrainNew](#) , [TrainAll](#) , [TrainRemove](#) , [TrainReset](#) }
- enum [WriteMode](#) { [NormalWrite](#) , [OverwriteAllFaces](#) , [OverwriteUnconfirmed](#) }

### Public Types inherited from [Digikam::MLPipelineFoundation](#)

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue](#)< [MLPipelinePackageFoundation](#) \* > [MLPipelineQueue](#)
- enum [MLPipelineStage](#) { [Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) , [Trainer](#) , [Writer](#) , [None](#) }

### Signals inherited from [Digikam::MLPipelineFoundation](#)

- void [finished](#) ()  
*Emitted when the last package has finished processing.*
- void [processed](#) (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package has finished processing.*
- void [processing](#) (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package begins processing.*
- void [progressValueChanged](#) (float progress)
- void [scheduled](#) ()  
*Emitted when processing is scheduled.*
- void [signalAddMoreWorkers](#) ()
- void [signalUpdateItemCount](#) (const qlonglong itemCount)
- void [skipped](#) (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void [started](#) (const QString &message)  
*Emitted when processing has started.*

### Protected Attributes inherited from [Digikam::FacePipelineBase](#)

- [FaceScanSettings](#) [settings](#)

### Protected Attributes inherited from [Digikam::MLPipelineFoundation](#)

- bool [cancelled](#) = false
- [QAtomicInteger](#)< int > [itemsProcessed](#) = 0
- quint64 [maxBufferSize](#) = 2147483648  
*2 GB default*
- [QMutex](#) [mutex](#)
- [QMap](#)< [MLPipelineStage](#), [MLPipelinePerformanceProfile](#) > [performanceProfileList](#)
- [QMap](#)< [MLPipelineStage](#), [MLPipelineQueue](#) \* > [queues](#)
- [QThreadPool](#) \* [threadPool](#) = nullptr
- [QMutex](#) [threadStageMutex](#)
- [QAtomicInteger](#)< int > [totalItemCount](#) = 0
- quint64 [usedBufferSize](#) = 0
- [QList](#)< [QFutureWatcher](#)< bool > \* > [watchList](#)

## 6.596.1 Member Function Documentation

### 6.596.1.1 addMoreWorkers()

```
void Digikam::FacePipelineDetect::addMoreWorkers ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.596.1.2 classifier()

```
bool Digikam::FacePipelineDetect::classifier ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.596.1.3 extractor()

```
bool Digikam::FacePipelineDetect::extractor ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.596.1.4 finder()

```
bool Digikam::FacePipelineDetect::finder ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.596.1.5 loader()

```
bool Digikam::FacePipelineDetect::loader ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.596.1.6 start()

```
bool Digikam::FacePipelineDetect::start ( ) [override], [virtual]
```

Reimplemented from [Digikam::MLPipelineFoundation](#).

### 6.596.1.7 trainer()

```
bool Digikam::FacePipelineDetect::trainer ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.596.1.8 writer()

```
bool Digikam::FacePipelineDetect::writer ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

## 6.597 Digikam::FacePipelineDetectRecognize Class Reference

Inheritance diagram for Digikam::FacePipelineDetectRecognize:



**Public Member Functions**

- **FacePipelineDetectRecognize** (const [FaceScanSettings](#) &\_settings)
- bool **start** () override

**Public Member Functions inherited from [Digikam::FacePipelineBase](#)**

- **FacePipelineBase** (const [FaceScanSettings](#) &\_settings)

**Public Member Functions inherited from [Digikam::MLPipelineFoundation](#)**

- virtual void **cancel** ()
- bool **hasFinished** () const

**Protected Member Functions**

- void **addMoreWorkers** () override
- bool **classifier** () override
- bool **extractor** () override
- bool **finder** () override
- bool **loader** () override
- bool **trainer** () override
- bool **writer** () override

**Protected Member Functions inherited from [Digikam::FacePipelineBase](#)**

- bool **commonFaceThumbnailExtractor** (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **commonFaceThumbnailLoader** (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **enqueue** ([MLPipelineQueue](#) \*thisQueue, [MLPipelinePackageFoundation](#) \*package) override

**Protected Member Functions inherited from [Digikam::MLPipelineFoundation](#)**

- bool **addWorker** (const [MLPipelineStage](#) &stage)
- bool **checkMoreWorkers** (int totalItemCount, int currentItemCount, bool useFullCpu)
- void **clearAllQueues** ()
- void **clearQueue** ([MLPipelineQueue](#) \*thisQueue)
- virtual [MLPipelinePackageFoundation](#) \* **dequeue** ([MLPipelineQueue](#) \*thisQueue)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [DImg](#) &\_thumbnail)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QIcon](#) &\_thumbnail)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QImage](#) &\_thumbnail)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, int totalItemCount, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceStart** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- [MLPipelinePackageFoundation](#) \* **queueEndSignal** () const
- void **showPipelinePerformance** () const
- void **stageEnd** ([MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage)
- void **stageStart** ([QThread::Priority](#) threadPriority, [MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage, [MLPipelineQueue](#) \*&thisQueue, [MLPipelineQueue](#) \*&nextQueue)
- void **waitForStart** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::FacePipelineBase](#)

- enum [FilterMode](#) { [ScanAll](#) , [ScanNew](#) , [TrainNew](#) , [TrainAll](#) , [TrainRemove](#) , [TrainReset](#) }
- enum [WriteMode](#) { [NormalWrite](#) , [OverwriteAllFaces](#) , [OverwriteUnconfirmed](#) }

### Public Types inherited from [Digikam::MLPipelineFoundation](#)

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue](#)< [MLPipelinePackageFoundation](#) \* > [MLPipelineQueue](#)
- enum [MLPipelineStage](#) { [Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) , [Trainer](#) , [Writer](#) , [None](#) }

### Signals inherited from [Digikam::MLPipelineFoundation](#)

- void **finished** ()  
*Emitted when the last package has finished processing.*
- void **processed** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package has finished processing.*
- void **processing** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package begins processing.*
- void **progressValueChanged** (float progress)
- void **scheduled** ()  
*Emitted when processing is scheduled.*
- void **signalAddMoreWorkers** ()
- void **signalUpdateItemCount** (const qlonglong itemCount)
- void **skipped** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void **started** (const [QString](#) &message)  
*Emitted when processing has started.*

### Protected Attributes inherited from [Digikam::FacePipelineBase](#)

- [FaceScanSettings](#) **settings**

### Protected Attributes inherited from [Digikam::MLPipelineFoundation](#)

- bool **cancelled** = false
- [QAtomicInteger](#)< int > **itemsProcessed** = 0
- quint64 **maxBufferSize** = 2147483648  
*2 GB default*
- [QMutex](#) **mutex**
- [QMap](#)< [MLPipelineStage](#), [MLPipelinePerformanceProfile](#) > **performanceProfileList**
- [QMap](#)< [MLPipelineStage](#), [MLPipelineQueue](#) \* > **queues**
- [QThreadPool](#) \* **threadPool** = nullptr
- [QMutex](#) **threadStageMutex**
- [QAtomicInteger](#)< int > **totalItemCount** = 0
- quint64 **usedBufferSize** = 0
- [QList](#)< [QFutureWatcher](#)< bool > \* > **watchList**

## 6.597.1 Member Function Documentation

### 6.597.1.1 addMoreWorkers()

```
void Digikam::FacePipelineDetectRecognize::addMoreWorkers ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.597.1.2 classifier()

```
bool Digikam::FacePipelineDetectRecognize::classifier ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.597.1.3 extractor()

```
bool Digikam::FacePipelineDetectRecognize::extractor ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.597.1.4 finder()

```
bool Digikam::FacePipelineDetectRecognize::finder ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.597.1.5 loader()

```
bool Digikam::FacePipelineDetectRecognize::loader ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.597.1.6 start()

```
bool Digikam::FacePipelineDetectRecognize::start ( ) [override], [virtual]
```

Reimplemented from [Digikam::MLPipelineFoundation](#).

### 6.597.1.7 trainer()

```
bool Digikam::FacePipelineDetectRecognize::trainer ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

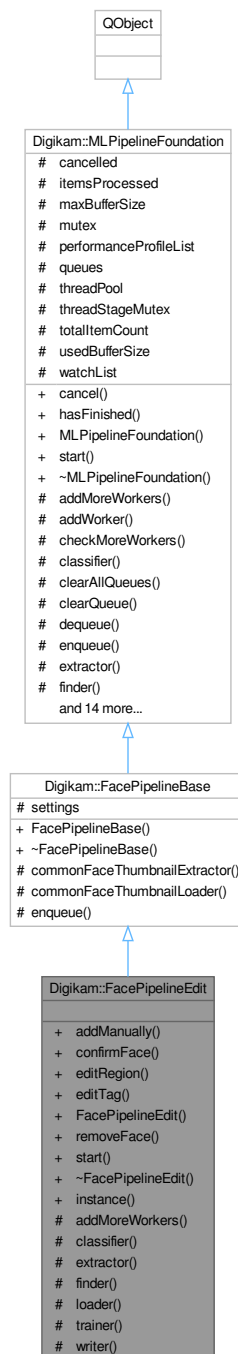
### 6.597.1.8 writer()

```
bool Digikam::FacePipelineDetectRecognize::writer ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

## 6.598 Digikam::FacePipelineEdit Class Reference

Inheritance diagram for Digikam::FacePipelineEdit:





### Public Member Functions

- [FaceTagsIface](#) **addManually** (const [ItemInfo](#) &info, const [DImg](#) &image, const [TagRegion](#) &region, bool retrain=true)
- [FaceTagsIface](#) **confirmFace** (const [ItemInfo](#) &info, const [FaceTagsIface](#) &face, int tagId, bool retrain=true)
- [FaceTagsIface](#) **editRegion** (const [ItemInfo](#) &info, const [FaceTagsIface](#) &face, const [TagRegion](#) &region, const [DImg](#) &image, bool retrain=true)
- [FaceTagsIface](#) **editTag** (const [ItemInfo](#) &info, const [FaceTagsIface](#) &face, int newTagId)
- void **removeFace** (const [ItemInfo](#) &info, const [FaceTagsIface](#) &face)
- bool **start** () override

### Public Member Functions inherited from [Digikam::FacePipelineBase](#)

- [FacePipelineBase](#) (const [FaceScanSettings](#) &\_settings)

### Public Member Functions inherited from [Digikam::MLPipelineFoundation](#)

- virtual void **cancel** ()
- bool **hasFinished** () const

### Static Public Member Functions

- static [FacePipelineEdit](#) \* **instance** ()

### Protected Member Functions

- void **addMoreWorkers** () override
- bool **classifier** () override
- bool **extractor** () override
- bool **finder** () override
- bool **loader** () override
- bool **trainer** () override
- bool **writer** () override

### Protected Member Functions inherited from [Digikam::FacePipelineBase](#)

- bool **commonFaceThumbnailExtractor** (const [QString](#) &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **commonFaceThumbnailLoader** (const [QString](#) &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **enqueue** ([MLPipelineQueue](#) \*thisQueue, [MLPipelinePackageFoundation](#) \*package) override

## Protected Member Functions inherited from [Digikam::MLPipelineFoundation](#)

- bool **addWorker** (const [MLPipelineStage](#) &stage)
- bool **checkMoreWorkers** (int totalItemCount, int currentItemCount, bool useFullCpu)
- void **clearAllQueues** ()
- void **clearQueue** ([MLPipelineQueue](#) \*thisQueue)
- virtual [MLPipelinePackageFoundation](#) \* **dequeue** ([MLPipelineQueue](#) \*thisQueue)
- void **notify** (MLPipelineNotification notification, const QString &\_name, const QString &\_path, int \_processed, const [DImg](#) &\_thumbnail)
- void **notify** (MLPipelineNotification notification, const QString &\_name, const QString &\_path, int \_processed, const [QIcon](#) &\_thumbnail)
- void **notify** (MLPipelineNotification notification, const QString &\_name, const QString &\_path, int \_processed, const [QImage](#) &\_thumbnail)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, int totalItemCount, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceStart** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- [MLPipelinePackageFoundation](#) \* **queueEndSignal** () const
- void **showPipelinePerformance** () const
- void **stageEnd** ([MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage)
- void **stageStart** ([QThread::Priority](#) threadPriority, [MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage, [MLPipelineQueue](#) \*&thisQueue, [MLPipelineQueue](#) \*&nextQueue)
- void **waitForStart** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::FacePipelineBase](#)

- enum [FilterMode](#) {  
[ScanAll](#) , [ScanNew](#) , [TrainNew](#) , [TrainAll](#) ,  
[TrainRemove](#) , [TrainReset](#) }
- enum [WriteMode](#) { [NormalWrite](#) , [OverwriteAllFaces](#) , [OverwriteUnconfirmed](#) }

## Public Types inherited from [Digikam::MLPipelineFoundation](#)

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue](#)< [MLPipelinePackageFoundation](#) \* > [MLPipelineQueue](#)
- enum [MLPipelineStage](#) {  
[Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) ,  
[Trainer](#) , [Writer](#) , [None](#) }

## Signals inherited from [Digikam::MLPipelineFoundation](#)

- void **finished** ()  
*Emitted when the last package has finished processing.*
- void **processed** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package has finished processing.*
- void **processing** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package begins processing.*
- void **progressValueChanged** (float progress)

- void **scheduled** ()  
*Emitted when processing is scheduled.*
- void **signalAddMoreWorkers** ()
- void **signalUpdateItemCount** (const qlonglong itemCount)
- void **skipped** (const MLPipelinePackageNotify::Ptr &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void **started** (const QString &message)  
*Emitted when processing has started.*

## Protected Attributes inherited from [Digikam::FacePipelineBase](#)

- [FaceScanSettings](#) **settings**

## Protected Attributes inherited from [Digikam::MLPipelineFoundation](#)

- bool **cancelled** = false
- QAtomicInteger< int > **itemsProcessed** = 0
- quint64 **maxBufferSize** = 2147483648  
*2 GB default*
- QMutex **mutex**
- QMap< [MLPipelineStage](#), [MLPipelinePerformanceProfile](#) > **performanceProfileList**
- QMap< [MLPipelineStage](#), [MLPipelineQueue](#) \* > **queues**
- QThreadPool \* **threadPool** = nullptr
- QMutex **threadStageMutex**
- QAtomicInteger< int > **totalItemCount** = 0
- quint64 **usedBufferSize** = 0
- QList< QFutureWatcher< bool > \* > **watchList**

### 6.598.1 Member Function Documentation

#### 6.598.1.1 addMoreWorkers()

```
void Digikam::FacePipelineEdit::addMoreWorkers ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

#### 6.598.1.2 classifier()

```
bool Digikam::FacePipelineEdit::classifier ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

#### 6.598.1.3 extractor()

```
bool Digikam::FacePipelineEdit::extractor ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

**6.598.1.4 finder()**

```
bool Digikam::FacePipelineEdit::finder ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

**6.598.1.5 loader()**

```
bool Digikam::FacePipelineEdit::loader ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

**6.598.1.6 start()**

```
bool Digikam::FacePipelineEdit::start ( ) [override], [virtual]
```

Reimplemented from [Digikam::MLPipelineFoundation](#).

**6.598.1.7 trainer()**

```
bool Digikam::FacePipelineEdit::trainer ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

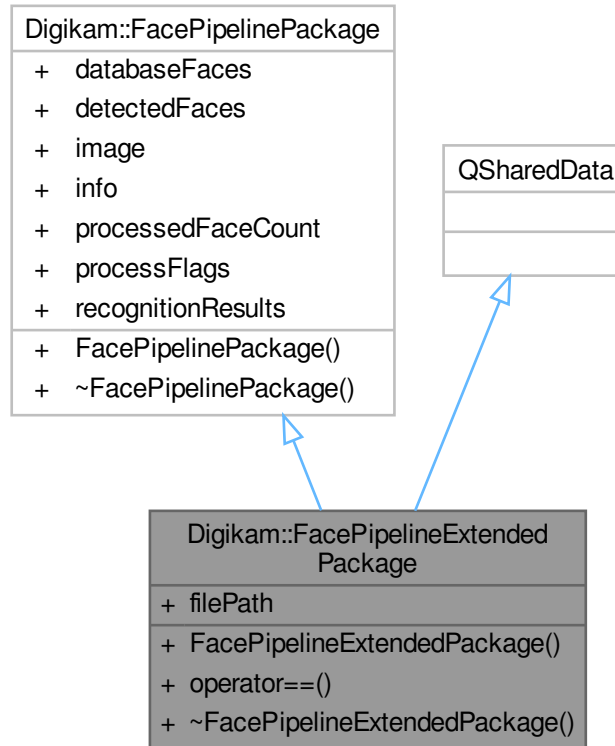
**6.598.1.8 writer()**

```
bool Digikam::FacePipelineEdit::writer ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

## 6.599 Digikam::FacePipelineExtendedPackage Class Reference

Inheritance diagram for Digikam::FacePipelineExtendedPackage:



### Public Types

- typedef QExplicitlySharedDataPointer< [FacePipelineExtendedPackage](#) > **Ptr**

### Public Types inherited from [Digikam::FacePipelinePackage](#)

- enum **ProcessFlag** {  
**NotProcessed** = 0 , **PreviewImageLoaded** = 1 << 0 , **ProcessedByDetector** = 1 << 1 , **ProcessedByRecognizer** = 1 << 2 ,  
**WrittenToDatabase** = 1 << 3 , **ProcessedByTrainer** = 1 << 4 }

### Public Member Functions

- bool **operator==** (const [LoadingDescription](#) &description) const

### Public Attributes

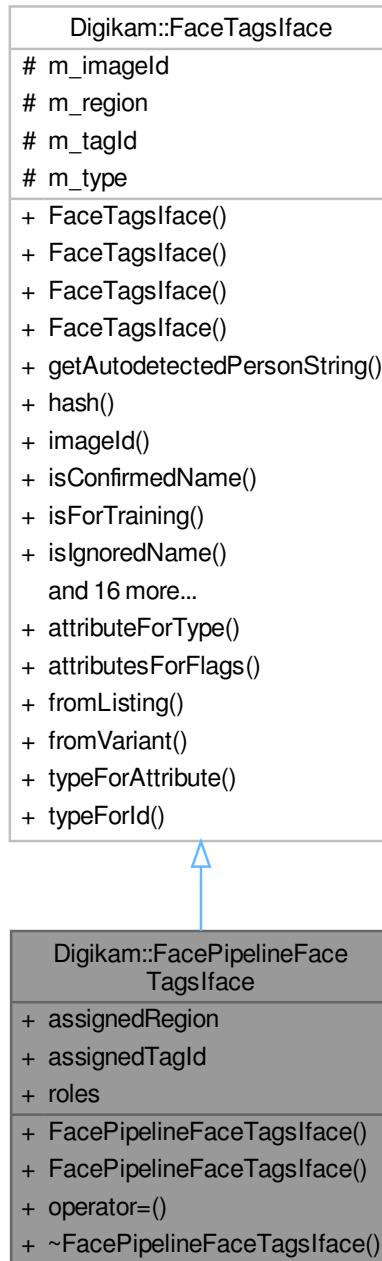
- QString **filePath**

## Public Attributes inherited from [Digikam::FacePipelinePackage](#)

- [FacePipelineFaceTagsfaceList](#) **databaseFaces**
- `QList< QRectF >` **detectedFaces**
- [DImg](#) **image**
- [ItemInfo](#) **info**
- `int` **processedFaceCount** = 0
- `ProcessFlags` **processFlags** = NotProcessed
- `QList< Identity >` **recognitionResults**

## 6.600 Digikam::FacePipelineFaceTagsIface Class Reference

Inheritance diagram for Digikam::FacePipelineFaceTagsIface:



### Public Types

- enum [Role](#) {
  - NoRole** = 0 , **GivenAsArgument** = 1 << 0 , **ReadFromDatabase** = 1 << 1 , **DetectedFromImage** = 1 <<

```

2 ,
ForRecognition = 1 << 10 , ForConfirmation = 1 << 11 , ForTraining = 1 << 12 , ForEditing = 1 << 13
,
Confirmed = 1 << 20 , Trained = 1 << 21 , Edited = 1 << 22 }

```

## Public Types inherited from [Digikam::FaceTagsIface](#)

- enum **Type** {
 **InvalidFace** = 0 , **UnknownName** = 1 << 0 , **UnconfirmedName** = 1 << 1 , **IgnoredName** = 1 << 2 ,
 **ConfirmedName** = 1 << 3 , **FaceForTraining** = 1 << 4 , **UnconfirmedTypes** = UnknownName |
 UnconfirmedName , **NormalFaces** = UnknownName | UnconfirmedName | IgnoredName | ConfirmedName
 ,
 **AllTypes** = UnknownName | UnconfirmedName | IgnoredName | ConfirmedName | FaceForTraining , **Type↔**
**First** = UnknownName , **TypeLast** = FaceForTraining }

## Public Member Functions

- FacePipelineFaceTagsIface** (const [FaceTagsIface](#) &face)
- FacePipelineFaceTagsIface** & **operator=** (const [FacePipelineFaceTagsIface](#) &other)

## Public Member Functions inherited from [Digikam::FaceTagsIface](#)

- FaceTagsIface** (const [FaceTagsIface](#) &other)
- FaceTagsIface** (const QString &attribute, qulonglong imageld, int tagId, const [TagRegion](#) &region)
- FaceTagsIface** (Type type, qulonglong imageld, int tagId, const [TagRegion](#) &region)
- QString **getAutodetectedPersonString** () const
- const QString **hash** () const
- qulonglong **imageld** () const
- bool **isConfirmedName** () const
- bool **isForTraining** () const
- bool **isIgnoredName** () const
- bool **isInvalidFace** () const
- bool **isNull** () const
- bool **isUnconfirmedName** () const
- bool **isUnconfirmedType** () const
- bool **isUnknownName** () const
- [FaceTagsIface](#) & **operator=** (const [FaceTagsIface](#) &other)
- bool **operator==** (const [FaceTagsIface](#) &other) const
- [TagRegion](#) **region** () const
- void **removeFaceTraining** () const
- void **setRegion** (const [TagRegion](#) &region)
- void **setTagId** (int tagId)
- void **setType** (Type type)
- int **tagId** () const
- QVariant **toVariant** () const
- Type **type** () const

## Public Attributes

- [TagRegion](#) **assignedRegion**
- int **assignedTagId** = 0
- Roles **roles** = NoRole



## Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::FaceTagsIface](#)

- static QString [attributeForType](#) (Type type)
- static QStringList [attributesForFlags](#) (TypeFlags flags)
- static [FaceTagsIface fromListing](#) (qulonglong imageid, const QList< QVariant > &values)
- static [FaceTagsIface fromVariant](#) (const QVariant &var)
- static Type [typeForAttribute](#) (const QString &attribute, int tagId=0)
- static Type [typeForId](#) (int tagId)

### Protected Attributes inherited from [Digikam::FaceTagsIface](#)

- qulonglong [m\\_imageId](#) = 0
- [TagRegion](#) [m\\_region](#)
- int [m\\_tagId](#) = 0
- Type [m\\_type](#) = InvalidFace

## 6.600.1 Member Enumeration Documentation

### 6.600.1.1 Role

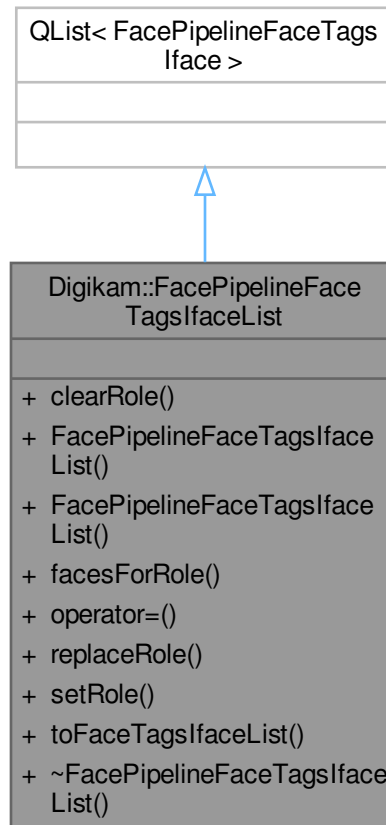
enum [Digikam::FacePipelineFaceTagsIface::Role](#)

#### Enumerator

GivenAsArgument	Source.
ForRecognition	Task.
Confirmed	Executed action (task is cleared).

## 6.601 Digikam::FacePipelineFaceTagsIfaceList Class Reference

Inheritance diagram for Digikam::FacePipelineFaceTagsIfaceList:

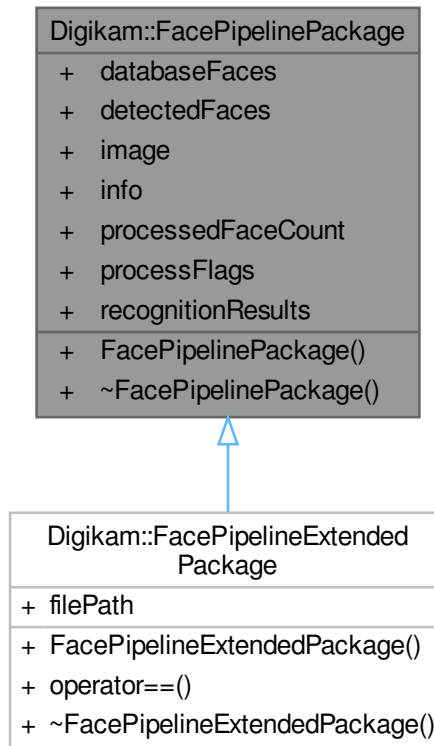


### Public Member Functions

- void **clearRole** (FacePipelineFaceTagsIface::Roles role)
- **FacePipelineFaceTagsIfaceList** (const QList< [FaceTagsIface](#) > &faces)
- [FacePipelineFaceTagsIfaceList](#) **facesForRole** (FacePipelineFaceTagsIface::Roles role) const
- [FacePipelineFaceTagsIfaceList](#) & **operator=** (const QList< [FaceTagsIface](#) > &faces)
- void **replaceRole** (FacePipelineFaceTagsIface::Roles remove, FacePipelineFaceTagsIface::Roles add)
- void **setRole** (FacePipelineFaceTagsIface::Roles role)
- QList< [FaceTagsIface](#) > **toFaceTagsIfaceList** () const

## 6.602 Digikam::FacePipelinePackage Class Reference

Inheritance diagram for Digikam::FacePipelinePackage:



### Public Types

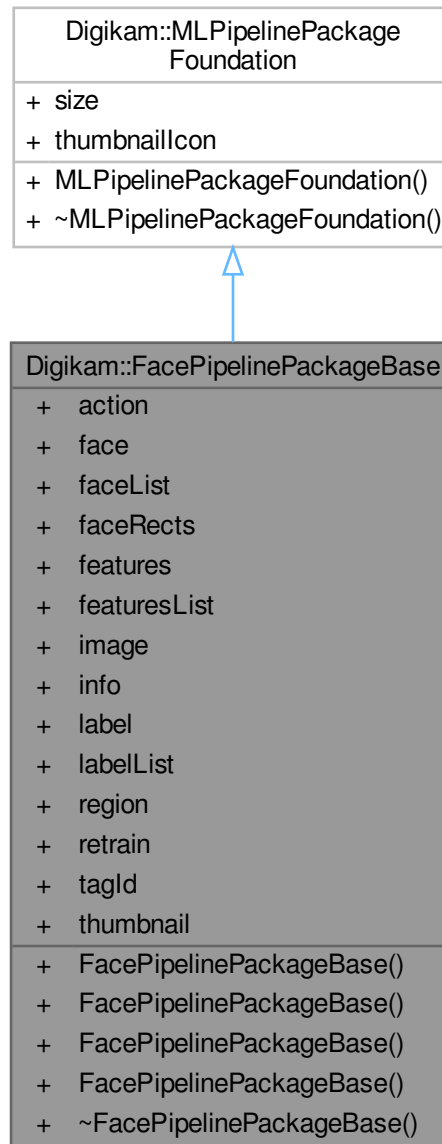
- enum **ProcessFlag** {
  - NotProcessed** = 0 , **PreviewImageLoaded** = 1 << 0 , **ProcessedByDetector** = 1 << 1 , **ProcessedByRecognizer** = 1 << 2 ,
  - WrittenToDatabase** = 1 << 3 , **ProcessedByTrainer** = 1 << 4 }

### Public Attributes

- [FacePipelineFaceTagsIfaceList](#) **databaseFaces**
- `QList< QRectF >` **detectedFaces**
- [DImg](#) **image**
- [ItemInfo](#) **info**
- `int` **processedFaceCount** = 0
- `ProcessFlags` **processFlags** = `NotProcessed`
- `QList< Identity >` **recognitionResults**

## 6.603 Digikam::FacePipelinePackageBase Class Reference

Inheritance diagram for Digikam::FacePipelinePackageBase:



### Public Types

- enum **EditPipelineAction** { **Confirm** , **Remove** , **EditTag** , **EditRegion** , **AddManually** }

### Public Member Functions

- **FacePipelinePackageBase** (const [ItemInfo](#) &\_info, const [FaceTagsIface](#) &\_face, int \_tagId, const [TagRegion](#) &\_region, const [DImg](#) &\_image, [EditPipelineAction](#) \_action, bool \_retrain)
- **FacePipelinePackageBase** (qulonglong \_imageId)
- **FacePipelinePackageBase** (qulonglong \_imageId, const [FaceTagsIface](#) &\_face)

### Public Attributes

- [EditPipelineAction](#) **action** = [EditPipelineAction::Confirm](#)
- [FaceTagsIface](#) **face**
- [QList](#)< [FaceTagsIface](#) > **faceList**
- [QList](#)< [QRectF](#) > **faceRects**
- [cv::Mat](#) **features**
- [QList](#)< [cv::Mat](#) > **featuresList**
- [DImg](#) **image**
- [ItemInfo](#) **info**
- int **label** = -1
- [QList](#)< int > **labelList**
- [TagRegion](#) **region**
- bool **retrain** = false
- int **tagId** = -1
- [QImage](#) **thumbnail**

### Public Attributes inherited from [Digikam::MLPipelinePackageFoundation](#)

- quint64 **size** = 0
- [QIcon](#) **thumbnailIcon**

## 6.604 Digikam::FacePipelineRecognize Class Reference

Inheritance diagram for Digikam::FacePipelineRecognize:



### Public Member Functions

- **FacePipelineRecognize** (const [FaceScanSettings](#) &\_settings)
- bool [start](#) () override

## Public Member Functions inherited from Digikam::FacePipelineBase

- **FacePipelineBase** (const [FaceScanSettings](#) &\_settings)

## Public Member Functions inherited from Digikam::MLPipelineFoundation

- virtual void [cancel](#) ()
- bool [hasFinished](#) () const

## Protected Member Functions

- void [addMoreWorkers](#) () override
- bool [classifier](#) () override
- bool [extractor](#) () override
- bool [finder](#) () override
- bool [loader](#) () override
- bool [trainer](#) () override
- bool [writer](#) () override

## Protected Member Functions inherited from Digikam::FacePipelineBase

- bool [commonFaceThumbnailExtractor](#) (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool [commonFaceThumbnailLoader](#) (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool [enqueue](#) ([MLPipelineQueue](#) \*thisQueue, [MLPipelinePackageFoundation](#) \*package) override

## Protected Member Functions inherited from Digikam::MLPipelineFoundation

- bool [addWorker](#) (const [MLPipelineStage](#) &stage)
- bool [checkMoreWorkers](#) (int totalItemCount, int currentItemCount, bool useFullCpu)
- void [clearAllQueues](#) ()
- void [clearQueue](#) ([MLPipelineQueue](#) \*thisQueue)
- virtual [MLPipelinePackageFoundation](#) \* [dequeue](#) ([MLPipelineQueue](#) \*thisQueue)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [DImg](#) &\_thumbnail)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QIcon](#) &\_thumbnail)
- void [notify](#) ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QImage](#) &\_thumbnail)
- void [pipelinePerformanceEnd](#) (const [MLPipelineStage](#) &stage, int totalItemCount, [QElapsedTimer](#) &timer)
- void [pipelinePerformanceEnd](#) (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- void [pipelinePerformanceStart](#) (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- [MLPipelinePackageFoundation](#) \* [queueEndSignal](#) () const
- void [showPipelinePerformance](#) () const
- void [stageEnd](#) ([MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage)
- void [stageStart](#) ([QThread::Priority](#) threadPriority, [MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage, [MLPipelineQueue](#) \*&thisQueue, [MLPipelineQueue](#) \*&nextQueue)
- void [waitForStart](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::FacePipelineBase](#)

- enum [FilterMode](#) { [ScanAll](#) , [ScanNew](#) , [TrainNew](#) , [TrainAll](#) , [TrainRemove](#) , [TrainReset](#) }
- enum [WriteMode](#) { [NormalWrite](#) , [OverwriteAllFaces](#) , [OverwriteUnconfirmed](#) }

### Public Types inherited from [Digikam::MLPipelineFoundation](#)

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue](#)< [MLPipelinePackageFoundation](#) \* > [MLPipelineQueue](#)
- enum [MLPipelineStage](#) { [Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) , [Trainer](#) , [Writer](#) , [None](#) }

### Signals inherited from [Digikam::MLPipelineFoundation](#)

- void [finished](#) ()  
*Emitted when the last package has finished processing.*
- void [processed](#) (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package has finished processing.*
- void [processing](#) (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package begins processing.*
- void [progressValueChanged](#) (float progress)
- void [scheduled](#) ()  
*Emitted when processing is scheduled.*
- void [signalAddMoreWorkers](#) ()
- void [signalUpdateItemCount](#) (const qlonglong itemCount)
- void [skipped](#) (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void [started](#) (const QString &message)  
*Emitted when processing has started.*

### Protected Attributes inherited from [Digikam::FacePipelineBase](#)

- [FaceScanSettings](#) [settings](#)

### Protected Attributes inherited from [Digikam::MLPipelineFoundation](#)

- bool [cancelled](#) = false
- [QAtomicInteger](#)< int > [itemsProcessed](#) = 0
- quint64 [maxBufferSize](#) = 2147483648  
*2 GB default*
- [QMutex](#) [mutex](#)
- [QMap](#)< [MLPipelineStage](#), [MLPipelinePerformanceProfile](#) > [performanceProfileList](#)
- [QMap](#)< [MLPipelineStage](#), [MLPipelineQueue](#) \* > [queues](#)
- [QThreadPool](#) \* [threadPool](#) = nullptr
- [QMutex](#) [threadStageMutex](#)
- [QAtomicInteger](#)< int > [totalItemCount](#) = 0
- quint64 [usedBufferSize](#) = 0
- [QList](#)< [QFutureWatcher](#)< bool > \* > [watchList](#)



## 6.604.1 Member Function Documentation

### 6.604.1.1 addMoreWorkers()

```
void Digikam::FacePipelineRecognize::addMoreWorkers ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.604.1.2 classifier()

```
bool Digikam::FacePipelineRecognize::classifier ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.604.1.3 extractor()

```
bool Digikam::FacePipelineRecognize::extractor ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.604.1.4 finder()

```
bool Digikam::FacePipelineRecognize::finder ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.604.1.5 loader()

```
bool Digikam::FacePipelineRecognize::loader ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.604.1.6 start()

```
bool Digikam::FacePipelineRecognize::start ( ) [override], [virtual]
```

Reimplemented from [Digikam::MLPipelineFoundation](#).

### 6.604.1.7 trainer()

```
bool Digikam::FacePipelineRecognize::trainer ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.604.1.8 writer()

```
bool Digikam::FacePipelineRecognize::writer ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

## 6.605 Digikam::FacePipelineReset Class Reference

Inheritance diagram for Digikam::FacePipelineReset:



**Public Member Functions**

- **FacePipelineReset** (const [FaceScanSettings](#) &\_settings)
- bool **start** () override

**Public Member Functions inherited from [Digikam::FacePipelineBase](#)**

- **FacePipelineBase** (const [FaceScanSettings](#) &\_settings)

**Public Member Functions inherited from [Digikam::MLPipelineFoundation](#)**

- virtual void **cancel** ()
- bool **hasFinished** () const

**Protected Member Functions**

- void **addMoreWorkers** () override
- bool **classifier** () override
- bool **extractor** () override
- bool **finder** () override
- bool **loader** () override
- bool **trainer** () override
- bool **writer** () override

**Protected Member Functions inherited from [Digikam::FacePipelineBase](#)**

- bool **commonFaceThumbnailExtractor** (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **commonFaceThumbnailLoader** (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **enqueue** ([MLPipelineQueue](#) \*thisQueue, [MLPipelinePackageFoundation](#) \*package) override

**Protected Member Functions inherited from [Digikam::MLPipelineFoundation](#)**

- bool **addWorker** (const [MLPipelineStage](#) &stage)
- bool **checkMoreWorkers** (int totalItemCount, int currentItemCount, bool useFullCpu)
- void **clearAllQueues** ()
- void **clearQueue** ([MLPipelineQueue](#) \*thisQueue)
- virtual [MLPipelinePackageFoundation](#) \* **dequeue** ([MLPipelineQueue](#) \*thisQueue)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [DImg](#) &\_thumbnail)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QIcon](#) &\_thumbnail)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QImage](#) &\_thumbnail)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, int totalItemCount, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceStart** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- [MLPipelinePackageFoundation](#) \* **queueEndSignal** () const
- void **showPipelinePerformance** () const
- void **stageEnd** ([MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage)
- void **stageStart** ([QThread::Priority](#) threadPriority, [MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage, [MLPipelineQueue](#) \*&thisQueue, [MLPipelineQueue](#) \*&nextQueue)
- void **waitForStart** ()

### Additional Inherited Members

#### Public Types inherited from [Digikam::FacePipelineBase](#)

- enum [FilterMode](#) { [ScanAll](#) , [ScanNew](#) , [TrainNew](#) , [TrainAll](#) , [TrainRemove](#) , [TrainReset](#) }
- enum [WriteMode](#) { [NormalWrite](#) , [OverwriteAllFaces](#) , [OverwriteUnconfirmed](#) }

#### Public Types inherited from [Digikam::MLPipelineFoundation](#)

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue](#)< [MLPipelinePackageFoundation](#) \* > [MLPipelineQueue](#)
- enum [MLPipelineStage](#) { [Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) , [Trainer](#) , [Writer](#) , [None](#) }

#### Signals inherited from [Digikam::MLPipelineFoundation](#)

- void **finished** ()  
*Emitted when the last package has finished processing.*
- void **processed** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package has finished processing.*
- void **processing** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package begins processing.*
- void **progressValueChanged** (float progress)
- void **scheduled** ()  
*Emitted when processing is scheduled.*
- void **signalAddMoreWorkers** ()
- void **signalUpdateItemCount** (const qlonglong itemCount)
- void **skipped** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void **started** (const [QString](#) &message)  
*Emitted when processing has started.*

#### Protected Attributes inherited from [Digikam::FacePipelineBase](#)

- [FaceScanSettings](#) **settings**

#### Protected Attributes inherited from [Digikam::MLPipelineFoundation](#)

- bool **cancelled** = false
- [QAtomicInteger](#)< int > **itemsProcessed** = 0
- quint64 **maxBufferSize** = 2147483648  
*2 GB default*
- [QMutex](#) **mutex**
- [QMap](#)< [MLPipelineStage](#), [MLPipelinePerformanceProfile](#) > **performanceProfileList**
- [QMap](#)< [MLPipelineStage](#), [MLPipelineQueue](#) \* > **queues**
- [QThreadPool](#) \* **threadPool** = nullptr
- [QMutex](#) **threadStageMutex**
- [QAtomicInteger](#)< int > **totalItemCount** = 0
- quint64 **usedBufferSize** = 0
- [QList](#)< [QFutureWatcher](#)< bool > \* > **watchList**

## 6.605.1 Member Function Documentation

### 6.605.1.1 addMoreWorkers()

```
void Digikam::FacePipelineReset::addMoreWorkers ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.605.1.2 classifier()

```
bool Digikam::FacePipelineReset::classifier ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.605.1.3 extractor()

```
bool Digikam::FacePipelineReset::extractor ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.605.1.4 finder()

```
bool Digikam::FacePipelineReset::finder ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.605.1.5 loader()

```
bool Digikam::FacePipelineReset::loader ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.605.1.6 start()

```
bool Digikam::FacePipelineReset::start ( ) [override], [virtual]
```

Reimplemented from [Digikam::MLPipelineFoundation](#).

### 6.605.1.7 trainer()

```
bool Digikam::FacePipelineReset::trainer ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

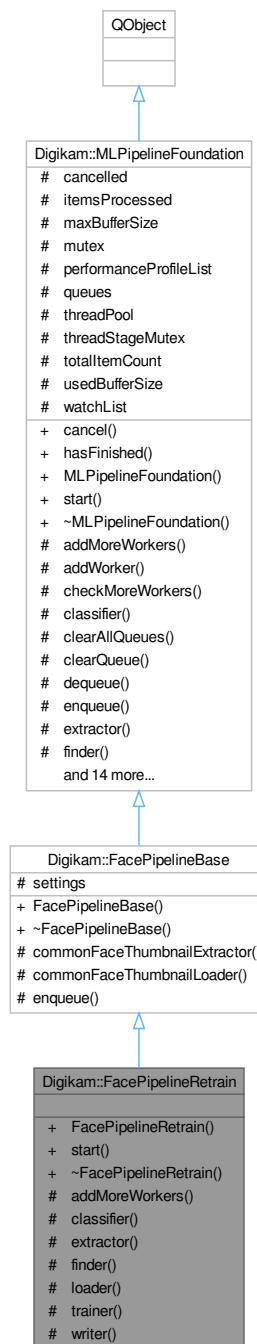
### 6.605.1.8 writer()

```
bool Digikam::FacePipelineReset::writer ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

## 6.606 Digikam::FacePipelineRetrain Class Reference

Inheritance diagram for Digikam::FacePipelineRetrain:



**Public Member Functions**

- **FacePipelineRetrain** (const [FaceScanSettings](#) &\_settings)
- bool **start** () override

**Public Member Functions inherited from [Digikam::FacePipelineBase](#)**

- **FacePipelineBase** (const [FaceScanSettings](#) &\_settings)

**Public Member Functions inherited from [Digikam::MLPipelineFoundation](#)**

- virtual void **cancel** ()
- bool **hasFinished** () const

**Protected Member Functions**

- void **addMoreWorkers** () override
- bool **classifier** () override
- bool **extractor** () override
- bool **finder** () override
- bool **loader** () override
- bool **trainer** () override
- bool **writer** () override

**Protected Member Functions inherited from [Digikam::FacePipelineBase](#)**

- bool **commonFaceThumbnailExtractor** (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **commonFaceThumbnailLoader** (const QString &pipelineName, [MLPipelineFoundation::MLPipelineStage](#) thisStage, [MLPipelineFoundation::MLPipelineStage](#) nextStage)
- bool **enqueue** ([MLPipelineQueue](#) \*thisQueue, [MLPipelinePackageFoundation](#) \*package) override

**Protected Member Functions inherited from [Digikam::MLPipelineFoundation](#)**

- bool **addWorker** (const [MLPipelineStage](#) &stage)
- bool **checkMoreWorkers** (int totalItemCount, int currentItemCount, bool useFullCpu)
- void **clearAllQueues** ()
- void **clearQueue** ([MLPipelineQueue](#) \*thisQueue)
- virtual [MLPipelinePackageFoundation](#) \* **dequeue** ([MLPipelineQueue](#) \*thisQueue)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [DImg](#) &\_thumbnail)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QIcon](#) &\_thumbnail)
- void **notify** ([MLPipelineNotification](#) notification, const QString &\_name, const QString &\_path, int \_processed, const [QImage](#) &\_thumbnail)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, int totalItemCount, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceEnd** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- void **pipelinePerformanceStart** (const [MLPipelineStage](#) &stage, [QElapsedTimer](#) &timer)
- [MLPipelinePackageFoundation](#) \* **queueEndSignal** () const
- void **showPipelinePerformance** () const
- void **stageEnd** ([MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage)
- void **stageStart** ([QThread::Priority](#) threadPriority, [MLPipelineStage](#) thisStage, [MLPipelineStage](#) nextStage, [MLPipelineQueue](#) \*&thisQueue, [MLPipelineQueue](#) \*&nextQueue)
- void **waitForStart** ()

## Additional Inherited Members

### Public Types inherited from [Digikam::FacePipelineBase](#)

- enum [FilterMode](#) { [ScanAll](#) , [ScanNew](#) , [TrainNew](#) , [TrainAll](#) , [TrainRemove](#) , [TrainReset](#) }
- enum [WriteMode](#) { [NormalWrite](#) , [OverwriteAllFaces](#) , [OverwriteUnconfirmed](#) }

### Public Types inherited from [Digikam::MLPipelineFoundation](#)

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue](#)< [MLPipelinePackageFoundation](#) \* > [MLPipelineQueue](#)
- enum [MLPipelineStage](#) { [Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) , [Trainer](#) , [Writer](#) , [None](#) }

### Signals inherited from [Digikam::MLPipelineFoundation](#)

- void **finished** ()  
*Emitted when the last package has finished processing.*
- void **processed** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package has finished processing.*
- void **processing** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one package begins processing.*
- void **progressValueChanged** (float progress)
- void **scheduled** ()  
*Emitted when processing is scheduled.*
- void **signalAddMoreWorkers** ()
- void **signalUpdateItemCount** (const qlonglong itemCount)
- void **skipped** (const [MLPipelinePackageNotify::Ptr](#) &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void **started** (const [QString](#) &message)  
*Emitted when processing has started.*

### Protected Attributes inherited from [Digikam::FacePipelineBase](#)

- [FaceScanSettings](#) **settings**

### Protected Attributes inherited from [Digikam::MLPipelineFoundation](#)

- bool **cancelled** = false
- [QAtomicInteger](#)< int > **itemsProcessed** = 0
- quint64 **maxBufferSize** = 2147483648  
*2 GB default*
- [QMutex](#) **mutex**
- [QMap](#)< [MLPipelineStage](#), [MLPipelinePerformanceProfile](#) > **performanceProfileList**
- [QMap](#)< [MLPipelineStage](#), [MLPipelineQueue](#) \* > **queues**
- [QThreadPool](#) \* **threadPool** = nullptr
- [QMutex](#) **threadStageMutex**
- [QAtomicInteger](#)< int > **totalItemCount** = 0
- quint64 **usedBufferSize** = 0
- [QList](#)< [QFutureWatcher](#)< bool > \* > **watchList**



## 6.606.1 Member Function Documentation

### 6.606.1.1 addMoreWorkers()

```
void Digikam::FacePipelineRetrain::addMoreWorkers ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.606.1.2 classifier()

```
bool Digikam::FacePipelineRetrain::classifier ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.606.1.3 extractor()

```
bool Digikam::FacePipelineRetrain::extractor ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.606.1.4 finder()

```
bool Digikam::FacePipelineRetrain::finder ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.606.1.5 loader()

```
bool Digikam::FacePipelineRetrain::loader ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

### 6.606.1.6 start()

```
bool Digikam::FacePipelineRetrain::start ( ) [override], [virtual]
```

Reimplemented from [Digikam::MLPipelineFoundation](#).

### 6.606.1.7 trainer()

```
bool Digikam::FacePipelineRetrain::trainer ( ) [inline], [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

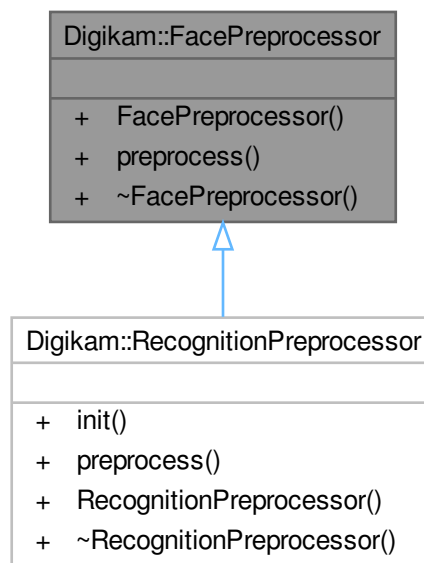
### 6.606.1.8 writer()

```
bool Digikam::FacePipelineRetrain::writer ( ) [override], [protected], [virtual]
```

Implements [Digikam::MLPipelineFoundation](#).

## 6.607 Digikam::FacePreprocessor Class Reference

Inheritance diagram for Digikam::FacePreprocessor:

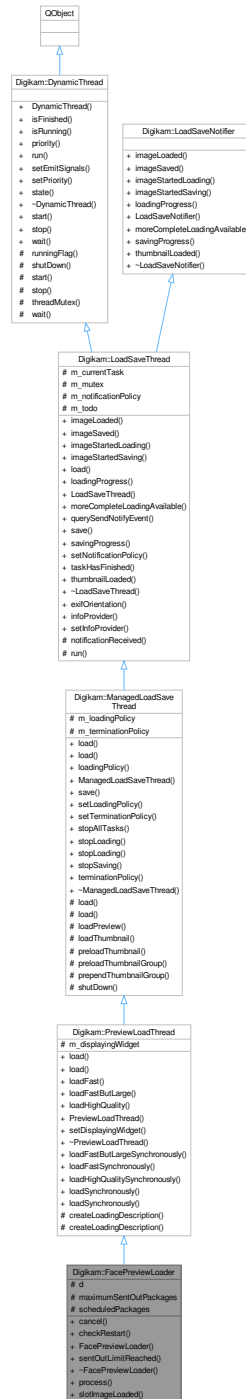


### Public Member Functions

- virtual cv::Mat **preprocess** (const cv::Mat &image) const =0

## 6.608 Digikam::FacePreviewLoader Class Reference

Inheritance diagram for Digikam::FacePreviewLoader:



### Public Slots

- void **process** (const FacePipelineExtendedPackage::Ptr &package)
- void **slotImageLoaded** (const LoadingDescription &loadingDescription, const DImg &img)

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals

- void **processed** (const [FacePipelineExtendedPackage::Ptr](#) &package)

## Signals inherited from [Digikam::LoadSaveThread](#)

- void **signalImageLoaded** (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)
- void **signalImageSaved** (const [QString](#) &filePath, bool success)
- void **signalImageStartedLoading** (const [LoadingDescription](#) &loadingDescription)
- void **signalImageStartedSaving** (const [QString](#) &filePath)
- void **signalLoadingProgress** (const [LoadingDescription](#) &loadingDescription, float progress)
- void **signalMoreCompleteLoadingAvailable** (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription)
- void **signalSavingProgress** (const [QString](#) &filePath, float progress)
- void **signalThumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const [QImage](#) &img)

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Public Member Functions

- void **cancel** ()
- void **checkRestart** ()
- **FacePreviewLoader** ([FacePipeline::Private](#) \*const dd)
- bool **sentOutLimitReached** () const

## Public Member Functions inherited from [Digikam::PreviewLoadThread](#)

- void **load** (const [LoadingDescription](#) &description)
- void **load** (const [QString](#) &filePath, const [PreviewSettings](#) &settings, int size=0)
- void **loadFast** (const [QString](#) &filePath, int size)
- void **loadFastButLarge** (const [QString](#) &filePath, int minimumSize)
- void **loadHighQuality** (const [QString](#) &filePath, [PreviewSettings::RawLoading](#) rawLoadingMode=[PreviewSettings::RawPreviewAutomatic](#))
- [PreviewLoadThread](#) ([QObject](#) \*const parent=nullptr)
- void **setDisplayingWidget** ([QWidget](#) \*const widget)

## Public Member Functions inherited from Digikam::ManagedLoadSaveThread

- void [load](#) (const [LoadingDescription](#) &description)
- void **load** (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- [LoadingPolicy](#) **loadingPolicy** () const
- [ManagedLoadSaveThread](#) (QObject \*const parent=nullptr)
- void [save](#) (const [DImg](#) &image, const [QString](#) &filePath, const [QString](#) &format)
- void [setLoadingPolicy](#) ([LoadingPolicy](#) policy)
- void **setTerminationPolicy** ([TerminationPolicy](#) terminationPolicy)
- void **stopAllTasks** ()
- void [stopLoading](#) (const [LoadingDescription](#) &desc, [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopLoading](#) (const [QString](#) &filePath=[QString](#)(), [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopSaving](#) (const [QString](#) &filePath=[QString](#)())
- [TerminationPolicy](#) **terminationPolicy** () const

## Public Member Functions inherited from Digikam::LoadSaveThread

- void [imageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img) override
- void [imageSaved](#) (const [QString](#) &filePath, bool success) override
- void [imageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription) override
- void [imageStartedSaving](#) (const [QString](#) &filePath) override
- void [load](#) (const [LoadingDescription](#) &description)
- void [loadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress) override
- [LoadSaveThread](#) (QObject \*const parent=nullptr)
- void [moreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription) override
- virtual bool **querySendNotifyEvent** () const
- void [save](#) (const [DImg](#) &image, const [QString](#) &filePath, const [QString](#) &format)
- void [savingProgress](#) (const [QString](#) &filePath, float progress) override
- void **setNotificationPolicy** ([NotificationPolicy](#) notificationPolicy)
- virtual void **taskHasFinished** ()
- void [thumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [QImage](#) &img) override
- [~LoadSaveThread](#) () override

## Public Member Functions inherited from Digikam::DynamicThread

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

## Protected Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- int **maximumSentOutPackages** = [qMin](#)([QThread::idealThreadCount](#)(), 4)  
*Upper limit for memory cost.*
- [PackageLoadingDescriptionList](#) **scheduledPackages**

### Protected Attributes inherited from [Digikam::PreviewLoadThread](#)

- `QWidget * m_displayingWidget = nullptr`

### Protected Attributes inherited from [Digikam::ManagedLoadSaveThread](#)

- `LoadingPolicy m_loadingPolicy = LoadingPolicyAppend`
- `TerminationPolicy m_terminationPolicy = TerminationPolicyTerminateLoading`

### Protected Attributes inherited from [Digikam::LoadSaveThread](#)

- `LoadSaveTask * m_currentTask = nullptr`
- `QMutex m_mutex`
- `NotificationPolicy m_notificationPolicy = NotificationPolicyTimeLimited`
- `QList< LoadSaveTask * > m_todo`

### Additional Inherited Members

### Public Types inherited from [Digikam::ManagedLoadSaveThread](#)

- enum `LoadingMode` { `LoadingModeNormal` , `LoadingModeShared` }
- enum `LoadingPolicy` { `LoadingPolicyFirstRemovePrevious` , `LoadingPolicyPrepend` , `LoadingPolicySimplePrepend` , `LoadingPolicyAppend` , `LoadingPolicySimpleAppend` , `LoadingPolicyPreload` }
- enum `LoadingTaskFilter` { `LoadingTaskFilterAll` , `LoadingTaskFilterPreloading` }
- enum `TerminationPolicy` { `TerminationPolicyTerminateLoading` , `TerminationPolicyTerminatePreloading` , `TerminationPolicyWait` , `TerminationPolicyTerminateAll` }

### Public Types inherited from [Digikam::LoadSaveThread](#)

- enum `AccessMode` { `AccessModeRead` , `AccessModeReadWrite` }
- enum `NotificationPolicy` { `NotificationPolicyDirect` , `NotificationPolicyTimeLimited` }

### Public Types inherited from [Digikam::DynamicThread](#)

- enum `State` { `Inactive` , `Scheduled` , `Running` , `Deactivating` }

### Static Public Member Functions inherited from [Digikam::PreviewLoadThread](#)

- static `DImg loadFastButLargeSynchronously` (const `QString` &filePath, int minimumSize, const `IccProfile` &profile=`IccProfile()`)
- static `DImg loadFastSynchronously` (const `QString` &filePath, int size, const `IccProfile` &profile=`IccProfile()`)
- static `DImg loadHighQualitySynchronously` (const `QString` &filePath, `PreviewSettings::RawLoading` raw←`LoadingMode=PreviewSettings::RawPreviewAutomatic`, const `IccProfile` &profile=`IccProfile()`)
- static `DImg loadSynchronously` (const `LoadingDescription` &description)
- static `DImg loadSynchronously` (const `QString` &filePath, const `PreviewSettings` &previewSettings, int size, const `IccProfile` &profile=`IccProfile()`)

### Static Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- static int [exifOrientation](#) (const QString &filePath, const [DMetadata](#) &metadata, bool isRaw, bool fromRaw↔  
EmbeddedPreview)
- static [LoadSaveFileInfoProvider](#) \* [infoProvider](#) ()
- static void [setInfoProvider](#) ([LoadSaveFileInfoProvider](#) \*const infoProvider)

### Protected Member Functions inherited from [Digikam::PreviewLoadThread](#)

- [LoadingDescription](#) [createLoadingDescription](#) (const QString &filePath, const [PreviewSettings](#) &settings,  
int size)

### Protected Member Functions inherited from [Digikam::ManagedLoadSaveThread](#)

- void [load](#) (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void [load](#) (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [LoadingPolicy](#) policy,  
[AccessMode](#) mode=[AccessModeReadWrite](#))
- void [loadPreview](#) (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- void [loadThumbnail](#) (const [LoadingDescription](#) &description)
- void [preloadThumbnail](#) (const [LoadingDescription](#) &description)
- void [preloadThumbnailGroup](#) (const QList< [LoadingDescription](#) > &descriptions)
- void [prependThumbnailGroup](#) (const QList< [LoadingDescription](#) > &descriptions)
- void [shutDown](#) ()

### Protected Member Functions inherited from [Digikam::LoadSaveThread](#)

- void [notificationReceived](#) ()
- void [run](#) () override

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

### Static Protected Member Functions inherited from [Digikam::PreviewLoadThread](#)

- static [LoadingDescription](#) [createLoadingDescription](#) (const QString &filePath, const [PreviewSettings](#) &set-  
tings, int size, const [IccProfile](#) &profile)

## 6.609 Digikam::FaceRejectionOverlay Class Reference

Inheritance diagram for Digikam::FaceRejectionOverlay:



### Signals

- void **rejectFaces** (const QList< QModelIndex > &indexes)



## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- **FaceRejectionOverlay** (QObject \*const parent)
- void [setActive](#) (bool active) override

## Public Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- [ItemViewHoverButton](#) \* **button** () const
- **HoverButtonDelegateOverlay** (QObject \*const parent)
- void [setActive](#) (bool active) override

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotClicked** ()

## Protected Slots inherited from [Digikam::HoverButtonDelegateOverlay](#)

- void **slotEntered** (const QModelIndex &index) override
- void **slotReset** () override

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void [slotEntered](#) (const QModelIndex &index)
- virtual void **slotLayoutChanged** ()
- virtual void [slotReset](#) ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

### Protected Member Functions

- bool [checkIndex](#) (const QModelIndex &index) const override
- [ItemViewHoverButton](#) \* [createButton](#) () override
- void [updateButton](#) (const QModelIndex &index) override
- void [widgetEnterEvent](#) () override
- void [widgetLeaveEvent](#) () override

## Protected Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- QWidget \* [createWidget](#) () override
- void [visualChange](#) () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- void [widgetLeaveNotifyMultiple](#) ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

### Additional Inherited Members

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.609.1 Member Function Documentation

### 6.609.1.1 checkIndex()

```
bool Digikam::FaceRejectionOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.609.1.2 createButton()

```
ItemViewHoverButton * Digikam::FaceRejectionOverlay::createButton ( ) [override], [protected],
[virtual]
```

Create your widget here. Pass view() as parent.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.609.1.3 setActive()

```
void Digikam::FaceRejectionOverlay::setActive (
    bool active ) [override], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.609.1.4 updateButton()

```
void Digikam::FaceRejectionOverlay::updateButton (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.609.1.5 widgetEnterEvent()

```
void Digikam::FaceRejectionOverlay::widgetEnterEvent ( ) [override], [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.609.1.6 widgetLeaveEvent()

```
void Digikam::FaceRejectionOverlay::widgetLeaveEvent ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.610 Digikam::FaceRejectionOverlayButton Class Reference

Inheritance diagram for Digikam::FaceRejectionOverlayButton:



## Public Member Functions

- **FaceRejectionOverlayButton** (QAbstractItemView \*const parentView)
- QSize **sizeHint** () const override

## Public Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- QModelIndex **index** () const
- void **initIcon** ()
- **ItemViewHoverButton** (QAbstractItemView \*const parentView)
- void **reset** ()
- void **setIndex** (const QModelIndex &index)
- void **setVisible** (bool visible) override

## Protected Member Functions

- QIcon **icon** () override
- void **updateToolTip** () override

## Protected Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- void **enterEvent** (QEnterEvent \*event)
- void **leaveEvent** (QEvent \*event)
- void **paintEvent** (QPaintEvent \*event)
- void **setup** ()

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ItemViewHoverButton](#)

- void **refreshIcon** ()
- void **setFadingValue** (int value)
- void **startFading** ()
- void **stopFading** ()

## Protected Attributes inherited from [Digikam::ItemViewHoverButton](#)

- QTimerLine \* **m\_fadingTimeLine** = nullptr
- int **m\_fadingValue** = 0
- QIcon **m\_icon**
- QPersistentModelIndex **m\_index**
- bool **m\_isHovered** = false

## 6.610.1 Member Function Documentation

### 6.610.1.1 icon()

```
QIcon Digikam::FaceRejectionOverlayButton::icon ( ) [override], [protected], [virtual]
```

Return your icon here. Will be queried again on toggle.

Implements [Digikam::ItemViewHoverButton](#).

### 6.610.1.2 sizeHint()

```
QSize Digikam::FaceRejectionOverlayButton::sizeHint ( ) const [override], [virtual]
```

Reimplement to match the size of your icon

Implements [Digikam::ItemViewHoverButton](#).

### 6.610.1.3 updateToolTip()

```
void Digikam::FaceRejectionOverlayButton::updateToolTip ( ) [override], [protected], [virtual]
```

Optionally update tooltip here. Will be called again on state change.

Reimplemented from [Digikam::ItemViewHoverButton](#).

## 6.611 Digikam::FaceScanSettings Class Reference

### Public Types

- enum [AlreadyScannedHandling](#) { [Skip](#) , [Rescan](#) , [ClearAll](#) , [RecognizeOnly](#) }
- enum [FaceDetectionModel](#) { [SSDMOBILENET](#) , [YOLOv3](#) , [YuNet](#) }
- enum [FaceDetectionSize](#) { [ExtraSmall](#) , [Small](#) , [Medium](#) , [Large](#) , [ExtraLarge](#) }
- enum [FaceRecognitionModel](#) { [OpenFace](#) , [SFace](#) }
- enum [ScanTask](#) { [DetectAndRecognize](#) , [RecognizeMarkedFaces](#) , [RetrainAll](#) , [Reset](#) }

### Public Attributes

- AlbumList **albums**  
*Albums to scan.*
- [AlreadyScannedHandling](#) **alreadyScannedHandling** = [Skip](#)
- int [detectAccuracy](#) = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Detection accuracy.*
- [FaceDetectionModel](#) **detectModel** = [FaceDetectionModel::YuNet](#)  
*Detection Model.*
- [FaceDetectionSize](#) **detectSize** = [FaceDetectionSize::Large](#)  
*Detection Model.*
- [ItemInfoList](#) **infos**  
*Image infos to scan.*
- int [recognizeAccuracy](#) = DNN\_MODEL\_THRESHOLD\_NOT\_SET  
*Detection accuracy.*
- [FaceRecognitionModel](#) **recognizeModel** = [FaceRecognitionModel::OpenFace](#)  
*Detection Model.*
- [ScanTask](#) **task** = [DetectAndRecognize](#)
- bool **useFullCpu** = false  
*Processing power.*
- bool **wholeAlbums** = false  
*Whole albums checked.*

## 6.611.1 Member Enumeration Documentation

### 6.611.1.1 AlreadyScannedHandling

enum `Digikam::FaceScanSettings::AlreadyScannedHandling`

To detect and recognize.

## Enumerator

Skip	Skip faces from images already scanned.
Rescan	Rescan faces from images already scanned.
ClearAll	Clear all faces data from images already scanned. Clear identities and training data from FacesDb.
RecognizeOnly	Recognize faces from images already scanned.

**6.611.1.2 FaceDetectionModel**

enum `Digikam::FaceScanSettings::FaceDetectionModel`

Face detection AI models.

## Enumerator

SSDMOBILENET	SSD MobileNet neural network inference [ <a href="https://github.com/arunponnusamy/cvlib">https://github.com/arunponnusamy/cvlib</a> ].
YOLOv3	YOLO neural network inference [ <a href="https://github.com/sthanhng/yoloface">https://github.com/sthanhng/yoloface</a> ].
YuNet	YuNet neural network inference [ <a href="https://github.com/opencv/opencv_zoo/tree/main">https://github.com/opencv/opencv_zoo/tree/main</a> ].

**6.611.1.3 FaceDetectionSize**

enum `Digikam::FaceScanSettings::FaceDetectionSize`

Face detection size.

**6.611.1.4 FaceRecognitionModel**

enum `Digikam::FaceScanSettings::FaceRecognitionModel`

Face recognition AI models.

## Enumerator

OpenFace	OpenFace pre-trained neural network model [ <a href="https://github.com/sahilshah/openface/tree/master">https://github.com/sahilshah/openface/tree/master</a> ].
SFace	SFace pre-trained neural network model [ <a href="https://github.com/opencv/opencv_zoo/blob/main/models/face_recognition_sface/">https://github.com/opencv/opencv_zoo/blob/main/models/face_recognition_sface/</a> ].

**6.611.1.5 ScanTask**

enum `Digikam::FaceScanSettings::ScanTask`

Different possible tasks processed while scanning operation.



## Enumerator

DetectAndRecognize	Detect and recognize faces only.
RecognizeMarkedFaces	Recognize already marked faces only.
RetrainAll	Retrain faces only.

## 6.611.2 Member Data Documentation

### 6.611.2.1 detectAccuracy

```
int Digikam::FaceScanSettings::detectAccuracy = DNN_MODEL_THRESHOLD_NOT_SET
```

use default value from dnnmodels.conf

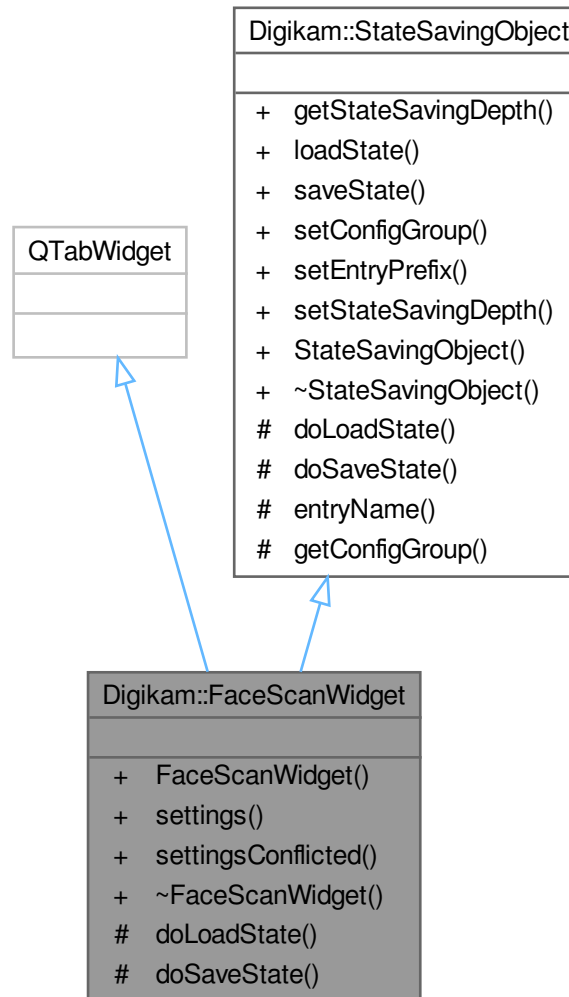
### 6.611.2.2 recognizeAccuracy

```
int Digikam::FaceScanSettings::recognizeAccuracy = DNN_MODEL_THRESHOLD_NOT_SET
```

use default value from dnnmodels.conf

## 6.612 Digikam::FaceScanWidget Class Reference

Inheritance diagram for Digikam::FaceScanWidget:



### Classes

- class [Private](#)

### Public Member Functions

- **FaceScanWidget** (`QWidget *const parent=nullptr`)
- [FaceScanSettings](#) **settings** () const
- bool **settingsConflicted** () const

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Protected Member Functions

- void [doLoadState](#) () override
- void [doSaveState](#) () override

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### 6.612.1 Member Function Documentation

#### 6.612.1.1 doLoadState()

```
void Digikam::FaceScanWidget::doLoadState ( ) [override], [protected], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation. ClearAll isn't a valid value anymore so set it Rescan. ClearAll is only used by ResetFacesDb in maintenance.

Implements [Digikam::StateSavingObject](#).

#### 6.612.1.2 doSaveState()

```
void Digikam::FaceScanWidget::doSaveState ( ) [override], [protected], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

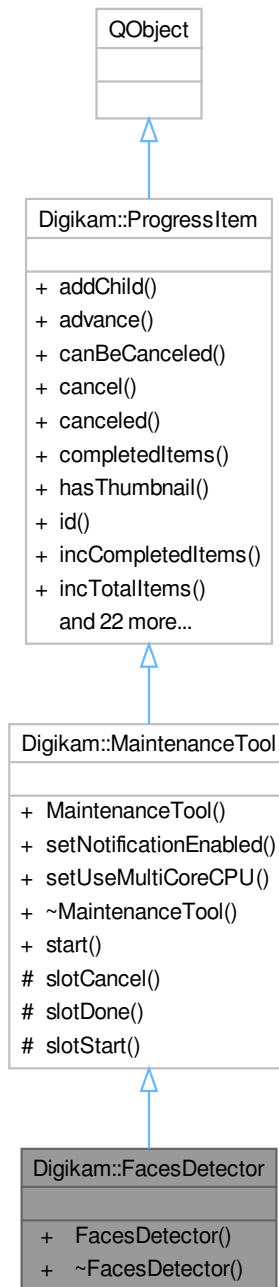
## 6.613 Digikam::FaceScanWidget::Private Class Reference

### Public Attributes

- [AlbumSelectors](#) \* **albumSelectors** = nullptr
- [SqueezedComboBox](#) \* **alreadyScannedBox** = nullptr
- const QString **configAlreadyScannedHandling** = QLatin1String("Already Scanned Handling")
- const QString **configMainTask** = QLatin1String("Face Scan Main Task")
- const QString **configName** = QLatin1String("Face Management Settings")
- const QString **configUseFullCpu** = QLatin1String("Use Full CPU")
- const QString **configValueDetect** = QLatin1String("Detect")
- const QString **configValueDetectAndRecognize** = QLatin1String("Detect and Recognize Faces")
- const QString **configValueRecognizedMarkedFaces** = QLatin1String("Recognize Marked Faces")
- [DIntNumInput](#) \* **detectAccuracyInput** = nullptr
- [QRadioButton](#) \* **detectAndRecognizeButton** = nullptr
- [QRadioButton](#) \* **detectButton** = nullptr
- [SqueezedComboBox](#) \* **detectModelBox** = nullptr
- [SqueezedComboBox](#) \* **detectSizeBox** = nullptr
- [FacesEngine](#) \* **facesDetector** = nullptr
- [QPushButton](#) \* **helpButton** = nullptr
- [DIntNumInput](#) \* **recognizeAccuracyInput** = nullptr
- [SqueezedComboBox](#) \* **recognizeModelBox** = nullptr
- [QRadioButton](#) \* **reRecognizeButton** = nullptr
- bool **settingsConflicted** = false
- [QWidget](#) \* **settingsTab** = nullptr
- [QTabWidget](#) \* **tabWidget** = nullptr
- [QCheckBox](#) \* **useFullCpuButton** = nullptr
- [QWidget](#) \* **workflowWidget** = nullptr

## 6.614 Digikam::FacesDetector Class Reference

Inheritance diagram for Digikam::FacesDetector:



### Public Types

- enum **InputSource** { **Albums** = 0 , **Infos** , **Ids** }

## Signals

- void **signalScanNotification** (const QString &msg, int type)

## Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

## Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- **FacesDetector** (const [FaceScanSettings](#) &settings, [ProgressItem](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- **MaintenanceTool** (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)
- virtual void [setUseMultiCoreCPU](#) (bool)

## Public Member Functions inherited from Digikam::ProgressItem

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool **advance** (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool **canBeCanceled** () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool **hasThumbnail** () const
- const QString & **id** () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & **label** () const
- [ProgressItem](#) \* **parent** () const
- unsigned int **progress** () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool canBeCanceled, bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)
- void **setProgress** (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void **setShowAtStart** (bool showAtStart)
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void **setStatus** (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void **setThumbnail** (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void **setUsesBusyIndicator** (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool **showAtStart** () const
- const QString & **status** () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool **usesBusyIndicator** () const

## Additional Inherited Members

## Public Slots inherited from Digikam::MaintenanceTool

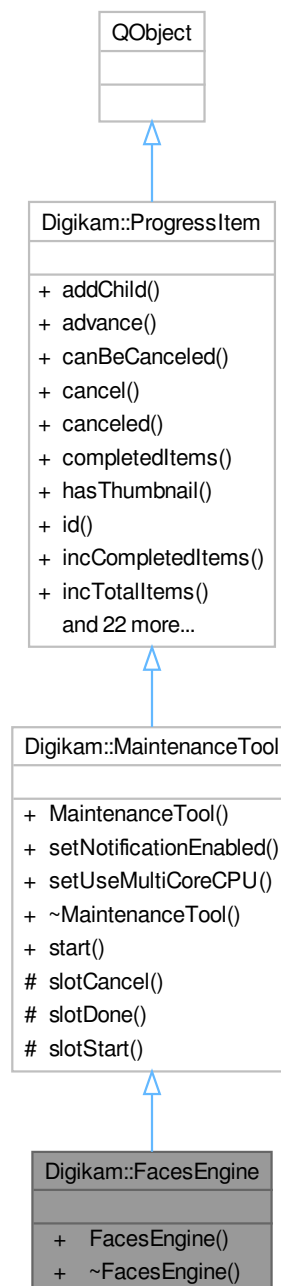
- void **start** ()

## Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.615 Digikam::FacesEngine Class Reference

Inheritance diagram for Digikam::FacesEngine:





## Public Types

- enum **InputSource** { **Albums** = 0 , **Infos** , **Ids** }

## Signals

- void **signalScanNotification** (const QString &msg, int type)

## Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

## Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- **FacesEngine** (const [FaceScanSettings](#) &settings, [ProgressItem](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- **MaintenanceTool** (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)
- virtual void [setUseMultiCoreCPU](#) (bool)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool **advance** (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool **canBeCanceled** () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool **hasThumbnail** () const
- const QString & **id** () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & **label** () const
- [ProgressItem](#) \* **parent** () const
- unsigned int **progress** () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool canBeCanceled, bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)
- void **setProgress** (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void **setShowAtStart** (bool showAtStart)
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void **setStatus** (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void **setThumbnail** (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void **setUsesBusyIndicator** (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool **showAtStart** () const
- const QString & **status** () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool **usesBusyIndicator** () const

## Additional Inherited Members

## Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

## Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.616 Digikam::FaceTags Class Reference

### Static Public Member Functions

- static QList< QString > [allPersonNames](#) ()
- static QList< QString > [allPersonPaths](#) ()
- static QList< int > [allPersonTags](#) ()
- static void [applyTagIdentityMapping](#) (int tagId, const QMap< QString, QString > &attributes)
- static void [ensureIsPerson](#) (int tagId, const QString &fullName=QString())
- static QString [faceNameForTag](#) (int tagId)
- static QString [getNameForRect](#) (qulonglong imageid, const QRect &faceRect)
- static int [getOrCreateTagForIdentity](#) (const QMap< QString, QString > &attributes)
- static int [getOrCreateTagForPerson](#) (const QString &name, int parentId=-1, const QString &fullName=QString())
- static QMap< QString, QString > [identityAttributes](#) (int tagId)
- static int [ignoredPersonTagId](#) ()
- static bool [isPerson](#) (int tagId)
- static bool [isSystemPersonTagId](#) (int tagId)
- static bool [isTheIgnoredPerson](#) (int tagId)
- static bool [isTheUnconfirmedPerson](#) (int tagId)
- static bool [isTheUnknownPerson](#) (int tagId)
- static int [personParentTag](#) ()
- static int [scannedForFacesTagId](#) ()
- static int [tagForPerson](#) (const QString &name, int parentId=-1, const QString &fullName=QString())
- static int [unconfirmedPersonTagId](#) ()
- static int [unknownPersonTagId](#) ()

### 6.616.1 Member Function Documentation

#### 6.616.1.1 allPersonNames()

```
QList< QString > Digikam::FaceTags::allPersonNames ( ) [static]
```

A method to return a list of all person tag names in the DB.

#### 6.616.1.2 allPersonPaths()

```
QList< QString > Digikam::FaceTags::allPersonPaths ( ) [static]
```

A method to return a list of all person tag paths in the DB.

### 6.616.1.3 allPersonTags()

```
QList< int > Digikam::FaceTags::allPersonTags ( ) [static]
```

A method to return a list of all person tags in the DB.

### 6.616.1.4 applyTagIdentityMapping()

```
void Digikam::FaceTags::applyTagIdentityMapping (
    int tagId,
    const QMap< QString, QString > & attributes ) [static]
```

Map an existing tag to a [FacesEngine Identity](#). Subsequently, the [Identity](#) can be retrieved via the `identityAttributes()`.

### 6.616.1.5 ensureIsPerson()

```
void Digikam::FaceTags::ensureIsPerson (
    int tagId,
    const QString & fullName = QString() ) [static]
```

Ensure that the given tag is a person tag. If not, it will be converted. Optionally, pass the full name. (tag name is not changed).

### 6.616.1.6 faceNameForTag()

```
QString Digikam::FaceTags::faceNameForTag (
    int tagId ) [static]
```

Return a person's name for a tag.

### 6.616.1.7 getOrCreateTagForIdentity()

```
int Digikam::FaceTags::getOrCreateTagForIdentity (
    const QMap< QString, QString > & attributes ) [static]
```

Use attributes as used by [FacesEngine](#) to identify or create a person tag; From the database, produce the identity attributes identifying the corresponding identity.

### 6.616.1.8 getOrCreateTagForPerson()

```
int Digikam::FaceTags::getOrCreateTagForPerson (
    const QString & name,
    int parentId = -1,
    const QString & fullName = QString() ) [static]
```

First, looks for the given person name in person tags, and returns an ID. If not, creates a new tag. Per default, `fullName` is the same as `name`.

### 6.616.1.9 isPerson()

```
bool Digikam::FaceTags::isPerson (
    int tagId ) [static]
```

Returns a boolean value indicating whether the given tagId represents a person.

### 6.616.1.10 personParentTag()

```
int Digikam::FaceTags::personParentTag ( ) [static]
```

The suggested parent tag for persons.

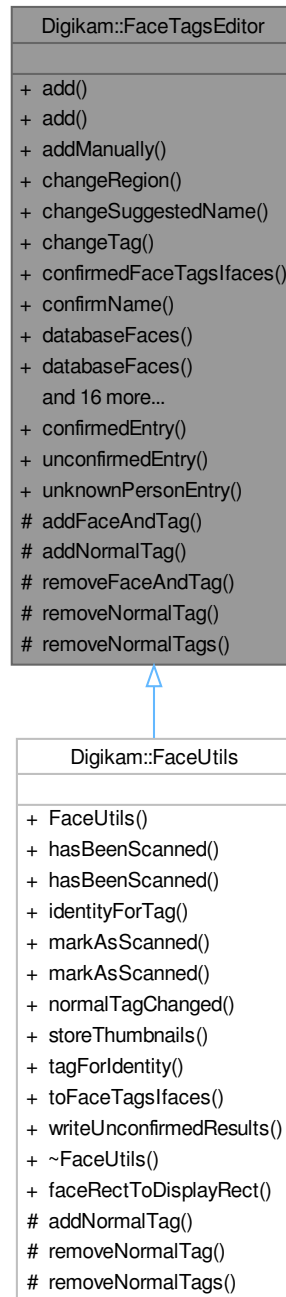
### 6.616.1.11 tagForPerson()

```
int Digikam::FaceTags::tagForPerson (
    const QString & name,
    int parentId = -1,
    const QString & fullName = QString() ) [static]
```

Looks for the given person name under the People tags tree, and returns an ID. Returns 0 if no name found. Per default, fullName is the same as name. As parentId of -1 signals to look for any tag, a valid parentId will limit the search to direct children of this tag. parentId of 0 means top-level tag.

## 6.617 Digikam::FaceTagsEditor Class Reference

Inheritance diagram for Digikam::FaceTagsEditor:



### Public Member Functions

- void `add` (const [FaceTagsiface](#) &face, bool trainFace=true)
- [FaceTagsiface](#) `add` (qlonglong imageid, int tagId, const [TagRegion](#) &region, bool trainFace=true)

- [FaceTagsIface](#) **addManually** (const [FaceTagsIface](#) &face)
- [FaceTagsIface](#) **changeRegion** (const [FaceTagsIface](#) &face, const [TagRegion](#) &newRegion)
- [FaceTagsIface](#) **changeSuggestedName** (const [FaceTagsIface](#) &previousEntry, int unconfirmedNameTagId)
- [FaceTagsIface](#) **changeTag** (const [FaceTagsIface](#) &face, int newTagId)
- [QList](#)< [FaceTagsIface](#) > **confirmedFaceTagsIfaces** (qulonglong imageid) const
- [FaceTagsIface](#) **confirmName** (const [FaceTagsIface](#) &face, int tagId=-1, const [TagRegion](#) &confirmedRegion=[TagRegion](#)())
- [QList](#)< [FaceTagsIface](#) > **databaseFaces** (qulonglong imageid) const
- [QList](#)< [FaceTagsIface](#) > **databaseFaces** (qulonglong imageid, [FaceTagsIface::TypeFlags](#) flags) const
- [QList](#)< [FaceTagsIface](#) > **databaseFacesForTraining** (qulonglong imageid) const
- int **faceCountForPersonInImage** (qulonglong imageid, int tagId) const
- [QList](#)< [ItemTagPair](#) > **faceItemTagPairs** (qulonglong imageid, [FaceTagsIface::TypeFlags](#) flags) const
- [QMap](#)< [QString](#), [QString](#) > **getSuggestedNames** (qulonglong id) const
- [QList](#)< [QRect](#) > **getTagRects** (qulonglong imageid) const
- [QList](#)< [FaceTagsIface](#) > **ignoredFaceTagsIfaces** (qulonglong imageid) const
- int **numberOfFaces** (qulonglong imageid) const
- void **removeAllFaces** (qulonglong imageid)
- void **removeFace** (const [FaceTagsIface](#) &face, bool touchTags=true)
- void **removeFace** (qulonglong imageid, const [QRect](#) &rect)
- void **removeFaces** (const [QList](#)< [FaceTagsIface](#) > &faces)
- bool **rotateFaces** (qulonglong imageid, const [QSize](#) &size, int oldOrientation, int newOrientation)
- [QList](#)< [FaceTagsIface](#) > **unconfirmedFaceTagsIfaces** (qulonglong imageid) const
- [QList](#)< [FaceTagsIface](#) > **unconfirmedNameFaceTagsIfaces** (qulonglong imageid) const

### Static Public Member Functions

- static [FaceTagsIface](#) **confirmedEntry** (const [FaceTagsIface](#) &face, int tagId=-1, const [TagRegion](#) &confirmedRegion=[TagRegion](#)())
- static [FaceTagsIface](#) **unconfirmedEntry** (qulonglong imageid, int tagId, const [TagRegion](#) &region)
- static [FaceTagsIface](#) **unknownPersonEntry** (qulonglong imageid, const [TagRegion](#) &region)

### Protected Member Functions

- void **addFaceAndTag** ([ItemTagPair](#) &pair, const [FaceTagsIface](#) &face, const [QStringList](#) &properties, bool addTag)
- virtual void **addNormalTag** (qulonglong imageid, int tagId)
- void **removeFaceAndTag** ([ItemTagPair](#) &pair, const [FaceTagsIface](#) &face, bool touchTags)
- virtual void **removeNormalTag** (qulonglong imageid, int tagId)
- virtual void **removeNormalTags** (qulonglong imageid, const [QList](#)< int > &tagIds)

## 6.617.1 Member Function Documentation

### 6.617.1.1 add()

```
void Digikam::FaceTagsEditor::add (
    const FaceTagsIface & face,
    bool trainFace = true )
```

Adds a new entry to the database. The convenience wrapper will return the newly created entry. If trainFace is true, the face will also be listed in the db as needing training. The tag of the face will, if necessary, be converted to a person tag.

### 6.617.1.2 addNormalTag()

```
void Digikam::FaceTagsEditor::addNormalTag (
    qlonglong imageId,
    int tagId ) [protected], [virtual]
```

Reimplemented in [Digikam::FaceUtils](#).

### 6.617.1.3 changeRegion()

```
FaceTagsIface Digikam::FaceTagsEditor::changeRegion (
    const FaceTagsIface & face,
    const TagRegion & newRegion )
```

Changes the region of the given entry. Returns the face with the new region set.

### 6.617.1.4 changeSuggestedName()

```
FaceTagsIface Digikam::FaceTagsEditor::changeSuggestedName (
    const FaceTagsIface & previousEntry,
    int unconfirmedNameTagId )
```

Switches an unknownPersonEntry or unconfirmedEntry to an unconfirmedEntry (with a different suggested name).

### 6.617.1.5 changeTag()

```
FaceTagsIface Digikam::FaceTagsEditor::changeTag (
    const FaceTagsIface & face,
    int newTagId )
```

Changes the tag of the given entry. Returns the face with the new Tag. Since a new Tag is going to be assigned to the Face, it's important to remove the association between the face and the old tagId.

If the face is being ignored and it was an unconfirmed or unknown face don't remove a possible tag. See bug 449142.

We store metadata of [FaceTags](#), if it's a confirmed person.

### 6.617.1.6 confirmedEntry()

```
FaceTagsIface Digikam::FaceTagsEditor::confirmedEntry (
    const FaceTagsIface & face,
    int tagId = -1,
    const TagRegion & confirmedRegion = TagRegion() ) [static]
```

Returns the entry that would be added if the given face is confirmed.



### 6.617.1.7 confirmName()

```
FaceTagsIface Digikam::FaceTagsEditor::confirmName (
    const FaceTagsIface & face,
    int tagId = -1,
    const TagRegion & confirmedRegion = TagRegion() )
```

Assign the name tag for given face entry. Pass the tagId if it changed or was newly assigned (UnknownName). Pass the new, corrected region if it changed. If the default values are passed, tag id or region are taken from the given face. The given face should be an unchanged entry read from the database. The confirmed tag will, if necessary, be converted to a person tag. Returns the newly inserted entry.

### 6.617.1.8 databaseFaces()

```
QList< FaceTagsIface > Digikam::FaceTagsEditor::databaseFaces (
    qlonglong imageid ) const
```

Reads the FaceTagsIfaces for the given image id from the database.

### 6.617.1.9 faceCountForPersonInImage()

```
int Digikam::FaceTagsEditor::faceCountForPersonInImage (
    qlonglong imageid,
    int tagId ) const
```

Returns the number of faces a particular person has in the specified image.

### 6.617.1.10 getSuggestedNames()

```
QMap< QString, QString > Digikam::FaceTagsEditor::getSuggestedNames (
    qlonglong id ) const
```

Returns a Map of Tag Regions (in XML format) to Suggested Name (from Face Recognition) for the given image. This function makes read operations to the database, and hence can be inefficient when called repeatedly. A cached version is provided in [ItemInfo](#), and should be preferred for intensive operations such as sorting, categorizing etc. For Unconfirmed Results, the value is stored as a tuple of (SuggestedId, Property, Region). Look at the digikam.db file for more details.

### 6.617.1.11 getTagRects()

```
QList< QRect > Digikam::FaceTagsEditor::getTagRects (
    qlonglong imageid ) const
```

Returns a list of all tag rectangles for the image. Unlike findAndTagFaces, this does not take a [DImg](#), because it returns only a QRect, not a Face, so no need of cropping a face rectangle.

### 6.617.1.12 numberOfFaces()

```
int Digikam::FaceTagsEditor::numberOfFaces (
    qlonglong imageid ) const
```

Returns the number of faces present in an image.

### 6.617.1.13 removeAllFaces()

```
void Digikam::FaceTagsEditor::removeAllFaces (
    qlonglong imageid )
```

Unassigns all face tags from the image and sets it's scanned property to false.

### 6.617.1.14 removeFace() [1/2]

```
void Digikam::FaceTagsEditor::removeFace (
    const FaceTagsIface & face,
    bool touchTags = true )
```

Remove the given face. If appropriate, the tag is also removed.

### 6.617.1.15 removeFace() [2/2]

```
void Digikam::FaceTagsEditor::removeFace (
    qlonglong imageid,
    const QRect & rect )
```

Remove a face or the face for a certain rect from an image.

### 6.617.1.16 removeNormalTag()

```
void Digikam::FaceTagsEditor::removeNormalTag (
    qlonglong imageId,
    int tagId ) [protected], [virtual]
```

Reimplemented in [Digikam::FaceUtils](#).

### 6.617.1.17 rotateFaces()

```
bool Digikam::FaceTagsEditor::rotateFaces (
    qlonglong imageId,
    const QSize & size,
    int oldOrientation,
    int newOrientation )
```

Rotate face tags.

### 6.617.1.18 unconfirmedEntry()

```
FaceTagsIface Digikam::FaceTagsEditor::unconfirmedEntry (
    qlonglong imageId,
    int tagId,
    const TagRegion & region ) [static]
```

Returns the entry that would be added if the given face is autodetected. If tagId is -1, the unknown person will be taken.

### 6.617.1.19 unconfirmedFaceTagsIfaces()

```
QList< FaceTagsIface > Digikam::FaceTagsEditor::unconfirmedFaceTagsIfaces (
    qlonglong imageid ) const
```

Returns list of Unconfirmed and Unknown faces in the Image. If you want just Unconfirmed Faces,

See also

[unconfirmedNameFaceTagsIfaces](#).

### 6.617.1.20 unconfirmedNameFaceTagsIfaces()

```
QList< FaceTagsIface > Digikam::FaceTagsEditor::unconfirmedNameFaceTagsIfaces (
    qlonglong imageid ) const
```

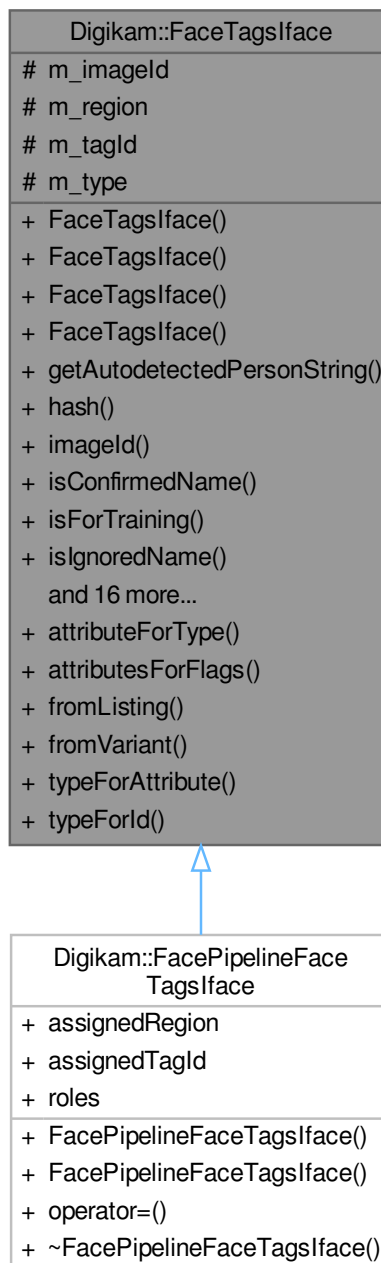
Returns a list of UnconfirmedFaces in the Image. Different from

See also

[unconfirmedFaceTagsIfaces.](#)

## 6.618 Digikam::FaceTagsIface Class Reference

Inheritance diagram for Digikam::FaceTagsIface:



## Public Types

- enum **Type** {  
**InvalidFace** = 0 , **UnknownName** = 1 << 0 , **UnconfirmedName** = 1 << 1 , **IgnoredName** = 1 << 2 ,  
**ConfirmedName** = 1 << 3 , **FaceForTraining** = 1 << 4 , **UnconfirmedTypes** = UnknownName |  
UnconfirmedName , **NormalFaces** = UnknownName | UnconfirmedName | IgnoredName | ConfirmedName  
,  
**AllTypes** = UnknownName | UnconfirmedName | IgnoredName | ConfirmedName | FaceForTraining , **Type**↔  
**First** = UnknownName , **TypeLast** = FaceForTraining }

## Public Member Functions

- **FaceTagsIface** (const [FaceTagsIface](#) &other)
- **FaceTagsIface** (const QString &attribute, qlonglong imageId, int tagId, const [TagRegion](#) &region)
- **FaceTagsIface** (Type type, qlonglong imageId, int tagId, const [TagRegion](#) &region)
- QString [getAutodetectedPersonString](#) () const
- const QString [hash](#) () const
- qlonglong [imageId](#) () const
- bool [isConfirmedName](#) () const
- bool [isForTraining](#) () const
- bool [isIgnoredName](#) () const
- bool [isInvalidFace](#) () const
- bool [isNull](#) () const
- bool [isUnconfirmedName](#) () const
- bool [isUnconfirmedType](#) () const
- bool [isUnknownName](#) () const
- [FaceTagsIface](#) & **operator=** (const [FaceTagsIface](#) &other)
- bool **operator==** (const [FaceTagsIface](#) &other) const
- [TagRegion](#) [region](#) () const
- void [removeFaceTraining](#) () const
- void [setRegion](#) (const [TagRegion](#) &region)
- void [setTagId](#) (int tagId)
- void [setType](#) (Type type)
- int [tagId](#) () const
- QVariant [toVariant](#) () const
- Type [type](#) () const

## Static Public Member Functions

- static QString [attributeForType](#) (Type type)
- static QStringList [attributesForFlags](#) (TypeFlags flags)
- static [FaceTagsIface](#) [fromListing](#) (qlonglong imageId, const QList< QVariant > &values)
- static [FaceTagsIface](#) [fromVariant](#) (const QVariant &var)
- static Type [typeForAttribute](#) (const QString &attribute, int tagId=0)
- static Type [typeForId](#) (int tagId)

## Protected Attributes

- qlonglong [m\\_imageId](#) = 0
- [TagRegion](#) [m\\_region](#)
- int [m\\_tagId](#) = 0
- Type [m\\_type](#) = InvalidFace

## 6.618.1 Member Function Documentation

### 6.618.1.1 attributeForType()

```
QString Digikam::FaceTagsIface::attributeForType (
    Type type ) [static]
```

Return the corresponding image tag property for the given type.

### 6.618.1.2 attributesForFlags()

```
QStringList Digikam::FaceTagsIface::attributesForFlags (
    TypeFlags flags ) [static]
```

Returns a list of all image tag properties for which flags are set.

### 6.618.1.3 fromListing()

```
FaceTagsIface Digikam::FaceTagsIface::fromListing (
    qlonglong imageid,
    const QList< QVariant > & values ) [static]
```

Create a [FaceTagsIface](#) from the extraValues returned from [ItemLister](#).

### 6.618.1.4 fromVariant()

```
FaceTagsIface Digikam::FaceTagsIface::fromVariant (
    const QVariant & var ) [static]
```

Writes the contents of this face - in a compact way - in the QVariant. Only native QVariant types are used, that is, the QVariant will not have a custom type, thus it can be compared by value by operator==.

### 6.618.1.5 getAutodetectedPersonString()

```
QString Digikam::FaceTagsIface::getAutodetectedPersonString ( ) const
```

Returns the string tagId + ',' + unconfirmedFace + ',' + regionXml.

### 6.618.1.6 hash()

```
const QString Digikam::FaceTagsIface::hash ( ) const
```

Generate a hash based on the imageId, tagId, and rect to uniquely identify this entry in the face training DB.

### 6.618.1.7 removeFaceTraining()

```
void Digikam::FaceTagsIface::removeFaceTraining ( ) const
```

Remove the face from face training based on the current imageId, tagId, and rect hash.

### 6.618.1.8 typeForAttribute()

```
FaceTagsIface::Type Digikam::FaceTagsIface::typeForAttribute (
    const QString & attribute,
    int tagId = 0 ) [static]
```

Return the Type for the given attribute. To distinguish between UnknownName and UnconfirmedName, the tagId must be given.

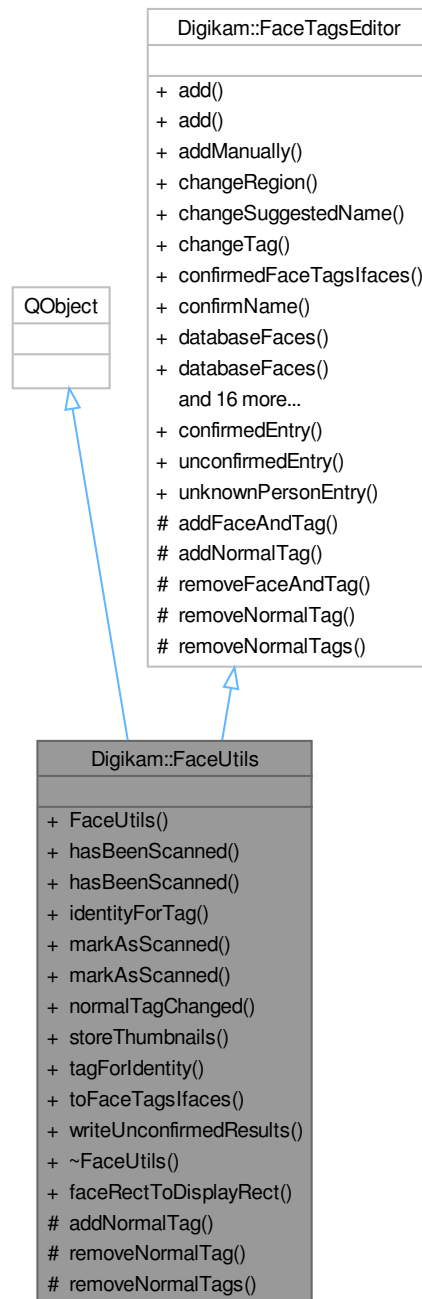
### 6.618.1.9 typeForId()

```
FaceTagsIface::Type Digikam::FaceTagsIface::typeForId (
    int tagId ) [static]
```

Returns the Face Type corresponding to the given TagId.

## 6.619 Digikam::FaceUtils Class Reference

Inheritance diagram for Digikam::FaceUtils:



### Public Types

- enum **FaceRecognitionSteps** { **DetectFaceRegions** , **DetectAndRecognize** }



## Public Member Functions

- **FaceUtils** (QObject \*const parent=nullptr)
- bool **hasBeenScanned** (const [ItemInfo](#) &info) const
- bool **hasBeenScanned** (qulonglong imageid) const
- [Identity](#) **identityForTag** (int tagId) const
- void **markAsScanned** (const [ItemInfo](#) &info, bool [hasBeenScanned](#)=true) const
- void **markAsScanned** (qulonglong imageid, bool [hasBeenScanned](#)=true) const
- bool **normalTagChanged** () const
- void **storeThumbnails** ([ThumbnailLoadThread](#) \*const thread, const QString &filePath, const QList<[FaceTagsIface](#)> &[databaseFaces](#), const [DImg](#) &image)
- int **tagForIdentity** (const [Identity](#) &identity) const
- QList<[FaceTagsIface](#)> **toFaceTagsIfaces** (qulonglong imageid, const QList< [QRectF](#) > &detectedFaces, const QList< [Identity](#) > &recognitionResults, const QSize &fullSize) const
- QList<[FaceTagsIface](#)> **writeUnconfirmedResults** (qulonglong imageid, const QList< [QRectF](#) > &detectedFaces, const QList< [Identity](#) > &recognitionResults, const QSize &fullSize)

## Public Member Functions inherited from [Digikam::FaceTagsEditor](#)

- void **add** (const [FaceTagsIface](#) &face, bool trainFace=true)
- [FaceTagsIface](#) **add** (qulonglong imageid, int tagId, const [TagRegion](#) &region, bool trainFace=true)
- [FaceTagsIface](#) **addManually** (const [FaceTagsIface](#) &face)
- [FaceTagsIface](#) **changeRegion** (const [FaceTagsIface](#) &face, const [TagRegion](#) &newRegion)
- [FaceTagsIface](#) **changeSuggestedName** (const [FaceTagsIface](#) &previousEntry, int unconfirmedNameTagId)
- [FaceTagsIface](#) **changeTag** (const [FaceTagsIface](#) &face, int newTagId)
- QList<[FaceTagsIface](#)> **confirmedFaceTagsIfaces** (qulonglong imageid) const
- [FaceTagsIface](#) **confirmName** (const [FaceTagsIface](#) &face, int tagId=-1, const [TagRegion](#) &confirmedRegion=[TagRegion](#)())
- QList<[FaceTagsIface](#)> **databaseFaces** (qulonglong imageid) const
- QList<[FaceTagsIface](#)> **databaseFaces** (qulonglong imageid, [FaceTagsIface::TypeFlags](#) flags) const
- QList<[FaceTagsIface](#)> **databaseFacesForTraining** (qulonglong imageid) const
- int **faceCountForPersonInImage** (qulonglong imageid, int tagId) const
- QList<[ItemTagPair](#)> **facetItemTagPairs** (qulonglong imageid, [FaceTagsIface::TypeFlags](#) flags) const
- QMap< QString, QString > **getSuggestedNames** (qulonglong id) const
- QList< [QRect](#) > **getTagRects** (qulonglong imageid) const
- QList<[FaceTagsIface](#)> **ignoredFaceTagsIfaces** (qulonglong imageid) const
- int **numberOfFaces** (qulonglong imageid) const
- void **removeAllFaces** (qulonglong imageid)
- void **removeFace** (const [FaceTagsIface](#) &face, bool touchTags=true)
- void **removeFace** (qulonglong imageid, const [QRect](#) &rect)
- void **removeFaces** (const QList<[FaceTagsIface](#)> &faces)
- bool **rotateFaces** (qulonglong imageid, const QSize &size, int oldOrientation, int newOrientation)
- QList<[FaceTagsIface](#)> **unconfirmedFaceTagsIfaces** (qulonglong imageid) const
- QList<[FaceTagsIface](#)> **unconfirmedNameFaceTagsIfaces** (qulonglong imageid) const

## Static Public Member Functions

- static [QRect](#) **faceRectToDisplayRect** (const [QRect](#) &rect)

## Static Public Member Functions inherited from [Digikam::FaceTagsEditor](#)

- static [FaceTagsIface](#) **confirmedEntry** (const [FaceTagsIface](#) &face, int tagId=-1, const [TagRegion](#) &confirmedRegion=[TagRegion](#)())
- static [FaceTagsIface](#) **unconfirmedEntry** (qulonglong imageid, int tagId, const [TagRegion](#) &region)
- static [FaceTagsIface](#) **unknownPersonEntry** (qulonglong imageid, const [TagRegion](#) &region)

## Protected Member Functions

- void [addNormalTag](#) (qulonglong imageId, int tagId) override
- void [removeNormalTag](#) (qulonglong imageId, int tagId) override
- void [removeNormalTags](#) (qulonglong imageId, const QList< int > &tagIds) override

## Protected Member Functions inherited from [Digikam::FaceTagsEditor](#)

- void [addFaceAndTag](#) ([ItemTagPair](#) &pair, const [FaceTagsIface](#) &face, const QStringList &properties, bool addTag)
- void [removeFaceAndTag](#) ([ItemTagPair](#) &pair, const [FaceTagsIface](#) &face, bool touchTags)

## 6.619.1 Member Function Documentation

### 6.619.1.1 addNormalTag()

```
void Digikam::FaceUtils::addNormalTag (
    qulonglong imageId,
    int tagId ) [override], [protected], [virtual]
```

Reimplemented from parent class.

Reimplemented from [Digikam::FaceTagsEditor](#).

### 6.619.1.2 faceRectToDisplayRect()

```
QRect Digikam::FaceUtils::faceRectToDisplayRect (
    const QRect & rect ) [static]
```

For display, it may be desirable to display a slightly larger region than the strict face rectangle. This returns a pixel margin commonly used to increase the rectangle size in all four directions.

### 6.619.1.3 hasBeenScanned()

```
bool Digikam::FaceUtils::hasBeenScanned (
    const ItemInfo & info ) const
```

Tells if the image has been scanned for faces or not

### 6.619.1.4 markAsScanned()

```
void Digikam::FaceUtils::markAsScanned (
    qulonglong imageid,
    bool hasBeenScanned = true ) const
```

Marks the image as scanned for faces.

### 6.619.1.5 removeNormalTag()

```
void Digikam::FaceUtils::removeNormalTag (
    qlonglong imageId,
    int tagId ) [override], [protected], [virtual]
```

If the face just removed was the final face associated with that Tag, reset Tag Icon.

Reimplemented from [Digikam::FaceTagsEditor](#).

### 6.619.1.6 removeNormalTags()

```
void Digikam::FaceUtils::removeNormalTags (
    qlonglong imageId,
    const QList< int > & tagIds ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::FaceTagsEditor](#).

### 6.619.1.7 storeThumbnails()

```
void Digikam::FaceUtils::storeThumbnails (
    ThumbnailLoadThread *const thread,
    const QString & filePath,
    const QList< FaceTagsIface > & databaseFaces,
    const DImg & image )
```

This uses a thumbnail load thread to load the image detail. If requested, the faces will be scaled to the given (fixed) size. Store the needed thumbnails for the given faces. This can be a huge optimization when the has already been loaded anyway.

Images in faces and thumbnails.

### 6.619.1.8 toFaceTagsifaces()

```
QList< FaceTagsIface > Digikam::FaceUtils::toFaceTagsIfaces (
    qlonglong imageid,
    const QList< QRectF > & detectedFaces,
    const QList< Identity > & recognitionResults,
    const QSize & fullSize ) const
```

Conversion.

Convert between [FacesEngine](#) results and [FaceTagsIface](#).

### 6.619.1.9 writeUnconfirmedResults()

```
QList< FaceTagsIface > Digikam::FaceUtils::writeUnconfirmedResults (
    qlonglong imageid,
    const QList< QRectF > & detectedFaces,
    const QList< Identity > & recognitionResults,
    const QSize & fullSize )
```

The given face list is a result of automatic detection and possibly recognition. The results are written to the database and merged with existing entries. The returned list contains the faces written to the database and has the same size as the given list. If a face was skipped (because of an existing entry), a null [FaceTagsIface](#) will be at this place.

## 6.620 Digikam::FacialRecognitionWrapper Class Reference

### Classes

- class [Private](#)

### Public Member Functions

- [Identity](#) [addIdentity](#) (const QMap< QString, QString > &attributes)
- void [addIdentityAttribute](#) (int id, const QString &attribute, const QString &value)
- void [addIdentityAttributes](#) (int id, const QMap< QString, QString > &attributes)
- [Identity](#) [addIdentityDebug](#) (const QMap< QString, QString > &attributes)
- QList< [Identity](#) > [allIdentities](#) () const  
*NOTE: For the documentation of standard attributes, see identity.h.*
- void [clearAllTraining](#) ()
- void [clearTraining](#) (const QList< [Identity](#) > &identitiesToClean)
- void [clearTraining](#) (const QString &hash)
- void [deleteIdentities](#) (QList< [Identity](#) > identitiesToBeDeleted)
- void [deleteIdentity](#) (const [Identity](#) &identityToBeDeleted)
- [FacialRecognitionWrapper](#) (const [FacialRecognitionWrapper](#) &)
- [Identity](#) [findIdentity](#) (const QMap< QString, QString > &attributes) const
- [Identity](#) [findIdentity](#) (const QString &attribute, const QString &value) const
- [Identity](#) [identity](#) (int id) const
- bool [integrityCheck](#) ()
- QVariantMap [parameters](#) () const
- [Identity](#) [recognizeFace](#) (QImage \*const image)
- QList< [Identity](#) > [recognizeFaces](#) (const QList< QImage \* > &images)
- QList< [Identity](#) > [recognizeFaces](#) ([ImageListProvider](#) \*const images)
- void [setIdentityAttributes](#) (int id, const QMap< QString, QString > &attributes)
- void [setParameter](#) (const QString &parameter, const QVariant &value)
- void [setParameters](#) (const [FaceScanSettings](#) &parameters)
- void [setParameters](#) (const QVariantMap &parameters)
- void [train](#) (const [Identity](#) &identityToBeTrained, const QList< QPair< QImage \*, QString > > &images)
- void [train](#) (const [Identity](#) &identityToBeTrained, const QPair< QImage \*, QString > &image)
- void [train](#) (const [Identity](#) &identityToBeTrained, [TrainingDataProvider](#) \*const data)
- void [train](#) (const QList< [Identity](#) > &identitiesToBeTrained, [TrainingDataProvider](#) \*const data)
- void [vacuum](#) ()

### 6.620.1 Member Function Documentation

#### 6.620.1.1 addIdentity()

```
Identity Digikam::FacialRecognitionWrapper::addIdentity (
    const QMap< QString, QString > & attributes )
```

Adds a new identity with the specified attributes. Please note that a UUID is automatically generated.

### 6.620.1.2 addIdentityAttributes()

```
void Digikam::FacialRecognitionWrapper::addIdentityAttributes (
    int id,
    const QMap< QString, QString > & attributes )
```

Adds or sets, resp., the attributes of an identity.

### 6.620.1.3 addIdentityDebug()

```
Identity Digikam::FacialRecognitionWrapper::addIdentityDebug (
    const QMap< QString, QString > & attributes )
```

This is the debug version of addIdentity, so the identity is only added to identityCache, but not into the recognition database.

### 6.620.1.4 allIdentities()

```
QList< Identity > Digikam::FacialRecognitionWrapper::allIdentities ( ) const
```

Returns all identities known to the database

### 6.620.1.5 clearAllTraining()

```
void Digikam::FacialRecognitionWrapper::clearAllTraining ( )
```

Deletes the training data for all identities, leaving the identities as such in the database.

### 6.620.1.6 clearTraining() [1/2]

```
void Digikam::FacialRecognitionWrapper::clearTraining (
    const QList< Identity > & identitiesToClean )
```

Deletes the training data for the given identity, leaving the identity as such in the database.

### 6.620.1.7 clearTraining() [2/2]

```
void Digikam::FacialRecognitionWrapper::clearTraining (
    const QString & hash )
```

Deletes the training image for the given identity, leaving the identity as such in the database.

### 6.620.1.8 deleteIdentities()

```
void Digikam::FacialRecognitionWrapper::deleteIdentities (
    QList< Identity > identitiesToBeDeleted )
```

Deletes a list of identities from the database.

**6.620.1.9 deleteIdentity()**

```
void Digikam::FacialRecognitionWrapper::deleteIdentity (
    const Identity & identityToBeDeleted )
```

Deletes an identity from the database.

**6.620.1.10 findIdentity() [1/2]**

```
Identity Digikam::FacialRecognitionWrapper::findIdentity (
    const QMap< QString, QString > & attributes ) const
```

Finds the identity matching the given attributes. Attributes are first checked with knowledge of their meaning. Secondly, all unknown attributes are used. Returns a null [Identity](#) if no match is possible or the map is empty.

**6.620.1.11 findIdentity() [2/2]**

```
Identity Digikam::FacialRecognitionWrapper::findIdentity (
    const QString & attribute,
    const QString & value ) const
```

Finds the first identity with matching attribute - value. Returns a null identity if no match is found or attribute is empty.

**6.620.1.12 integrityCheck()**

```
bool Digikam::FacialRecognitionWrapper::integrityCheck ( )
```

Checks the integrity and returns true if everything is fine.

**6.620.1.13 recognizeFaces()**

```
QList< Identity > Digikam::FacialRecognitionWrapper::recognizeFaces (
    ImageListProvider *const images )
```

Returns the recommended size if you want to scale face images for recognition. Larger images can be passed, but may be downscaled. Performs recognition. The face details to be recognized are passed by the provider. For each entry in the provider, in 1-to-1 mapping, a recognized identity or the null identity is returned.

**6.620.1.14 setParameter()**

```
void Digikam::FacialRecognitionWrapper::setParameter (
    const QString & parameter,
    const QVariant & value )
```

Tunes backend parameters. Available parameters: "accuracy", synonymous: "threshold", range: 0-1, type: float Determines recognition threshold, 0->accept very insecure recognitions, 1-> be very sure about a recognition.

"k-nearest" : limit the number of nearest neighbors for KNN "recognizeModel" : sets the recognizer model used to instantiate the correct recognizer

### 6.620.1.15 train() [1/2]

```
void Digikam::FacialRecognitionWrapper::train (
    const Identity & identityToBeTrained,
    const QPair< QImage *, QString > & image )
```

Performs training by using image data directly.

These are convenience functions for simple setups. If you want good performance and/or a more versatile implementation, be sure to implement your own [TrainingDataProvider](#) and use one of the above functions.

### 6.620.1.16 train() [2/2]

```
void Digikam::FacialRecognitionWrapper::train (
    const QList< Identity > & identitiesToBeTrained,
    TrainingDataProvider *const data )
```

Performs training. The identities which have new images to be trained are given. An empty list means that all identities are checked.

All needed data will be queried from the provider.

An identifier for the current training context is given, which can identify the application or group of collections. (It is assumed that training from different contexts is based on non-overlapping collections of images. Keep it always constant for your app.)

### 6.620.1.17 vacuum()

```
void Digikam::FacialRecognitionWrapper::vacuum ( )
```

Shrinks the database.

## 6.621 Digikam::FacialRecognitionWrapper::Private Class Reference

### Public Member Functions

- void **applyParameters** ()
- void **clear** (const QList< int > &idsToClear)
- void **clear** (const QString &hash)
- [Identity](#) **findByAttribute** (const QString &attribute, const QString &value) const
- [Identity](#) **findByAttributes** (const QString &attribute, const QMap< QString, QString > &valueMap) const
- void **trainIdentityBatch** (const QList< [Identity](#) > &identitiesToBeTrained, [TrainingDataProvider](#) \*const data)

### Static Public Member Functions

- static bool **identityContains** (const [Identity](#) &identity, const QString &attribute, const QString &value)

## Public Attributes

- bool **dbAvailable** = false
- QHash< int, [Identity](#) > **identityCache**
- QVariantMap **parameters**
- [FaceScanSettings::FaceRecognitionModel](#) **recognizeModel** = [FaceScanSettings::FaceRecognitionModel::SFace](#)
- [OpenCVDNNFaceRecognizer](#) \* **recognizer** = nullptr
- int **ref** = 1
- QThreadPool \* **removeThreadPool** = nullptr
- QFuture< bool > **removeThreadResult**
- QReadWriteLock **trainingLock**

## 6.621.1 Member Function Documentation

### 6.621.1.1 `findByAttributes()`

```
Identity Digikam::FacialRecognitionWrapper::Private::findByAttributes (
    const QString & attribute,
    const QMap< QString, QString > & valueMap ) const
```

NOTE: Takes care that there may be multiple values of attribute in valueMap.

### 6.621.1.2 `identityContains()`

```
bool Digikam::FacialRecognitionWrapper::Private::identityContains (
    const Identity & identity,
    const QString & attribute,
    const QString & value ) [static]
```

NOTE: Takes care that there may be multiple values of attribute in identity's attributes.



## 6.622 Digikam::FFmpegBinary Class Reference

Inheritance diagram for Digikam::FFmpegBinary:



### Public Member Functions

- **FFmpegBinary** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DBinaryIface](#)

- virtual QString **baseName** () const
- virtual bool **checkDir** ()
- virtual bool **checkDirForPath** (const QString &path)
- **DBinaryIface** (const QString &binaryName, const QString &minimalVersion, const QString &header, const int headerLine, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- **DBinaryIface** (const QString &binaryName, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- const QString & **description** () const
- bool **developmentVersion** () const
- virtual QString **directory** () const
- bool **hasError** () const
- bool **isFound** () const
- bool **isValid** () const
- virtual QString **minimalVersion** () const
- virtual QString **path** () const
- virtual QString **path** (const QString &dir) const
- virtual QString **projectName** () const
- virtual bool **recheckDirectories** ()
- virtual void **setup** (const QString &prev=QString())
- virtual QUrl **url** () const
- const QString & **version** () const
- bool **versionsRight** () const
- bool **versionsRight** (const float) const

## Static Public Member Functions

- static QString **ffmpegToolBin** ()

## Static Public Member Functions inherited from [Digikam::DBinaryIface](#)

- static QString **goodBaseName** (const QString &b)

## Additional Inherited Members

## Public Slots inherited from [Digikam::DBinaryIface](#)

- virtual void **slotAddPossibleSearchDirectory** (const QString &dir)
- virtual void **slotAddSearchDirectory** (const QString &dir)
- virtual void **slotNavigateAndCheck** ()

## Signals inherited from [Digikam::DBinaryIface](#)

- void **signalBinaryValid** ()
- void **signalSearchDirectoryAdded** (const QString &dir)

## Protected Member Functions inherited from Digikam::DBinaryIface

- QString **findHeader** (const QStringList &output, const QString &header) const
- virtual bool **parseHeader** (const QString &output)
- virtual QString **readConfig** ()
- void **setVersion** (QString &version)
- virtual void **writeConfig** ()

## Protected Attributes inherited from Digikam::DBinaryIface

- const QStringList **m\_binaryArguments**
- const QString **m\_binaryBaseName**
- QLabel \* **m\_binaryLabel** = nullptr
- const bool **m\_checkVersion**
- const QString **m\_configGroup**
- QString **m\_description**
- bool **m\_developmentVersion** = false
- QLabel \* **m\_downloadButton** = nullptr
- bool **m\_hasError** = false
- const int **m\_headerLine**
- const QString **m\_headerStarts**
- bool **m\_isFound** = false
- QLineEdit \* **m\_lineEdit** = nullptr
- const QString **m\_minimalVersion**
- QPushButton \* **m\_pathButton** = nullptr
- QString **m\_pathDir** = QLatin1String("")
- QFrame \* **m\_pathWidget** = nullptr
- const QString **m\_projectName**
- QSet< QString > **m\_searchPaths**
- QLabel \* **m\_statusIcon** = nullptr
- const QUrl **m\_url**
- QString **m\_version** = QLatin1String("")
- QLabel \* **m\_versionLabel** = nullptr

## 6.623 Digikam::FFmpegConfigHelper Class Reference

### Static Public Member Functions

- static FFMpegProperties [getAudioCodecsProperties](#) ()
- static FFMpegProperties [getExtensionsProperties](#) ()
- static FFMpegProperties [getVideoCodecsProperties](#) ()

### 6.623.1 Member Function Documentation

#### 6.623.1.1 getAudioCodecsProperties()

FFMpegProperties Digikam::FFmpegConfigHelper::getAudioCodecsProperties ( ) [static]

Return a map of Audio Codec Name with a list of properties:

- Codecs description.
- Read support.
- Write support.

### 6.623.1.2 getExtensionsProperties()

```
FFMpegProperties Digikam::FFMpegConfigHelper::getExtensionsProperties ( ) [static]
```

Return a map,of File extensions supported with a list of properties:

- Format description.

### 6.623.1.3 getVideoCodecsProperties()

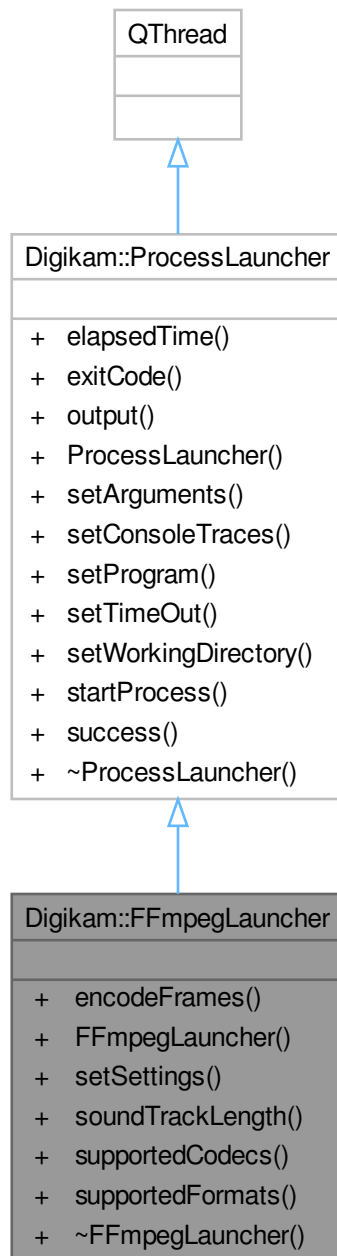
```
FFMpegProperties Digikam::FFMpegConfigHelper::getVideoCodecsProperties ( ) [static]
```

Return a map of Video Codec Name with a list of properties:

- Codecs description.
- Read support.
- Write support.

## 6.624 Digikam::FFmpegLauncher Class Reference

Inheritance diagram for Digikam::FFmpegLauncher:



### Public Member Functions

- void [encodeFrames](#) ()
- **FFmpegLauncher** (QObject \*const parent=nullptr)

- void [setSettings](#) ([VidSlideSettings](#) \*const settings)
- QTime [soundTrackLength](#) (const QString &audioPath)
- QMap< QString, QString > [supportedCodecs](#) ()
- QMap< QString, QString > [supportedFormats](#) ()

## Public Member Functions inherited from [Digikam::ProcessLauncher](#)

- qint64 [elapsedTime](#) () const
- int [exitCode](#) () const
- QString [output](#) () const
- **ProcessLauncher** (QObject \*const parent=nullptr)
- void **setArguments** (const QStringList &args)
- void [setConsoleTraces](#) (bool b)
- void **setProgram** (const QString &prog)
- void **setTimeout** (int msec)
- void **setWorkingDirectory** (const QString &dir)
- void [startProcess](#) ()
- bool [success](#) () const

## Additional Inherited Members

## Signals inherited from [Digikam::ProcessLauncher](#)

- void **signalComplete** (bool [success](#), int [exitCode](#))

## 6.624.1 Member Function Documentation

### 6.624.1.1 [encodeFrames\(\)](#)

```
void Digikam::FFmpegLauncher::encodeFrames ( )
```

Encode frames in a separated thread.

### 6.624.1.2 [setSettings\(\)](#)

```
void Digikam::FFmpegLauncher::setSettings (
    VidSlideSettings *const settings )
```

Set encoding frames settings.

### 6.624.1.3 [soundTrackLength\(\)](#)

```
QTime Digikam::FFmpegLauncher::soundTrackLength (
    const QString & audioPath )
```

Return the length of an audio file. If duration cannot be decoded, it returns a null QTime.

#### 6.624.1.4 supportedCodecs()

```
QMap< QString, QString > Digikam::FFmpegLauncher::supportedCodecs ( )
```

Get the map of supported codecs with features.

#### 6.624.1.5 supportedFormats()

```
QMap< QString, QString > Digikam::FFmpegLauncher::supportedFormats ( )
```

Get the map of supported formats with features.

## 6.625 Digikam::FieldQueryBuilder Class Reference

### Public Member Functions

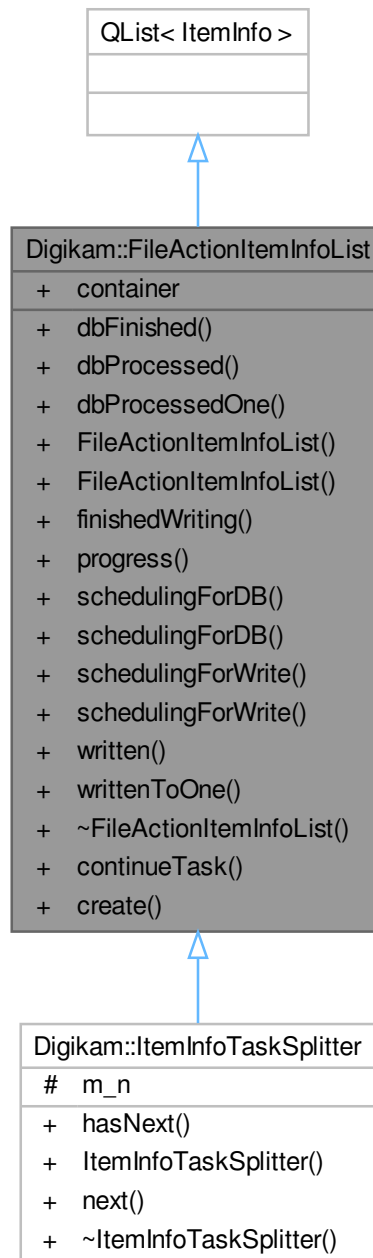
- void **addChoiceIntField** (const QString &name)
- void **addChoiceStringField** (const QString &name)
- void **addDateField** (const QString &name)
- void **addDoubleField** (const QString &name)
- void **addIntBitmaskField** (const QString &name)
- void **addIntField** (const QString &name)
- void **addLongField** (const QString &name)
- void **addLongListField** (const QString &name)
- void **addPosition** ()
- void **addRectanglePositionSearch** (double lon1, double lat1, double lon2, double lat2) const
- void **addStringField** (const QString &name)
- **FieldQueryBuilder** (QString &sql, [SearchXmlCachingReader](#) &reader, QList< QVariant > \*boundValues, [ItemQueryPostHooks](#) \*const hooks, SearchXml::Relation relation)
- QString **prepareForLike** (const QString &str) const

### Public Attributes

- QList< QVariant > \* **boundValues**
- [ItemQueryPostHooks](#) \* **hooks** = nullptr
- [SearchXmlCachingReader](#) & **reader**
- SearchXml::Relation **relation** = SearchXml::Equal
- QString & **sql**

## 6.626 Digikam::FileActionItemInfoList Class Reference

Inheritance diagram for Digikam::FileActionItemInfoList:



### Public Member Functions

- void **dbFinished** () const
- void **dbProcessed** (int numberOfInfos) const



- void **dbProcessedOne** () const  
*db worker progress info*
- **FileActionItemInfoList** (const [FileActionItemInfoList](#) &copy)
- void **finishedWriting** () const
- [FileActionProgressItemContainer](#) \* **progress** () const
- void **schedulingForDB** (const QString &action, [FileActionProgressItemCreator](#) \*const creator)
- void **schedulingForDB** (int numberOfInfos, const QString &action, [FileActionProgressItemCreator](#) \*const creator)  
*before sending to db worker*
- void **schedulingForWrite** (const QString &action, [FileActionProgressItemCreator](#) \*const creator) const
- void **schedulingForWrite** (int numberOfInfos, const QString &action, [FileActionProgressItemCreator](#) \*const creator) const  
*db worker calls this before sending to file worker*
- void **written** (int numberOfInfos) const
- void **writtenToOne** () const  
*file worker calls this when finished*

### Static Public Member Functions

- static [FileActionItemInfoList](#) **continueTask** (const QList< [ItemInfo](#) > &list, [FileActionProgressItemContainer](#) \*const container)
- static [FileActionItemInfoList](#) **create** (const QList< [ItemInfo](#) > &list)

### Public Attributes

- QExplicitlySharedDataPointer< [FileActionProgressItemContainer](#) > **container**

## 6.627 Digikam::FileActionMngr Class Reference

Inheritance diagram for Digikam::FileActionMngr:



### Classes

- class [Private](#)

### Public Types

- enum `GroupAction` { `AddToGroup` , `RemoveFromGroup` , `SplitGroup` }

### Public Slots

- void `addToGroup` (const [ItemInfo](#) &pick, const QList< [ItemInfo](#) > &infos)
- void `applyMetadata` (const QList< [ItemInfo](#) > &infos, const [DisjointMetadata](#) &hub)

- void **applyMetadata** (const QList< [ItemInfo](#) > &infos, [DisjointMetadata](#) \*hub)
- void **assignColorLabel** (const [ItemInfo](#) &info, int colorId)
- void **assignColorLabel** (const QList< [ItemInfo](#) > &infos, int colorId)
- void **assignPickLabel** (const [ItemInfo](#) &info, int pickId)
- void **assignPickLabel** (const QList< [ItemInfo](#) > &infos, int pickId)
- void **assignRating** (const [ItemInfo](#) &info, int rating)
- void **assignRating** (const QList< [ItemInfo](#) > &infos, int rating)
- void **assignTag** (const [ItemInfo](#) &info, int tagID)
- void **assignTag** (const QList< [ItemInfo](#) > &infos, int tagID)
- void **assignTags** (const [ItemInfo](#) &info, const QList< int > &tagIDs)
- void **assignTags** (const QList< [ItemInfo](#) > &infos, const QList< int > &tagIDs)
- void **assignTags** (const QList< qlonglong > &imageIds, const QList< int > &tagIDs)
- void **copyAttributes** (const [ItemInfo](#) &source, const QString &derivedPath)
- void **copyAttributes** (const [ItemInfo](#) &source, const QStringList &derivedPaths)
- void **removeFromGroup** (const [ItemInfo](#) &info)
- void **removeFromGroup** (const QList< [ItemInfo](#) > &infos)
- void **removeTag** (const [ItemInfo](#) &info, int tagID)
- void **removeTag** (const QList< [ItemInfo](#) > &infos, int tagID)
- void **removeTags** (const [ItemInfo](#) &info, const QList< int > &tagIDs)
- void **removeTags** (const QList< [ItemInfo](#) > &infos, const QList< int > &tagIDs)
- void **setExifOrientation** (const QList< [ItemInfo](#) > &infos, int orientation)
- void **transform** (const QList< [ItemInfo](#) > &infos, [MetaEngineRotation::TransformationAction](#) action)
- void **ungroup** (const [ItemInfo](#) &info)
- void **ungroup** (const QList< [ItemInfo](#) > &infos)

## Signals

- void **signalImageChangeFailed** (const QString &message, const QStringList &fileNames)

## Public Member Functions

- bool **isActive** ()
- bool **requestShutDown** ()
- void **shutDown** ()

## Static Public Member Functions

- static [FileActionMngr](#) \* **instance** ()

## Friends

- class [FileActionMngrCreator](#)

## 6.627.1 Member Function Documentation

### 6.627.1.1 transform

```
void Digikam::FileActionMngr::transform (
    const QList< ItemInfo > & infos,
    MetaEngineRotation::TransformationAction action ) [slot]
```

Flip or rotate. Note: The NoTransformation action is interpreted as Exif auto-rotate

## 6.628 Digikam::FileActionMngr::Private Class Reference

Inheritance diagram for Digikam::FileActionMngr::Private:



### Public Slots

- void **slotImageDataChanged** (const QString &path, bool removeThumbnails, bool notifyCache)
- void **slotLastProgressItemCompleted** ()
- void **slotSleepTimer** ()

## Signals

- void **signalAddTags** (const [FileActionItemInfoList](#) &infos, const QList< int > &tagIDs)
- void **signalApplyMetadata** (const [FileActionItemInfoList](#) &infos, [DisjointMetadata](#) \*hub)
- void **signalAssignColorLabel** (const [FileActionItemInfoList](#) &infos, int colorId)
- void **signalAssignPickLabel** (const [FileActionItemInfoList](#) &infos, int pickId)
- void **signalAssignRating** (const [FileActionItemInfoList](#) &infos, int rating)
- void **signalCopyAttributes** (const [FileActionItemInfoList](#) &infos, const QStringList &derivedPaths)
- void **signalEditGroup** (int groupAction, const [ItemInfo](#) &pick, const [FileActionItemInfoList](#) &infos)
- void **signalRemoveTags** (const [FileActionItemInfoList](#) &infos, const QList< int > &tagIDs)
- void **signalSetExifOrientation** (const [FileActionItemInfoList](#) &infos, int orientation)
- void **signalTasksFinished** ()
- void **signalTransform** (const [FileActionItemInfoList](#) &infos, int orientation)

## Public Member Functions

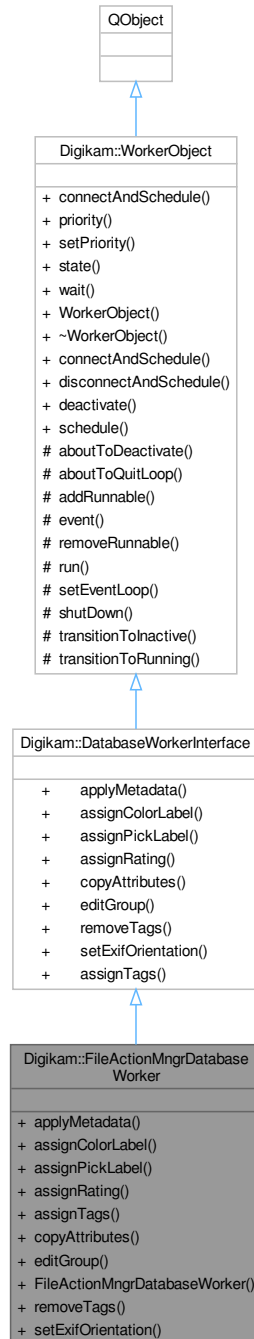
- void **applyMetadata** (const [FileActionItemInfoList](#) &infos, [DisjointMetadata](#) \*hub)
- void **assignColorLabel** (const [FileActionItemInfoList](#) &infos, int colorId)
- void **assignPickLabel** (const [FileActionItemInfoList](#) &infos, int pickId)
- void **assignRating** (const [FileActionItemInfoList](#) &infos, int rating)
- void **assignTags** (const [FileActionItemInfoList](#) &infos, const QList< int > &tagIDs)
- void **connectDatabaseToFileWorker** ()
- void **connectToDatabaseWorker** ()
- void **copyAttributes** (const [FileActionItemInfoList](#) &infos, const QStringList &derivedPaths)
- [PrivateProgressItemCreator](#) \* **dbProgressCreator** ()
- void **editGroup** (int groupAction, const [ItemInfo](#) &pick, const [FileActionItemInfoList](#) &infos)
- [PrivateProgressItemCreator](#) \* **fileProgressCreator** ()
- bool **isActive** () const
- **Private** ([FileActionMngr](#) \*const qq)
- void **removeTags** (const [FileActionItemInfoList](#) &infos, const QList< int > &tagIDs)
- void **setExifOrientation** (const [FileActionItemInfoList](#) &infos, int orientation)
- bool **shallSendForWriting** (qulonglong id)
  - *db worker will send info to file worker if returns true*
- void **startingToWrite** (const QList< [ItemInfo](#) > &infos)
  - *file worker calls this when receiving a task*
- void **transform** (const [FileActionItemInfoList](#) &infos, int orientation)

## Public Attributes

- QString **dbMessage**
- [PrivateProgressItemCreator](#) **dbProgress**
- [DatabaseWorkerInterface](#) \* **dbWorker** = nullptr
- [PrivateProgressItemCreator](#) **fileProgress**
- [ParallelAdapter](#)< [FileWorkerInterface](#) > \* **fileWorker**
- QMutex **mutex**
- [FileActionMngr](#) \* **q** = nullptr
- QSet< qulonglong > **scheduledToWrite**
- QTimer \* **sleepTimer** = nullptr
- QString **writerMessage**

## 6.629 Digikam::FileActionMngrDatabaseWorker Class Reference

Inheritance diagram for Digikam::FileActionMngrDatabaseWorker:



### Public Member Functions

- void `applyMetadata` (const `FileActionItemInfoList` &infos, `DisjointMetadata` \*hub) override
- void `assignColorLabel` (const `FileActionItemInfoList` &infos, int colorId) override

- void [assignPickLabel](#) (const [FileActionItemInfoList](#) &infos, int pickId) override
- void [assignRating](#) (const [FileActionItemInfoList](#) &infos, int rating) override
- void [assignTags](#) (const [FileActionItemInfoList](#) &infos, const QList< int > &tagIDs) override
- void [copyAttributes](#) (const [FileActionItemInfoList](#) &infos, const QStringList &derivedPaths) override
- void [editGroup](#) (int groupAction, const [ItemInfo](#) &pick, const [FileActionItemInfoList](#) &infos) override
- **FileActionMngrDatabaseWorker** ([FileActionMngr::Private](#) \*const dd)
- void [removeTags](#) (const [FileActionItemInfoList](#) &infos, const QList< int > &tagIDs) override
- void [setExifOrientation](#) (const [FileActionItemInfoList](#) &infos, int orientation) override

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const QObject \*sender, const char \*signal, const char \*method, Qt::↔ ConnectionType type=Qt::AutoConnection) const
- QThread::Priority **priority** () const
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DatabaseWorkerInterface](#)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=FlushSignals)
- void [schedule](#) ()

## Signals inherited from [Digikam::DatabaseWorkerInterface](#)

- void **writeMetadata** ([FileActionItemInfoList](#) infos, int flag)
- void **writeMetadataToFiles** ([FileActionItemInfoList](#) infos)
- void **writeOrientationToFiles** ([FileActionItemInfoList](#) infos, int orientation)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

### 6.629.1 Member Function Documentation

#### 6.629.1.1 [applyMetadata\(\)](#)

```
void Digikam::FileActionMngrDatabaseWorker::applyMetadata (
    const FileActionItemInfoList & infos,
    DisjointMetadata * hub ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

#### 6.629.1.2 [assignColorLabel\(\)](#)

```
void Digikam::FileActionMngrDatabaseWorker::assignColorLabel (
    const FileActionItemInfoList & infos,
    int colorId ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

#### 6.629.1.3 [assignPickLabel\(\)](#)

```
void Digikam::FileActionMngrDatabaseWorker::assignPickLabel (
    const FileActionItemInfoList & infos,
    int pickId ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

#### 6.629.1.4 [assignRating\(\)](#)

```
void Digikam::FileActionMngrDatabaseWorker::assignRating (
    const FileActionItemInfoList & infos,
    int rating ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).



### 6.629.1.5 assignTags()

```
void Digikam::FileActionMngrDatabaseWorker::assignTags (
    const FileActionItemInfoList & infos,
    const QList< int > & tagIDs ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

### 6.629.1.6 copyAttributes()

```
void Digikam::FileActionMngrDatabaseWorker::copyAttributes (
    const FileActionItemInfoList & infos,
    const QStringList & derivedPaths ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

### 6.629.1.7 editGroup()

```
void Digikam::FileActionMngrDatabaseWorker::editGroup (
    int groupAction,
    const ItemInfo & pick,
    const FileActionItemInfoList & infos ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

### 6.629.1.8 removeTags()

```
void Digikam::FileActionMngrDatabaseWorker::removeTags (
    const FileActionItemInfoList & infos,
    const QList< int > & tagIDs ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

### 6.629.1.9 setExifOrientation()

```
void Digikam::FileActionMngrDatabaseWorker::setExifOrientation (
    const FileActionItemInfoList & infos,
    int orientation ) [override], [virtual]
```

Reimplemented from [Digikam::DatabaseWorkerInterface](#).

## 6.630 Digikam::FileActionMngrFileWorker Class Reference

Inheritance diagram for Digikam::FileActionMngrFileWorker:



### Public Member Functions

- **FileActionMngrFileWorker** ([FileActionMngr::Private](#) \*const dd)
- void [transform](#) (const [FileActionItemInfoList](#) &infos, int orientation) override

- void [writeMetadata](#) (const [FileActionItemInfoList](#) &infos, int flags) override
- void [writeMetadataToFiles](#) (const [FileActionItemInfoList](#) &infos) override
- void [writeOrientationToFiles](#) (const [FileActionItemInfoList](#) &infos, int orientation) override

### Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const char \*method, [Qt::](#)←  
ConnectionType type=[Qt::AutoConnection](#)) const
- [QThread::Priority](#) **priority** () const
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

### Public Slots inherited from [Digikam::FileWorkerInterface](#)

### Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

### Signals inherited from [Digikam::FileWorkerInterface](#)

- void **imageChangeFailed** (const [QString](#) &message, const [QStringList](#) &fileNames)
- void **imageDataChanged** (const [QString](#) &path, bool removeThumbnails, bool notifyCache)

### Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

### Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#) \*receiver,  
const char \*method, [Qt::](#)ConnectionType type=[Qt::AutoConnection](#))
- static bool **disconnectAndSchedule** (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#)  
\*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

### 6.630.1 Member Function Documentation

#### 6.630.1.1 transform()

```
void Digikam::FileActionMngrFileWorker::transform (
    const FileActionItemInfoList & infos,
    int orientation ) [override], [virtual]
```

Reimplemented from [Digikam::FileWorkerInterface](#).

#### 6.630.1.2 writeMetadata()

```
void Digikam::FileActionMngrFileWorker::writeMetadata (
    const FileActionItemInfoList & infos,
    int flags ) [override], [virtual]
```

Reimplemented from [Digikam::FileWorkerInterface](#).

#### 6.630.1.3 writeMetadataToFiles()

```
void Digikam::FileActionMngrFileWorker::writeMetadataToFiles (
    const FileActionItemInfoList & infos ) [override], [virtual]
```

Reimplemented from [Digikam::FileWorkerInterface](#).

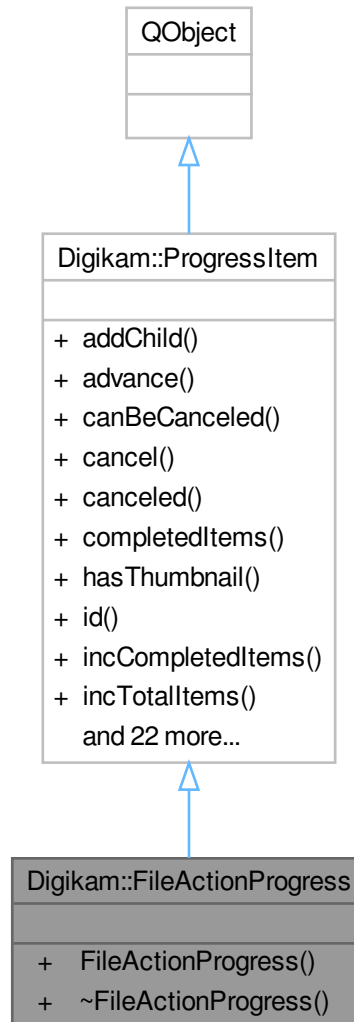
#### 6.630.1.4 writeOrientationToFiles()

```
void Digikam::FileActionMngrFileWorker::writeOrientationToFiles (
    const FileActionItemInfoList & infos,
    int orientation ) [override], [virtual]
```

Reimplemented from [Digikam::FileWorkerInterface](#).

## 6.631 Digikam::FileActionProgress Class Reference

Inheritance diagram for Digikam::FileActionProgress:



### Signals

- void `signalComplete ()`

### Signals inherited from [Digikam::ProgressItem](#)

- void `progressItemAdded (ProgressItem *item)`  
*Emitted when a new ProgressItem is added.*
- void `progressItemCanceled (ProgressItem *item)`

Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.

- void **progressItemCanceledById** (const QString &id)
- void **progressItemCompleted** (ProgressItem \*item)
 

Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.
- void **progressItemLabel** (ProgressItem \*item, const QString &label)
 

Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.
- void **progressItemProgress** (ProgressItem \*item, unsigned int v)
 

Emitted when the progress value of an item changes.
- void **progressItemStatus** (ProgressItem \*item, const QString &mess)
 

Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.
- void **progressItemThumbnail** (ProgressItem \*item, const QPixmap &thumb)
 

Emitted when the thumbnail data must be set in item.
- void **progressItemUsesBusyIndicator** (ProgressItem \*item, bool value)
 

Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.

## Public Member Functions

- **FileActionProgress** (const QString &name)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void **addChild** (ProgressItem \*const kiddo)
- bool **advance** (unsigned int v)
 

Advance total items processed by n values and update percentage in progressbar.
- bool **canBeCanceled** () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool **hasThumbnail** () const
- const QString & **id** () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & **label** () const
- ProgressItem \* **parent** () const
- unsigned int **progress** () const
- **ProgressItem** (ProgressItem \*const parent, const QString &id, const QString &label, const QString &status, bool canBeCanceled, bool hasThumb)
- void **removeChild** (ProgressItem \*const kiddo)
- void **reset** ()
 

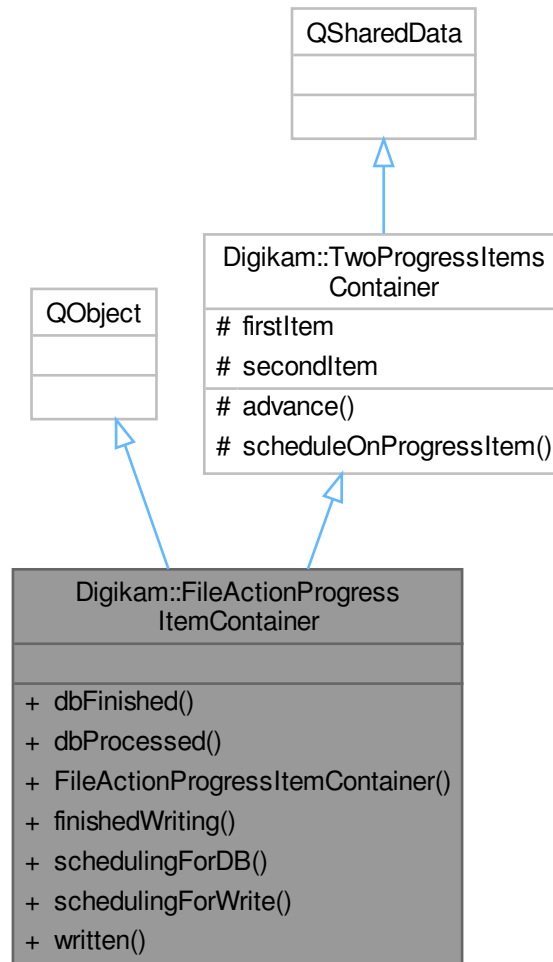
Reset the progress value of this item to 0 and the status string to the empty string.
- void **setComplete** ()
 

Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.
- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)

- void [setProgress](#) (unsigned int v)  
*Set the progress (percentage of completion) value of this item.*
- void [setShowAtStart](#) (bool [showAtStart](#))  
*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void [setStatus](#) (const QString &v)  
*Set the string to be used for showing this item's current status.*
- void [setThumbnail](#) (const QIcon &icon)  
*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void [setUsesBusyIndicator](#) (bool useBusyIndicator)  
*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool [showAtStart](#) () const
- const QString & [status](#) () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()  
*Recalculate progress according to total/completed items and update.*
- bool [usesBusyIndicator](#) () const

## 6.632 Digikam::FileActionProgressItemContainer Class Reference

Inheritance diagram for Digikam::FileActionProgressItemContainer:



### Signals

- void **signalWritingDone** ()

### Public Member Functions

- void **dbFinished** ()
- void **dbProcessed** (int numberOfInfos)
- void **finishedWriting** ()
- void **schedulingForDB** (int numberOfInfos, const QString &action, [FileActionProgressItemCreator](#) \*const creator)
- void **schedulingForWrite** (int numberOfInfos, const QString &action, [FileActionProgressItemCreator](#) \*const creator)
- void **written** (int numberOfInfos)



**Additional Inherited Members****Protected Member Functions inherited from [Digikam::TwoProgressItemsContainer](#)**

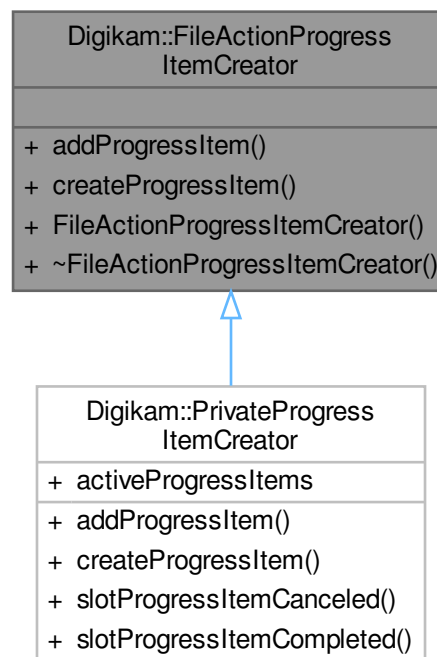
- void **advance** (QAtomicPointer< [ProgressItem](#) > &ptr, int n)
- void **scheduleOnProgressItem** (QAtomicPointer< [ProgressItem](#) > &ptr, int total, const QString &action, [FileActionProgressItemCreator](#) \*const creator)

**Protected Attributes inherited from [Digikam::TwoProgressItemsContainer](#)**

- QAtomicPointer< [ProgressItem](#) > **firstItem**
- QAtomicPointer< [ProgressItem](#) > **secondItem**

**6.633 Digikam::FileActionProgressItemCreator Class Reference**

Inheritance diagram for Digikam::FileActionProgressItemCreator:

**Public Member Functions**

- virtual void **addProgressItem** ([ProgressItem](#) \*const item)=0
- virtual [ProgressItem](#) \* **createProgressItem** (const QString &action) const =0

## 6.634 Digikam::FilePropertiesOption Class Reference

Inheritance diagram for Digikam::FilePropertiesOption:



### Protected Member Functions

- QString `parseOperation` (`ParseSettings` &settings, const `QRegularExpressionMatch` &match) override

## Protected Member Functions inherited from Digikam::Rule

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from Digikam::Rule

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from Digikam::Rule

- void **signalTokenTriggered** (const QString &)

## Public Member Functions inherited from Digikam::Option

- **Option** (const QString &name, const QString &description)
- **Option** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from Digikam::Rule

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- **ParseResults** **parse** (**ParseSettings** &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Static Public Member Functions inherited from Digikam::Rule

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from Digikam::Rule

- virtual void **slotTokenTriggered** (const QString &)

## 6.634.1 Member Function Documentation

### 6.634.1.1 parseOperation()

```
QString Digikam::FilePropertiesOption::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [virtual]
```

TODO: describe me

## Parameters

<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Option](#).

## 6.635 Digikam::FileReadLocker Class Reference

### Public Member Functions

- **FileReadLocker** (const QString &filePath)

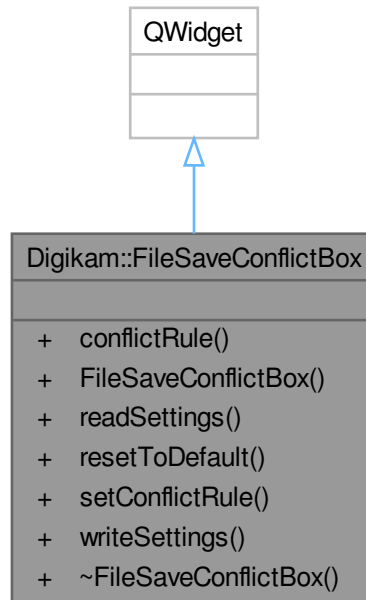
## 6.636 Digikam::FileReadWriteLockKey Class Reference

### Public Member Functions

- **FileReadWriteLockKey** (const QString &filePath)
- void **lockForRead** ()
- void **lockForWrite** ()
- bool **tryLockForRead** ()
- bool **tryLockForRead** (int timeout)
- bool **tryLockForWrite** ()
- bool **tryLockForWrite** (int timeout)
- void **unlock** ()

## 6.637 Digikam::FileSaveConflictBox Class Reference

Inheritance diagram for Digikam::FileSaveConflictBox:



### Public Types

- enum `ConflictRule` { `OVERWRITE = 0` , `DIFFNAME` , `SKIPFILE` }

### Signals

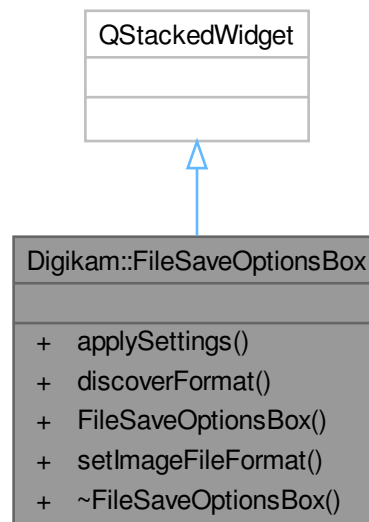
- void `signalConflictButtonChanged` (int)

### Public Member Functions

- ConflictRule `conflictRule` () const
- `FileSaveConflictBox` (QWidget \*const parent, bool addSkip=false)
- void `readSettings` (const KConfigGroup &group)
- void `resetToDefault` ()
- void `setConflictRule` (ConflictRule r)
- void `writeSettings` (KConfigGroup &group)

## 6.638 Digikam::FileSaveOptionsBox Class Reference

Inheritance diagram for Digikam::FileSaveOptionsBox:



### Public Types

- enum `FORMAT` {  
`NONE = 0`, `JPEG`, `PNG`, `TIFF`,  
`PGF`, `JXL`, `WEBP`, `AVIF` }

### Public Member Functions

- void `applySettings` ()
- `FORMAT` `discoverFormat` (const QString &filename, `FORMAT` fallback=`NONE`)
- `FileSaveOptionsBox` (QWidget \*const parent=nullptr)  
*Constructor. Don't forget to call `setDialog` after creation of the dialog.*
- void `setImageFileFormat` (const QString &)
- `~FileSaveOptionsBox` () override

### 6.638.1 Member Enumeration Documentation

#### 6.638.1.1 FORMAT

enum `Digikam::FileSaveOptionsBox::FORMAT`

## Enumerator

NONE	<p>Warning</p> <p>Order is important here. See filesaveoptionbox.cpp which use these values to fill a stack of widgets.</p>
------	---

## 6.638.2 Constructor & Destructor Documentation

### 6.638.2.1 FileSaveOptionsBox()

```
Digikam::FileSaveOptionsBox::FileSaveOptionsBox (
    QWidget *const parent = nullptr ) [explicit]
```

## Parameters

<i>parent</i>	the parent for Qt's parent child mechanism
---------------	--

### 6.638.2.2 ~FileSaveOptionsBox()

```
Digikam::FileSaveOptionsBox::~FileSaveOptionsBox ( ) [override]
```

Destructor.

## 6.638.3 Member Function Documentation

### 6.638.3.1 discoverFormat()

```
FileSaveOptionsBox::FORMAT Digikam::FileSaveOptionsBox::discoverFormat (
    const QString & filename,
    FileSaveOptionsBox::FORMAT fallback = NONE )
```

Tries to discover a file format that has options to change based on a filename.

## Parameters

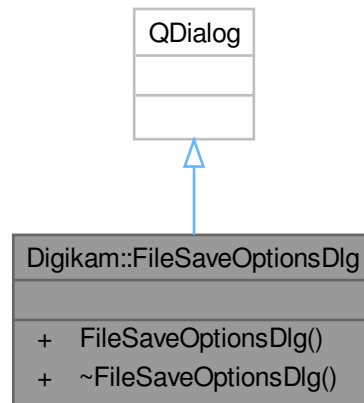
<i>filename</i>	file name to discover the desired format from
<i>fallback</i>	the fallback format to return if no format could be discovered based on the filename

## Returns

file format guessed from the file name or the given fallback format if no format could be guessed based on the file name

## 6.639 Digikam::FileSaveOptionsDlg Class Reference

Inheritance diagram for Digikam::FileSaveOptionsDlg:

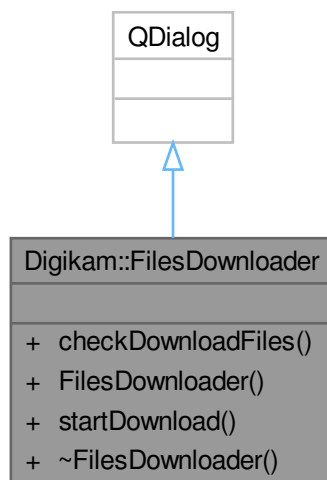


### Public Member Functions

- `FileSaveOptionsDlg` (`QWidget *const parent`, `FileSaveOptionsBox *const options`)

## 6.640 Digikam::FilesDownloader Class Reference

Inheritance diagram for Digikam::FilesDownloader:



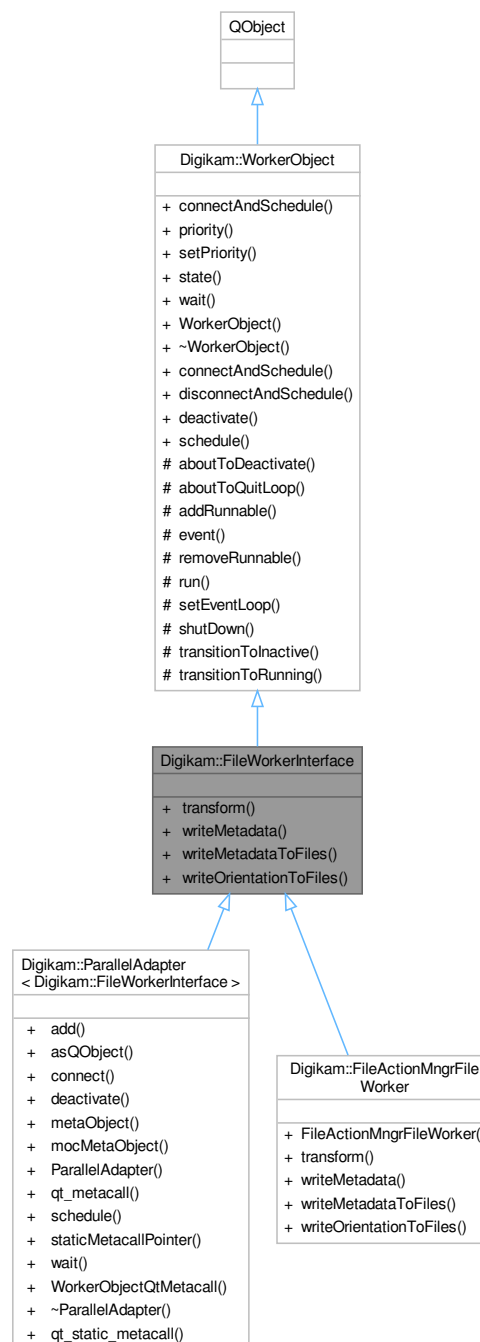


## Public Member Functions

- bool **checkDownloadFiles** () const
- **FilesDownloader** (QWidget \*const parent=nullptr)
- void **startDownload** ()

## 6.641 Digikam::FileWorkerInterface Class Reference

Inheritance diagram for Digikam::FileWorkerInterface:



### Public Slots

- virtual void **writeOrientationToFiles** (const [FileActionItemInfoList](#) &, int)

### Public Slots inherited from [Digikam::WorkerObject](#)

- void **deactivate** ([DeactivatingMode](#) mode=[FlushSignals](#))
- void **schedule** ()

### Signals

- void **imageChangeFailed** (const QString &message, const QStringList &fileNames)
- void **imageDataChanged** (const QString &path, bool removeThumbnails, bool notifyCache)

### Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

### Public Member Functions

- virtual void **transform** (const [FileActionItemInfoList](#) &, int)
- virtual void **writeMetadata** (const [FileActionItemInfoList](#) &, int)
- virtual void **writeMetadataToFiles** (const [FileActionItemInfoList](#) &)

### Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const char \*method, Qt::↔ ConnectionType type=Qt::AutoConnection) const
- QThread::Priority **priority** () const
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

### Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## 6.642 Digikam::FileWriteLocker Class Reference

### Public Member Functions

- **FileWriteLocker** (const QString &filePath)

## 6.643 Digikam::FilmContainer Class Reference

### Classes

- class [ListItem](#)
- class [Private](#)

### Public Types

- enum **CNFilmProfile** {  
**CNNeutral** = 0 , **CNKodakGold100** , **CNKodakGold200** , **CNKodakEktar100** ,  
**CNKodakProfessionalPortra160NC** , **CNKodakProfessionalPortra160VC** , **CNKodakProfessional**↔  
**Portra400NC** , **CNKodakProfessionalPortra400VC** ,  
**CNKodakProfessionalPortra800Box** , **CNKodakProfessionalPortra800P1** , **CNKodakProfessional**↔  
**Portra800P2** , **CNKodakProfessionalNewPortra160** ,  
**CNKodakProfessionalNewPortra400** , **CNKodakFarbwelt100** , **CNKodakFarbwelt200** , **CNKodak**↔  
**Farbwelt400** ,  
**CNKodakRoyalGold400** , **CNAgfaphotoVistaPlus200** , **CNAgfaphotoVistaPlus400** , **CNFujicolor**↔  
**Pro160S** ,  
**CNFujicolorPro160C** , **CNFujicolorNPL160** , **CNFujicolorPro400H** , **CNFujicolorPro800Z** ,  
**CNFujicolorSuperiaReala** , **CNFujicolorSuperia100** , **CNFujicolorSuperia200** , **CNFujicolorSuperia**↔  
**Xtra400** ,  
**CNFujicolorSuperiaXtra800** , **CNFujicolorTrueDefinition400** , **CNFujicolorSuperia1600** }

### Public Member Functions

- bool **applyBalance** () const
- CNFilmProfile **cnType** () const
- double **exposure** () const
- **FilmContainer** (CNFilmProfile profile, double gamma, bool sixteenBit)
- double **gamma** () const
- void **setApplyBalance** (bool val)
- void **setCNType** (CNFilmProfile profile)
- void **setExposure** (double strength)
- void **setGamma** (double val)
- void **setSixteenBit** (bool val)
- void **setWhitePoint** (const [DColor](#) &wp)
- [CBContainer](#) **toCB** () const
- [LevelsContainer](#) **toLevels** () const
- [DColor](#) **whitePoint** () const

### Static Public Member Functions

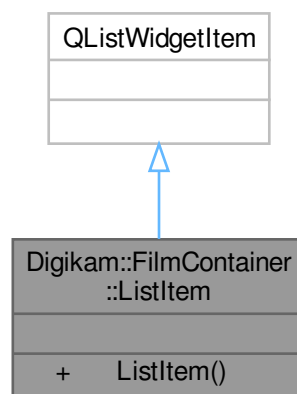
- static QList< [ListItem](#) \* > **profileItemList** (QListWidget \*const view)

### Static Public Attributes

- static const QMap< int, QString > **profileMap** = FilmContainer::profileMapInitializer()

## 6.644 Digikam::FilmContainer::ListItem Class Reference

Inheritance diagram for Digikam::FilmContainer::ListItem:



### Public Member Functions

- **ListItem** (const QString &text, QListWidget \*const parent, CNFilmProfile type)

## 6.645 Digikam::FilmContainer::Private Class Reference

### Public Attributes

- bool **applyBalance** = true
- CNFilmProfile **cnType** = CNNeutral
- double **exposure** = 1.0
- double **gamma** = 1.0
- [FilmProfile](#) **profile** = [FilmProfile](#)(1.0, 1.0, 1.0)
- bool **sixteenBit** = false
- [DColor](#) **whitePoint** = [DColor](#)([QColor](#)("white"), false)

## 6.646 Digikam::FilmFilter Class Reference

Inheritance diagram for Digikam::FilmFilter:



### Classes

- class [Private](#)

**Public Member Functions**

- **FilmFilter** (*DImg* \*const orgImage, *QObject* \*const parent=nullptr, const [FilmContainer](#) &settings=[FilmContainer](#)())
- **FilmFilter** (*QObject* \*const parent=nullptr)
- [FilterAction](#) **filterAction** () override
- *QString* **filterIdentifier** () const override
- void [readParameters](#) (const [FilterAction](#) &action) override

**Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)**

- virtual void **cancelFilter** ()
- [DImgThreadedFilter](#) (*DImg* \*const orgImage, *QObject* \*const parent, const *QString* &name=*QString*())
- [DImgThreadedFilter](#) (*QObject* \*const parent=nullptr, const *QString* &name=*QString*())
- const *QString* & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- *QList*< int > **multithreadedSteps** (int *stop*, int *start*=0) const
- virtual bool **parametersSuccessfullyRead** () const
- virtual *QString* **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const *QString* &name)
- void **setFilterVersion** (int *version*)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void **setupAndStartDirectly** (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int *progressBegin*=0, int *progressEnd*=100)
- void **setupFilter** (const [DImg](#) &orgImage)
- virtual void **startFilter** ()
- virtual void **startFilterDirectly** ()
- virtual *QList*< int > **supportedVersions** () const

**Public Member Functions inherited from [Digikam::DynamicThread](#)**

- [DynamicThread](#) (*QObject* \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- *QThread*::Priority **priority** () const
- void **setEmitSignals** (bool *emitThem*)
- void **setPriority** (*QThread*::Priority *priority*)
- State **state** () const
- [~DynamicThread](#) () override

**Static Public Member Functions**

- static int **CurrentVersion** ()
- static *QString* **DisplayableName** ()
- static *QString* **FilterIdentifier** ()
- static *QList*< int > **SupportedVersions** ()

**Additional Inherited Members****Public Types inherited from [Digikam::DynamicThread](#)**

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false



## 6.646.1 Member Function Documentation

### 6.646.1.1 filterAction()

`FilterAction Digikam::FilmFilter::filterAction ( ) [override], [virtual]`

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.646.1.2 filterIdentifier()

`QString Digikam::FilmFilter::filterIdentifier ( ) const [inline], [override], [virtual]`

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.646.1.3 readParameters()

```
void Digikam::FilmFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.647 Digikam::FilmFilter::Private Class Reference

### Public Attributes

- [FilmContainer](#) film

## 6.648 Digikam::FilmGrainContainer Class Reference

### Public Member Functions

- `bool isDirty ( ) const`

### Public Attributes

- `bool addChrominanceBlueNoise = false`
- `bool addChrominanceRedNoise = false`
- `bool addLuminanceNoise = true`
- `int chromaBlueHighlights = -100`
- `int chromaBlueIntensity = 25`
- `int chromaBlueMidtones = 0`
- `int chromaBlueShadows = -100`
- `int chromaRedHighlights = -100`
- `int chromaRedIntensity = 25`
- `int chromaRedMidtones = 0`
- `int chromaRedShadows = -100`
- `int grainSize = 1`
- `int lumaHighlights = -100`
- `int lumaIntensity = 25`
- `int lumaMidtones = 0`
- `int lumaShadows = -100`
- `bool photoDistribution = false`

## 6.649 Digikam::FilmGrainFilter Class Reference

Inheritance diagram for Digikam::FilmGrainFilter:



### Public Member Functions

- **FilmGrainFilter** (`Dimg *const orgImage`, `QObject *const parent=nullptr`, `const FilmGrainContainer &settings=FilmGrainContainer()`)

- [FilmGrainFilter](#) ([DImgThreadedFilter](#) \*const parentFilter, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const [FilmGrainContainer](#) &settings=[FilmGrainContainer](#)())
- **FilmGrainFilter** (QObject \*const parent=nullptr)
- [FilterAction](#) filterAction () override
- QString filterIdentifier () const override
- void readParameters (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.649.1 Constructor & Destructor Documentation

### 6.649.1.1 FilmGrainFilter()

```
Digikam::FilmGrainFilter::FilmGrainFilter (
    DImgThreadedFilter *const parentFilter,
    const DImg & orgImage,
    const DImg & destImage,
    int progressBegin = 0,
    int progressEnd = 100,
    const FilmGrainContainer & settings = FilmGrainContainer() ) [explicit]
```

Constructor for slave mode: execute immediately in current thread with specified master filter.

## 6.649.2 Member Function Documentation

### 6.649.2.1 filterAction()

```
FilterAction Digikam::FilmGrainFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.649.2.2 filterIdentifier()

```
QString Digikam::FilmGrainFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

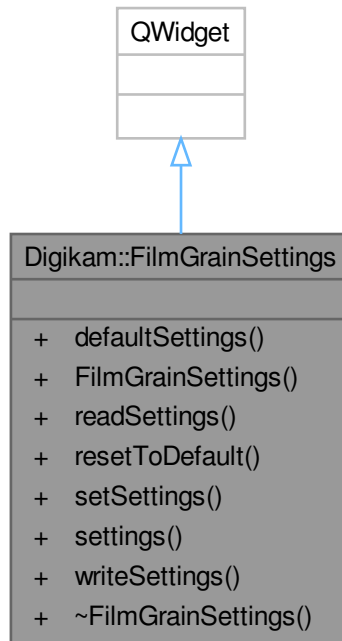
### 6.649.2.3 readParameters()

```
void Digikam::FilmGrainFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.650 Digikam::FilmGrainSettings Class Reference

Inheritance diagram for Digikam::FilmGrainSettings:



### Signals

- void **signalSettingsChanged** ()

### Public Member Functions

- [FilmGrainContainer](#) **defaultSettings** () const
- **FilmGrainSettings** (QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const [FilmGrainContainer](#) &settings)
- [FilmGrainContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.651 Digikam::FilmProfile Class Reference

### Public Member Functions

- double **balance** (int channel) const
- double **dmax** (int channel) const
- **FilmProfile** (double rdm=0.0, double gdm=0.0, double bdm=0.0)
- [FilmProfile](#) & **setBalance** (double rB, double gB, double bB)
- [FilmProfile](#) & **setWp** (double rWp, double gWp, double bWp)
- double **wp** (int channel) const

## 6.652 Digikam::Filter Class Reference

### Public Member Functions

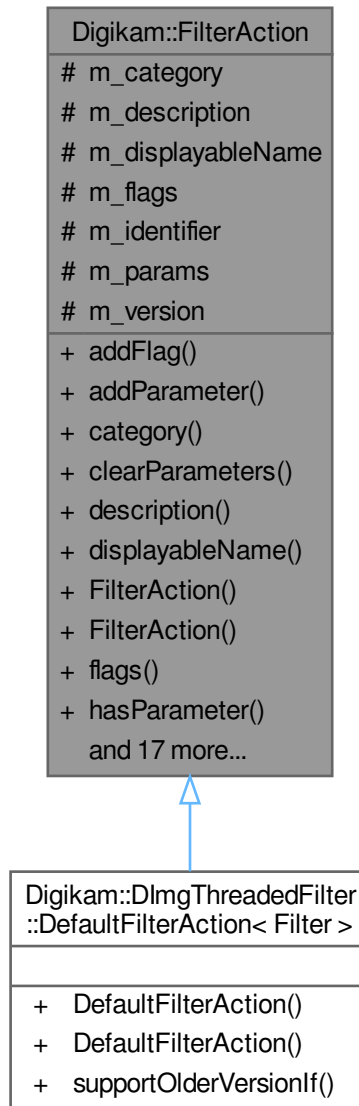
- void **fromString** (const QString &filter)
- bool **match** (const QStringList &wildcards, const QString &name)
- bool **matchesCurrentFilter** (const [CamItemInfo](#) &item)
- const QStringList & **mimeWildcards** (const QString &mime)
- const QRegularExpression & **regexp** (const QString &wildcard)
- QString **toString** ()

### Public Attributes

- QStringList **fileFilter**
- QHash< QString, QRegularExpression > **filterHash**
- QStringList **ignoreExtensions**
- QStringList **ignoreNames**
- QString **mimeFilter**
- QHash< QString, QStringList > **mimeHash**
- QString **name**
- bool **onlyNew** = false
- QStringList **pathFilter**

## 6.653 Digikam::FilterAction Class Reference

Inheritance diagram for Digikam::FilterAction:



### Public Types

- enum `Category` { `ReproducibleFilter` = 0 , `ComplexFilter` = 1 , `DocumentedHistory` = 2 , `CategoryFirst` = `ReproducibleFilter` , `CategoryLast` = `DocumentedHistory` }
- enum `Flag` { `ExplicitBranch` = 1 << 0 }



## Public Member Functions

- void **addFlag** (Flags flags)
- void **addParameter** (const QString &key, const QVariant &value)
  - Sets parameter, removing all other values for the same key.*
- [Category](#) **category** () const
- void **clearParameters** ()
  - Clear all parameters.*
- QString **description** () const
- QString **displayName** () const
- **FilterAction** (const QString &identifier, int version, [Category](#) category=[ReproducibleFilter](#))
- Flags **flags** () const
- bool **hasParameter** (const QString &key) const
- bool **hasParameters** () const
- QString **identifier** () const
- bool **isNull** () const
- bool **operator==** (const [FilterAction](#) &other) const
- QVariant & **parameter** (const QString &key)
- const QVariant **parameter** (const QString &key) const
- template<typename T >
  - T **parameter** (const QString &key) const
- template<typename T >
  - T **parameter** (const QString &key, const T &defaultValue) const
- QHash< QString, QVariant > & **parameters** ()
- const QHash< QString, QVariant > & **parameters** () const
- void **removeFlag** (Flags flags)
- void **removeParameters** (const QString &key)
  - Removes all parameters for key.*
- void **setDescription** (const QString &description)
- void **setDisplayName** (const QString &displayName)
- void **setFlags** (Flags flags)
- void **setParameters** (const QHash< QString, QVariant > &params)
  - Replaces parameters.*
- int **version** () const

## Protected Attributes

- [Category](#) **m\_category** = [ReproducibleFilter](#)
  - NOTE: Value class, do not create a d-pointer.*
- QString **m\_description**
- QString **m\_displayableName**
- Flags **m\_flags**
- QString **m\_identifier**
- QHash< QString, QVariant > **m\_params**
- int **m\_version** = 0

## 6.653.1 Member Enumeration Documentation

### 6.653.1.1 Category

```
enum Digikam::FilterAction::Category
```

## Enumerator

ReproducibleFilter	<p><b>Note</b></p> <p>Do not change existing values, they are written to XML in files! When given the set of stored parameters and the original data, an identical result will be produced.</p>
ComplexFilter	The operation is documented and a number of parameters may be known, but the identical result cannot be reproduced. It may be possible to produce a sufficiently similar result.
DocumentedHistory	The source images are known, a textual description may be added, but there is no way to automatically replay

**6.653.1.2 Flag**

```
enum Digikam::FilterAction::Flag
```

## Enumerator

ExplicitBranch	The editing step of this filter action explicitly branches from the parent. This is an optional hint that the result is meant as a new version.
----------------	---

**6.653.2 Member Function Documentation****6.653.2.1 description()**

```
QString Digikam::FilterAction::description ( ) const
```

Returns a description / comment for this action. In the case of DocumentedHistory, this may be the most useful value.

**6.653.2.2 hasParameters()**

```
bool Digikam::FilterAction::hasParameters ( ) const
```

Access parameters. A parameters is a key -> value pair. Keys need to be unique.

**6.653.2.3 identifier()**

```
QString Digikam::FilterAction::identifier ( ) const
```

Returns a technical identifier for the filter used to produce this action. Can include a namespace. Example↔  
: digikam:charcoal

**6.653.2.4 parameter()** [1/2]

```
template<typename T >
T Digikam::FilterAction::parameter (
    const QString & key ) const [inline]
```

Returns parameter converted from QVariant to given type

**6.653.2.5 parameter()** [2/2]

```
template<typename T >
T Digikam::FilterAction::parameter (
    const QString & key,
    const T & defaultValue ) const [inline]
```

Read parameter with a default value: If there is a parameter for the given key, return it converted from QVariant to the template type. If there is no parameter, return the given default value.

**6.653.2.6 version()**

```
int Digikam::FilterAction::version ( ) const
```

Returns the version ( $\geq 1$ ) of the filter used to produce this action. When a filter / tool is found by the identifier, it can decide by the version if it supports this action and which parameters it expects.

## 6.654 Digikam::FilterActionFilter Class Reference

Inheritance diagram for Digikam::FilterActionFilter:



### Public Member Functions

- void **addFilterAction** (const [FilterAction](#) &action)
- void **addFilterActions** (const QList< [FilterAction](#) > &actions)

- `QList< FilterAction > appliedFilterActions ()` const
- `bool completelyApplied ()` const
- `FilterAction failedAction ()` const
- `int failedActionIndex ()` const
- `QString failedActionMessage ()` const
- `FilterAction filterAction ()` override
- `FilterActionFilter (QObject *const parent=nullptr)`
- `QList< FilterAction > filterActions ()` const
- `QString filterIdentifier ()` const override
- `bool isComplexAction ()` const
- `bool isReproducible ()` const
- `bool isSupported ()` const
- `void readParameters (const FilterAction &) override`
- `void setContinueOnError (bool cont)`
- `void setFilterAction (const FilterAction &action)`
- `void setFilterActions (const QList< FilterAction > &actions)`

### Public Member Functions inherited from `Digikam::DImgThreadedFilter`

- `virtual void cancelFilter ()`
- `DImgThreadedFilter (DImg *const orgImage, QObject *const parent, const QString &name=QString())`
- `DImgThreadedFilter (QObject *const parent=nullptr, const QString &name=QString())`
- `const QString & filterName ()`
- `int filterVersion ()` const
- `DImg getTargetImage ()`
- `QList< int > multithreadedSteps (int stop, int start=0)` const
- `virtual bool parametersSuccessfullyRead ()` const
- `virtual QString readParametersError (const FilterAction &actionThatFailed)` const
- `void setFilterName (const QString &name)`
- `void setFilterVersion (int version)`
- `void setOriginalImage (const DImg &orgImage)`
- `void setupAndStartDirectly (const DImg &orgImage, DImgThreadedFilter *const master, int progressBegin=0, int progressEnd=100)`
- `void setupFilter (const DImg &orgImage)`
- `virtual void startFilter ()`
- `virtual void startFilterDirectly ()`
- `virtual QList< int > supportedVersions ()` const

### Public Member Functions inherited from `Digikam::DynamicThread`

- `DynamicThread (QObject *const parent=nullptr)`
- `bool isFinished ()` const
- `bool isRunning ()` const
- `QThread::Priority priority ()` const
- `void setEmitSignals (bool emitThem)`
- `void setPriority (QThread::Priority priority)`
- `State state ()` const
- `~DynamicThread ()` override

### Protected Member Functions

- `void filterImage ()` override

### Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cleanupFilter](#) ()
- [DImgThreadedFilter](#) ([DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void [initFilter](#) ()
- void [initMaster](#) ()
- void [initSlave](#) ([DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- virtual int [modulateProgress](#) (int progress)
- void [postProgress](#) (int progress)
- virtual void [prepareDestImage](#) ()
- void [run](#) () override
- void [setSlave](#) ([DImgThreadedFilter](#) \*const slave)

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

### Public Slots inherited from [Digikam::DynamicThread](#)

- void [start](#) ()
- void [stop](#) ()
- void [wait](#) ()

### Signals inherited from [Digikam::DImgThreadedFilter](#)

- void [finished](#) (bool success)
- void [progress](#) (int progress)
- void [started](#) ()

### Signals inherited from [Digikam::DynamicThread](#)

- void [finished](#) ()
- void [starting](#) ()

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- [DImg m\\_destImage](#)
- [DImgThreadedFilter \\* m\\_master](#) = nullptr
- [QString m\\_name](#)
- [DImg m\\_orgImage](#)
- [int m\\_progressBegin](#) = 0
- [int m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- [int m\\_progressSpan](#) = 0
- [DImgThreadedFilter \\* m\\_slave](#) = nullptr
- [int m\\_version](#) = 1
- [bool m\\_wasCancelled](#) = false

### 6.654.1 Constructor & Destructor Documentation

#### 6.654.1.1 FilterActionFilter()

```
Digikam::FilterActionFilter::FilterActionFilter (
    QObject *const parent = nullptr ) [explicit]
```

A meta-filter applying other filter according to a list of FilterActions

### 6.654.2 Member Function Documentation

#### 6.654.2.1 appliedFilterActions()

```
QList< FilterAction > Digikam::FilterActionFilter::appliedFilterActions ( ) const
```

Returns the list of applied filter actions. This is probably identical to filterActions, but it can differ in some situations:

- if [completelyApplied\(\)](#) is false, it will contain only the successful actions
- the list is regenerated by the filters. If filterActions contains actions with an older version, still supported by the filter, the filter will now possibly return the newer, current version

#### 6.654.2.2 completelyApplied()

```
bool Digikam::FilterActionFilter::completelyApplied ( ) const
```

After the thread was run, you can find out if application was successful. A precondition is that at least [isComplexAction\(\)](#) and [isSupported\(\)](#) returns true. If all filters applied cleanly, [completelyApplied\(\)](#) returns true. [appliedActions\(\)](#) returns all applied actions, if [completelyApplied\(\)](#), the same as [filterActions\(\)](#). If not completely Applied, [failedAction\(\)](#) returns the action that failed, [failedActionIndex](#) its index in [filterActions\(\)](#), and [failedActionMessage](#) an optional error message. Note that [finished\(true\)](#) does not mean that [completelyApplied\(\)](#) is also true.

### 6.654.2.3 filterAction()

```
FilterAction Digikam::FilterActionFilter::filterAction ( ) [inline], [override], [virtual]
```

These methods do not make sense here. Use filterActions.

Implements [Digikam::DImgThreadedFilter](#).

### 6.654.2.4 filterIdentifier()

```
QString Digikam::FilterActionFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.654.2.5 filterImage()

```
void Digikam::FilterActionFilter::filterImage ( ) [override], [protected], [virtual]
```

Main image filter method. Override in subclass.

Implements [Digikam::DImgThreadedFilter](#).

### 6.654.2.6 isComplexAction()

```
bool Digikam::FilterActionFilter::isComplexAction ( ) const
```

Returns true if all FilterActions are reproducible or are ComplexFilters. That means the identical result may not be reproducible, but a sufficiently similar result may be available and apply will probably complete.

### 6.654.2.7 isReproducible()

```
bool Digikam::FilterActionFilter::isReproducible ( ) const
```

Returns true if all FilterActions are reproducible

### 6.654.2.8 isSupported()

```
bool Digikam::FilterActionFilter::isSupported ( ) const
```

Returns true if all actions are supported.

### 6.654.2.9 readParameters()

```
void Digikam::FilterActionFilter::readParameters (
    const FilterAction & ) [inline], [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).



**6.654.2.10 setContinueOnError()**

```
void Digikam::FilterActionFilter::setContinueOnError (
    bool cont )
```

Per default, the filter will stop when it encounters an unsupported action. If you want it to continue, set this to true. Only the last occurred error will then be reported.

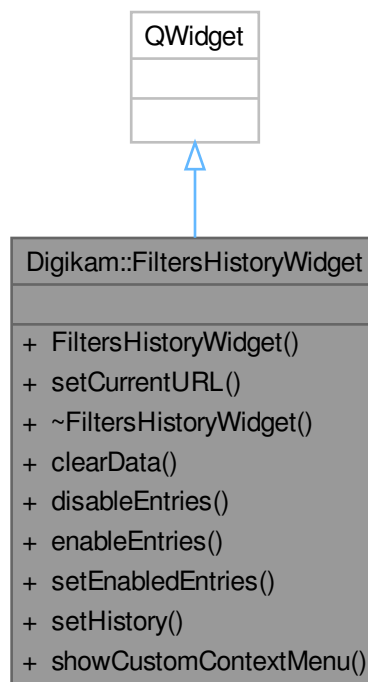
**6.654.2.11 setFilterActions()**

```
void Digikam::FilterActionFilter::setFilterActions (
    const QList< FilterAction > & actions )
```

Set - or add to existing list - the given filter actions

**6.655 Digikam::FiltersHistoryWidget Class Reference**

Inheritance diagram for Digikam::FiltersHistoryWidget:

**Public Slots**

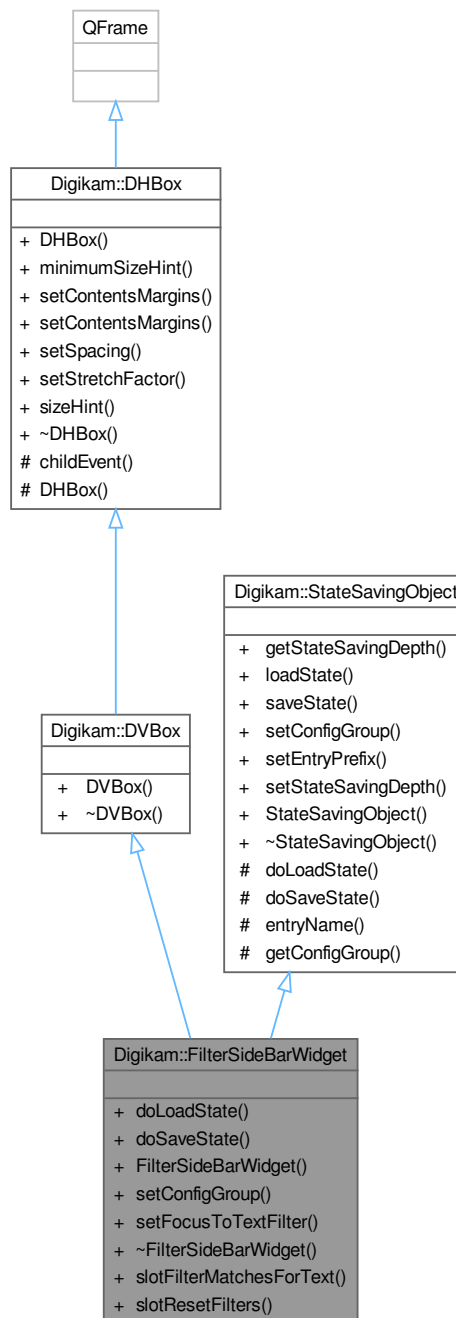
- void **clearData** ()
- void **disableEntries** (int count)
- void **enableEntries** (int count)
- void **setEnabledEntries** (int count)
- void **setHistory** (const [DImageHistory](#) &history)
- void **showCustomContextMenu** (const QPoint &position)

## Public Member Functions

- **FiltersHistoryWidget** (QWidget \*const parent)
- void **setCurrentURL** (const QUrl &url=QUrl())

## 6.656 Digikam::FilterSideBarWidget Class Reference

Inheritance diagram for Digikam::FilterSideBarWidget:



## Public Slots

- void **slotFilterMatchesForText** (bool)
- void **slotResetFilters** ()

## Signals

- void **signalGeolocationFilterChanged** ([ItemFilterSettings::GeolocationCondition](#))
- void **signalMimeTypeFilterChanged** (int)
- void **signalRatingFilterChanged** (int, [ItemFilterSettings::RatingCondition](#), bool)
- void **signalSearchTextFilterChanged** (const [SearchTextFilterSettings](#) &)
- void **signalTagFilterChanged** (const QList< int > &includedTags, const QList< int > &excludedTags, [ItemFilterSettings::MatchingCondition](#) matchingCond, bool showUnTagged, const QList< int > &clTagIds, const QList< int > &plTagIds)

## Public Member Functions

- void **doLoadState** () override
- void **doSaveState** () override
- [FilterSideBarWidget](#) (QWidget \*const parent, [TagModel](#) \*const tagFilterModel)
- void **setConfigGroup** (const KConfigGroup &group) override
- void **setFocusToTextFilter** ()
- [~FilterSideBarWidget](#) () override

## Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.656.1 Detailed Description

[Sidebar](#) widget containing the all filter widgets.

### Author

jwienke

## 6.656.2 Constructor & Destructor Documentation

### 6.656.2.1 [FilterSideBarWidget\(\)](#)

```
Digikam::FilterSideBarWidget::FilterSideBarWidget (
    QWidget *const parent,
    TagModel *const tagFilterModel ) [explicit]
```

Constructor.

#### Parameters

<i>parent</i>	the parent for qt parent child mechanism
<i>tagFilterModel</i>	tag model to work on

### 6.656.2.2 [~FilterSideBarWidget\(\)](#)

```
Digikam::FilterSideBarWidget::~~FilterSideBarWidget ( ) [override]
```

Destructor.

## 6.656.3 Member Function Documentation

### 6.656.3.1 doLoadState()

```
void Digikam::FilterSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.656.3.2 doSaveState()

```
void Digikam::FilterSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.656.3.3 setConfigGroup()

```
void Digikam::FilterSideBarWidget::setConfigGroup (
    const KConfigGroup & group ) [override], [virtual]
```

Sets a dedicated config group that will be used to store and reload the state from. If this method is not called, a group based on the object name is used.

You can re-implement this method to pass the group set here to child objects. Don't forget to call this method in your implementation.

#### Parameters

<i>group</i>	config group to use for state saving and restoring
--------------	--

Reimplemented from [Digikam::StateSavingObject](#).

### 6.656.3.4 signalTagFilterChanged

```
void Digikam::FilterSideBarWidget::signalTagFilterChanged (
    const QList< int > & includedTags,
    const QList< int > & excludedTags,
    ItemFilterSettings::MatchingCondition matchingCond,
    bool showUnTagged,
    const QList< int > & c1TagIds,
    const QList< int > & p1TagIds ) [signal]
```

Emitted if the selected filter has changed.

#### Parameters

<i>includedTags</i>	a list of included tag ids
---------------------	----------------------------

## Parameters

<i>excludedTags</i>	a list of excluded tag ids
<i>matchingCond</i>	condition to join the selected tags
<i>showUnTagged</i>	if this is true, only photos without a tag shall be shown
<i>clTagIds</i>	a list of color label tag ids
<i>plTagIds</i>	a list of pick label tag ids

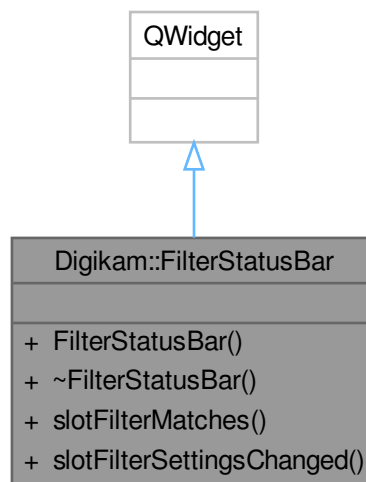
**6.656.3.5 slotResetFilters**

```
void Digikam::FilterSideBarWidget::slotResetFilters ( ) [slot]
```

Resets all selected filters.

**6.657 Digikam::FilterStatusBar Class Reference**

Inheritance diagram for Digikam::FilterStatusBar:

**Public Slots**

- void **slotFilterMatches** (bool)
- void **slotFilterSettingsChanged** (const [ItemFilterSettings](#) &settings)

**Signals**

- void **signalPopupFiltersView** ()
- void **signalResetFilters** ()

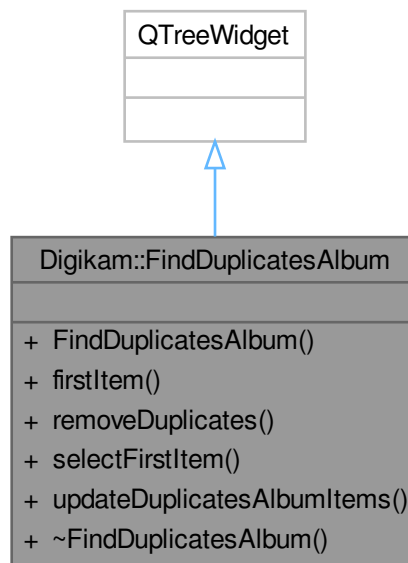
**Public Member Functions**

- **FilterStatusBar** (QWidget \*const parent)

**6.658 Digikam::FindDuplicatesAlbum Class Reference**

The [FindDuplicatesAlbum](#) class Widgets used to show all reference images.

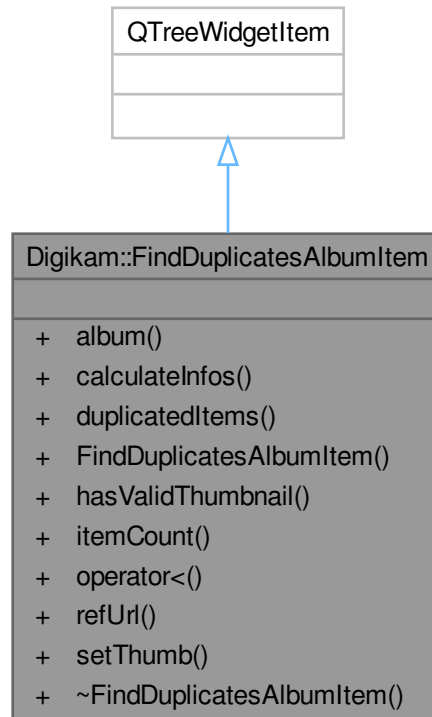
Inheritance diagram for Digikam::FindDuplicatesAlbum:

**Public Member Functions**

- **FindDuplicatesAlbum** (QWidget \*const parent=nullptr)
- QTreeWidgetItem \* **firstItem** ()
- void **removeDuplicates** ()
- void **selectFirstItem** ()
- void **updateDuplicatesAlbumItems** (const QList< [SAlbum](#) \* > &sAlbumsToRebuild, const QList< qlong-long > &deletedImages)

## 6.659 Digikam::FindDuplicatesAlbumItem Class Reference

Inheritance diagram for Digikam::FindDuplicatesAlbumItem:



### Public Types

- enum **Column** {  
**REFERENCE\_IMAGE** = 0 , **REFERENCE\_DATE** = 1 , **REFERENCE\_ALBUM** = 2 , **RESULT\_COUNT** = 3 ,  
**AVG\_SIMILARITY** = 4 }

### Public Member Functions

- [SAlbum](#) \* **album** () const
- void **calculateInfos** (const QList< qlonglong > &deletedImages=QList< qlonglong >())
- QList< [ItemInfo](#) > **duplicatedItems** ()
- **FindDuplicatesAlbumItem** (QTreeWidgetItem \*const parent, [SAlbum](#) \*const album)
- bool **isValidThumbnail** () const
- int **itemCount** () const
- bool **operator<** (const QTreeWidgetItem &other) const override
- QUrl **refUrl** () const
- void **setThumb** (const QPixmap &pix, bool hasThumb=true)



## 6.659.1 Member Function Documentation

### 6.659.1.1 calculateInfos()

```
void Digikam::FindDuplicatesAlbumItem::calculateInfos (
    const QList< qlonglong > & deletedImages = QList<qlonglong>() )
```

Calculates the duplicates count and average similarity.

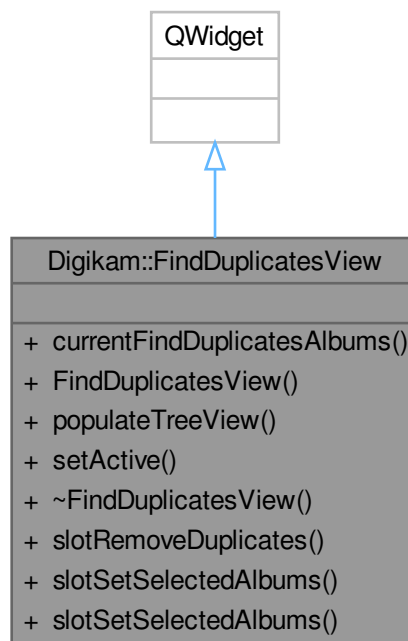
### 6.659.1.2 itemCount()

```
int Digikam::FindDuplicatesAlbumItem::itemCount ( ) const
```

Returns the item count.

## 6.660 Digikam::FindDuplicatesView Class Reference

Inheritance diagram for Digikam::FindDuplicatesView:



### Public Slots

- void **slotRemoveDuplicates** ()
- void **slotSetSelectedAlbums** (const QList< [PAlbum](#) \* > &albums)
- void **slotSetSelectedAlbums** (const QList< [TAlbum](#) \* > &albums)

## Signals

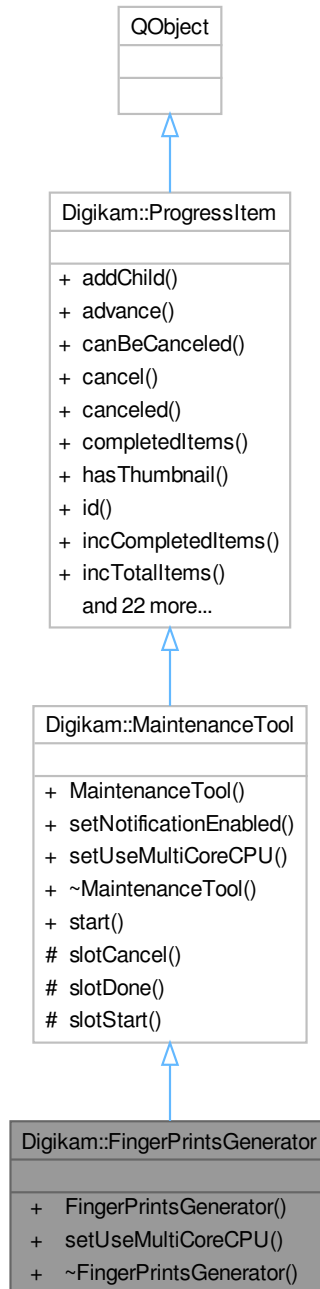
- void **signalScanNotification** (const QString &msg, int type)

## Public Member Functions

- QList< [SAlbum](#) \* > **currentFindDuplicatesAlbums** () const
- **FindDuplicatesView** (QWidget \*const parent=nullptr)
- void **populateTreeView** ()
- void **setActive** (bool val)

## 6.661 Digikam::FingerPrintsGenerator Class Reference

Inheritance diagram for Digikam::FingerPrintsGenerator:



### Signals

- void **signalScanNotification** (const QString &msg, int type)

## Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

## Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void [progressItemCanceledById](#) (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- [FingerPrintsGenerator](#) (const bool rebuildAll, const AlbumList &list=AlbumList(), [ProgressItem](#) \*const parent=nullptr)
- void [setUseMultiCoreCPU](#) (bool b) override

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- [MaintenanceTool](#) (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)

## Public Member Functions inherited from Digikam::ProgressItem

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool **advance** (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool **canBeCanceled** () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool **hasThumbnail** () const
- const QString & **id** () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & **label** () const
- [ProgressItem](#) \* **parent** () const
- unsigned int **progress** () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool canBeCanceled, bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)
- void **setProgress** (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void **setShowAtStart** (bool showAtStart)
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void **setStatus** (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void **setThumbnail** (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void **setUsesBusyIndicator** (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool **showAtStart** () const
- const QString & **status** () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool **usesBusyIndicator** () const

### Additional Inherited Members

## Public Slots inherited from Digikam::MaintenanceTool

- void **start** ()

## Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

### 6.661.1 Constructor & Destructor Documentation

#### 6.661.1.1 FingerPrintsGenerator()

```
Digikam::FingerPrintsGenerator::FingerPrintsGenerator (
    const bool rebuildAll,
    const AlbumList & list = AlbumList(),
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor using AlbumList as argument. If list is empty, whole Albums collection is processed.

### 6.661.2 Member Function Documentation

#### 6.661.2.1 setUseMultiCoreCPU()

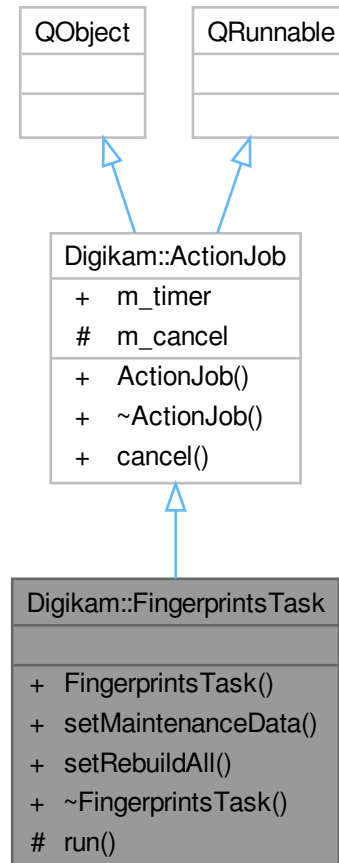
```
void Digikam::FingerPrintsGenerator::setUseMultiCoreCPU (
    bool ) [override], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented from [Digikam::MaintenanceTool](#).

## 6.662 Digikam::FingerprintsTask Class Reference

Inheritance diagram for Digikam::FingerprintsTask:



### Signals

- void **signalFinished** (const [ItemInfo](#) &, const QImage &)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Member Functions

- void **setMaintenanceData** ([MaintenanceData](#) \*const data=nullptr)
- void **setRebuildAll** (bool b)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Protected Member Functions

- void [run](#) () override

### Additional Inherited Members

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Public Attributes inherited from [Digikam::ActionJob](#)

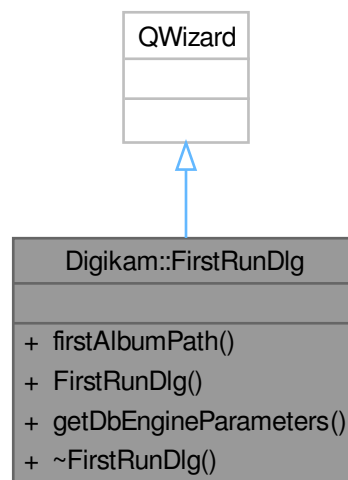
- QElapsedTimer [m\\_timer](#)

### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.663 Digikam::FirstRunDlg Class Reference

Inheritance diagram for Digikam::FirstRunDlg:





**Public Member Functions**

- QString **firstAlbumPath** () const
- **FirstRunDlg** (QWidget \*const parent=nullptr)
- [DbEngineParameters](#) **getDbEngineParameters** () const

**6.664 Digikam::FocusPoint Class Reference****Public Types**

- enum [TypePoint](#) { [Inactive](#) = 0 , [InFocus](#) = 1 , [Selected](#) = 2 , [SelectedInFocus](#) = 3 }

**Public Member Functions**

- **FocusPoint** (const [FocusPoint](#) &other)
- **FocusPoint** (const QRectF &rectF)
- **FocusPoint** (float x\_position, float y\_position, float width, float height)
- [FocusPoint](#) (float x\_position, float y\_position, float width, float height, [TypePoint](#) type)
- QPointF **getCenterPosition** () const
- QRectF **getRect** () const
- QRect **getRectBySize** (const QSize &size) const
- QSizeF **getSize** () const
- [TypePoint](#) **getType** () const
- QString **getTypeDescription** () const
- [FocusPoint](#) & **operator=** (const [FocusPoint](#) &other)
- void **setCenterPosition** (float x\_position, float y\_position)
- void **setRect** (const QRectF &rectF)
- void **setSize** (float width, float height)
- void **setType** ([TypePoint](#) type)

**6.664.1 Member Enumeration Documentation****6.664.1.1 TypePoint**

```
enum Digikam::FocusPoint::TypePoint
```

**Enumerator**

Inactive	The AF-point is not active.
InFocus	The AF-point is in focus.
Selected	The AF-point is selected but not in focus.
SelectedInFocus	The AF-point is selected and in focus.

## 6.664.2 Constructor & Destructor Documentation

### 6.664.2.1 FocusPoint()

```
Digikam::FocusPoint::FocusPoint (
    float x_position,
    float y_position,
    float width,
    float height,
    TypePoint type )
```

Focus point container constructors. Position and size are in float and a relative to the original image size. Typically, the area is define as percents of values depending of image size used to extract information from metadata. Like this, focus area can be drawn easily over a resized version of image.

## 6.664.3 Member Function Documentation

### 6.664.3.1 getRectBySize()

```
QRect Digikam::FocusPoint::getRectBySize (
    const QSize & size ) const
```

Return the real aera properties in image coordinates depending of the size.

### 6.664.3.2 operator=()

```
FocusPoint & Digikam::FocusPoint::operator= (
    const FocusPoint & other )
```

Equivalent to the copy constructor

### 6.664.3.3 setCenterPosition()

```
void Digikam::FocusPoint::setCenterPosition (
    float x_position,
    float y_position )
```

Accessors to relative properties of focus point area.

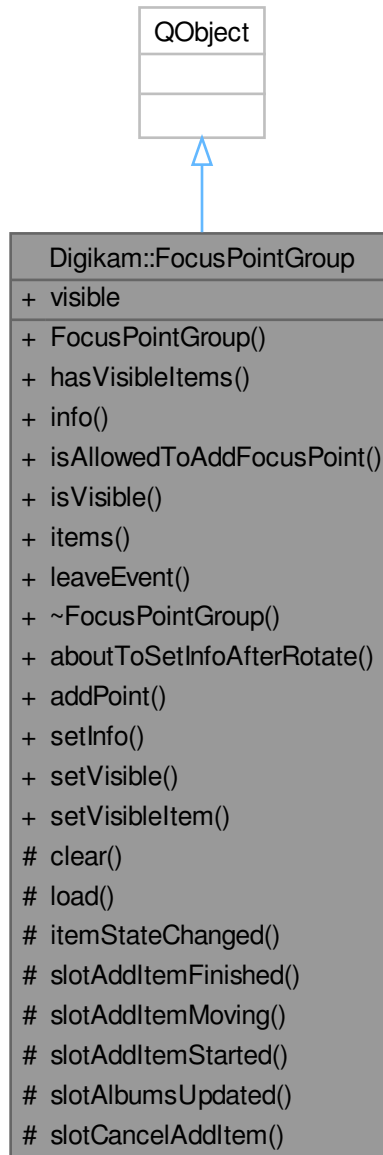
### 6.664.3.4 setType()

```
void Digikam::FocusPoint::setType (
    TypePoint type )
```

Focus point type properties accessor. See TypePoint enum definition for details.

## 6.665 Digikam::FocusPointGroup Class Reference

Inheritance diagram for Digikam::FocusPointGroup:



### Classes

- class [Private](#)

### Public Slots

- void **aboutToSetInfoAfterRotate** (const [ItemInfo](#) &info)

- void **addPoint** ()
- void **setInfo** (const [ItemInfo](#) &info)
- void **setVisible** (bool visible)
- void **setVisibleItem** ([RegionFrameItem](#) \*const item)

### Public Member Functions

- **FocusPointGroup** ([GraphicsDImgView](#) \*const view)
- bool **hasVisibleItems** () const
- [ItemInfo](#) **info** () const
- bool **isAllowedToAddFocusPoint** () const
- bool **isVisible** () const
- QList< [RegionFrameItem](#) \* > **items** () const
- void **leaveEvent** (QEvent \*)

### Protected Slots

- void **itemStateChanged** (int)
- void **slotAddItemFinished** (const QRectF &rect)
- void **slotAddItemMoving** (const QRectF &rect)
- void **slotAddItemStarted** (const QPointF &pos)
- void **slotAlbumsUpdated** (int type)
- void **slotCancelAddItem** ()

### Protected Member Functions

- void **clear** ()
- void **load** ()

### Properties

- bool **visible**

## 6.665.1 Member Function Documentation

### 6.665.1.1 setInfo

```
void Digikam::FocusPointGroup::setInfo (
    const ItemInfo & info ) [slot]
```

Sets the current [ItemInfo](#)

### 6.665.1.2 setVisible

```
void Digikam::FocusPointGroup::setVisible (
    bool visible ) [slot]
```

Shows or hides the frames

## 6.666 Digikam::FocusPointGroup::Private Class Reference

### Public Member Functions

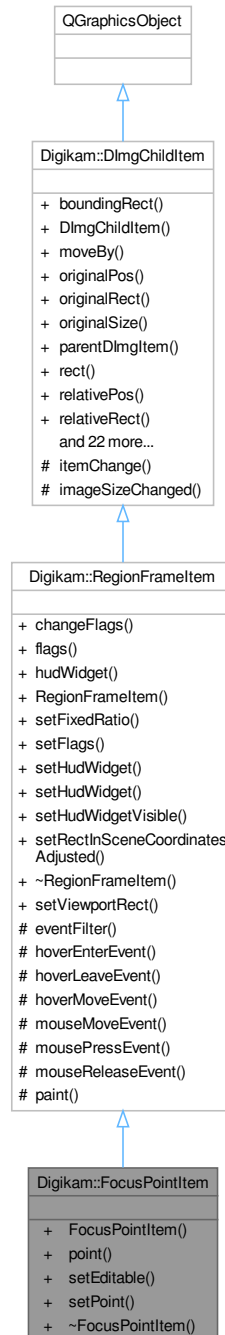
- [FocusPointItem](#) \* **addItem** (const [FocusPoint](#) &point)
- void **applyVisible** ()
- [FocusPointItem](#) \* **createItem** (const [FocusPoint](#) &point) const
- **Private** ([FocusPointGroup](#) \*const qq)

### Public Attributes

- bool **exifRotate** = true
- [ItemInfo](#) **info**
- [QList](#)< [FocusPointItem](#) \* > **items**
- [FocusPointItem](#) \* **manuallyAddedItem** = nullptr
- [ClickDragReleaseItem](#) \* **manuallyAddWrapItem** = nullptr
- [FocusPointGroup](#) \*const **q** = nullptr
- [FocusPointGroupState](#) **state** = NoPoints
- [GraphicsDImgView](#) \* **view** = nullptr
- [ItemVisibilityController](#) \* **visibilityController** = nullptr

## 6.667 Digikam::FocusPointItem Class Reference

Inheritance diagram for Digikam::FocusPointItem:



### Public Member Functions

- **FocusPointItem** (QGraphicsItem \*const parent)
- **FocusPoint point** () const
- void **setEditable** (bool allowEdit)
- void **setPoint** (const FocusPoint &point)

## Public Member Functions inherited from [Digikam::RegionFrameItem](#)

- void **changeFlags** (Flags flags, bool addOrRemove)
- Flags **flags** () const
- QGraphicsWidget \* **hudWidget** () const
- **RegionFrameItem** (QGraphicsItem \*const parent)
- void **setFixedRatio** (double ratio)
- void **setFlags** (Flags flags)
- void **setHudWidget** (QGraphicsWidget \*const hudWidget)
- void **setHudWidget** (QWidget \*const widget, Qt::WindowFlags wFlags=Qt::WindowFlags())
- void **setHudWidgetVisible** (bool visible)
- void **setRectInSceneCoordinatesAdjusted** (const QRectF &rect)

## Public Member Functions inherited from [Digikam::DImgChildItem](#)

- QRectF **boundingRect** () const override
- **DImgChildItem** (QGraphicsItem \*const parent=nullptr)
- void **moveBy** (qreal dx, qreal dy)
- QPoint **originalPos** () const
- QRect **originalRect** () const
- QSize **originalSize** () const
- [GraphicsDImgItem](#) \* **parentDImgItem** () const
- QRectF **rect** () const
- QPointF **relativePos** () const
- QRectF **relativeRect** () const
- QSizeF **relativeSize** () const
- void **setOriginalPos** (const QPointF &posInOriginal)
- void **setOriginalPos** (qreal x, qreal y)
- void **setOriginalRect** (const QRectF &rect)
- void **setOriginalRect** (qreal x, qreal y, qreal width, qreal height)
- void **setOriginalSize** (const QSizeF &sizeInOriginal)
- void **setOriginalSize** (qreal width, qreal height)
- void **setPos** (const QPointF &zoomedPos)
- void **setPos** (qreal x, qreal y)
- void **setRect** (const QRectF &rect)
- void **setRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRectInSceneCoordinates** (const QRectF &rect)
- void **setRelativePos** (const QPointF &relativePosition)
- void **setRelativePos** (qreal x, qreal y)
- void **setRelativeRect** (const QRectF &rect)
- void **setRelativeRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRelativeSize** (const QSizeF &relativeSize)
- void **setRelativeSize** (qreal width, qreal height)
- void **setSize** (const QSizeF &zoomedSize)
- void **setSize** (qreal width, qreal height)
- QSizeF **size** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::RegionFrameItem](#)

- enum **Flag** { **NoFlags** = 0 , **ShowResizeHandles** = 1 << 0 , **MoveByDrag** = 1 << 1 , **GeometryEditable** = ShowResizeHandles | MoveByDrag }

### Public Slots inherited from [Digikam::RegionFrameItem](#)

- void [setViewportRect](#) (const QRectF &rect)

### Signals inherited from [Digikam::RegionFrameItem](#)

- void [geometryEdited](#) ()

### Signals inherited from [Digikam::DImgChildItem](#)

- void [geometryChanged](#) ()
- void [geometryOnImageChanged](#) ()
- void [positionChanged](#) ()
- void [positionOnImageChanged](#) ()
- void [sizeChanged](#) ()
- void [sizeOnImageChanged](#) ()

### Protected Slots inherited from [Digikam::DImgChildItem](#)

- void [imageSizeChanged](#) (const QSizeF &)

### Protected Member Functions inherited from [Digikam::RegionFrameItem](#)

- bool [eventFilter](#) (QObject \*watched, QEvent \*event) override
- void [hoverEnterEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [hoverLeaveEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [hoverMoveEvent](#) (QGraphicsSceneHoverEvent \*event) override
- void [mouseMoveEvent](#) (QGraphicsSceneMouseEvent \*) override
- void [mousePressEvent](#) (QGraphicsSceneMouseEvent \*) override
- void [mouseReleaseEvent](#) (QGraphicsSceneMouseEvent \*) override
- void [paint](#) (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget=nullptr) override

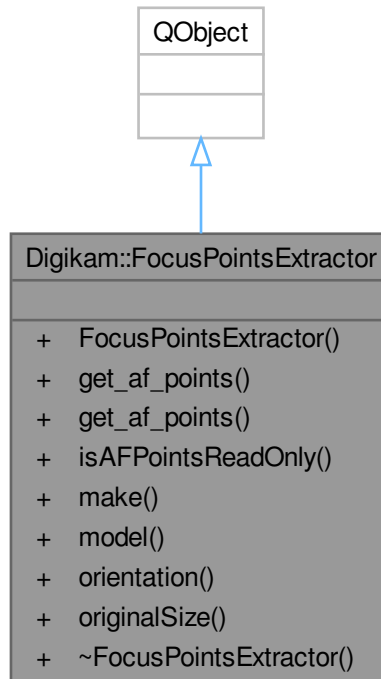
### Protected Member Functions inherited from [Digikam::DImgChildItem](#)

- QVariant [itemChange](#) (GraphicsItemChange change, const QVariant &value) override



## 6.668 Digikam::FocusPointsExtractor Class Reference

Inheritance diagram for Digikam::FocusPointsExtractor:



### Public Types

- using `ListAFPoints` = `QList< FocusPoint >`

### Public Member Functions

- `FocusPointsExtractor` (`QObject *const parent`, `const QString &path`)
- `ListAFPoints get_af_points ()`
- `ListAFPoints get_af_points (FocusPoint::TypePoint type)`
- `bool isAFPointsReadOnly () const`
- `QString make () const`
- `QString model () const`
- `MetaEngine::ImageOrientation orientation () const`
- `QSize originalSize () const`

## 6.668.1 Member Typedef Documentation

### 6.668.1.1 ListAFPoints

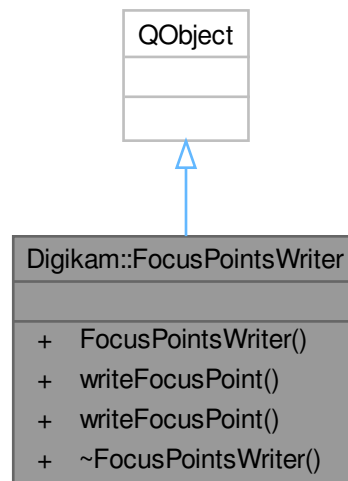
```
using Digikam::FocusPointsExtractor::ListAFPoints = QList<FocusPoint>
```

A list used to store focus points of a image extracted from meta data

With extract() function, an exiftool parser is used to read data from metadata and lists all focus points. Each focus point is defined by their relative centers coordinate and relative size. Each point has own type (Inactive, Infocus, Selected, SelectedInFocus)

## 6.669 Digikam::FocusPointsWriter Class Reference

Inheritance diagram for Digikam::FocusPointsWriter:



### Public Member Functions

- `FocusPointsWriter` (`QObject *const parent`, `const QString &path`)
- void `writeFocusPoint` (`const FocusPoint &point`)
- void `writeFocusPoint` (`const QRectF &rectF`)

## 6.670 Digikam::FrameOsd Class Reference

### Public Member Functions

- void `insertMessageOsdToFrame` (`QImage &frame`, `const QSize &JPEGsize`, `const QString &mess`)
- void `insertOsdToFrame` (`QImage &frame`, `const QUrl &url`, `const FrameOsdSettings &settings`, `const DInfoInterface *const info`)
- void `populateOSD` (`const QUrl &url`, `const FrameOsdSettings &settings`, `const DInfoInterface *const info`)
- void `printComments` (`const QString &comments`)
- void `printTags` (`QStringList &tags`)

## Public Attributes

- QString **m\_desc** = QLatin1String("")
- Qt::Alignment **m\_descAlign** = Qt::AlignLeft
- QColor **m\_descBg** = Qt::darkGray
- QFont **m\_descFnt** = QFont(QLatin1String("Monospace"))
- QPoint **m\_descPos** = QPoint(10, 10)

## 6.670.1 Member Function Documentation

### 6.670.1.1 insertMessageOsdToFrame()

```
void Digikam::FrameOsd::insertMessageOsdToFrame (
    QImage & frame,
    const QSize & JPEGsize,
    const QString & mess )
```

Insert message OSD on broken frame or end frame

### 6.670.1.2 insertOsdToFrame()

```
void Digikam::FrameOsd::insertOsdToFrame (
    QImage & frame,
    const QUrl & url,
    const FrameOsdSettings & settings,
    const DInfoInterface *const info )
```

Insert OSD on frame.

### 6.670.1.3 populateOSD()

```
void Digikam::FrameOsd::populateOSD (
    const QUrl & url,
    const FrameOsdSettings & settings,
    const DInfoInterface *const info )
```

Populate OSD items properties base on Url

### 6.670.1.4 printComments()

```
void Digikam::FrameOsd::printComments (
    const QString & comments )
```

print comments

### 6.670.1.5 printTags()

```
void Digikam::FrameOsd::printTags (
    QStringList & tags )
```

print tags

## 6.671 Digikam::FrameOsdSettings Class Reference

### Public Member Functions

- void [readSettings](#) (const KConfigGroup &group)
- void [writeSettings](#) (KConfigGroup &group)

### Public Attributes

- QFont [osdFont](#) = QFontDatabase::systemFont(QFontDatabase::GeneralFont)
- bool [printApertureFocal](#) = false  
*Print camera Aperture and Focal while streaming.*
- bool [printCapIfNoTitle](#) = false  
*Print image captions if no title available while streaming.*
- bool [printComment](#) = false  
*Print picture comment while streaming.*
- bool [printDate](#) = true  
*Print picture creation date while streaming.*
- bool [printExpoSensitivity](#) = false  
*Print camera Exposure and Sensitivity while streaming.*
- bool [printLensModel](#) = false  
*Print camera Lens model while streaming.*
- bool [printMakeModel](#) = false  
*Print camera Make and Model while streaming.*
- bool [printName](#) = true  
*Print picture file name while streaming.*
- bool [printRating](#) = false  
*Print rating while streaming.*
- bool [printTags](#) = false  
*Print tags title while streaming.*
- bool [printTitle](#) = false  
*Print image title while streaming.*

### 6.671.1 Member Function Documentation

#### 6.671.1.1 readSettings()

```
void Digikam::FrameOsdSettings::readSettings (
    const KConfigGroup & group )
```

Read and write settings in config file between sessions.

### 6.671.2 Member Data Documentation

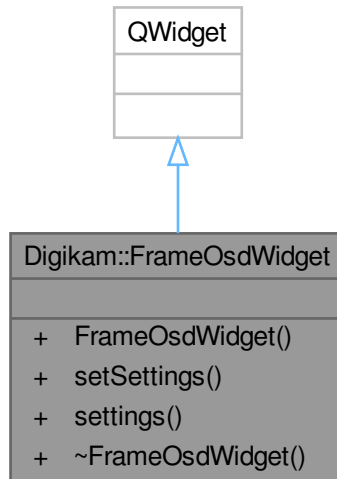
#### 6.671.2.1 osdFont

```
QFont Digikam::FrameOsdSettings::osdFont = QFontDatabase::systemFont (QFontDatabase::General↔
Font)
```

Font for the display of osd text.

## 6.672 Digikam::FrameOsdWidget Class Reference

Inheritance diagram for Digikam::FrameOsdWidget:



### Signals

- void **signalSettingsChanged** ()

### Public Member Functions

- **FrameOsdWidget** (QWidget \*const parent)
- void **setSettings** (const [FrameOsdSettings](#) &settings)
- [FrameOsdSettings](#) **settings** () const

## 6.673 Digikam::FrameUtils Class Reference

### Static Public Member Functions

- static QImage **makeFramedImage** (const QString &file, const QSize &outSize)
- static QImage **makeScaledImage** (QImage &img, const QSize &outSize)

## 6.674 Digikam::FreeRotationContainer Class Reference

### Public Types

- enum **AutoCropTypes** { **NoAutoCrop** = 0 , **WidestArea** , **LargestArea** }

**Public Attributes**

- double **angle** = 0.0
- bool **antiAlias** = true
- int **autoCrop** = NoAutoCrop
- QColor **backgroundColor** = Qt::black
- QSize **newSize**
- int **orgH** = 0
- int **orgW** = 0

## 6.675 Digikam::FreeRotationFilter Class Reference

Inheritance diagram for Digikam::FreeRotationFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **FreeRotationFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [FreeRotationContainer](#) &settings=[FreeRotationContainer](#)())
- **FreeRotationFilter** (QObject \*const parent=nullptr)
- QSize **getNewSize** () const
- void **readParameters** (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cancelFilter** ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > **multithreadedSteps** (int stop, int start=0) const
- virtual bool **parametersSuccessfullyRead** () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void **setFilterVersion** (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void **setupAndStartDirectly** (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void **setupFilter** (const [DImg](#) &orgImage)
- virtual void **startFilter** ()
- virtual void **startFilterDirectly** ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static double **calculateAngle** (const QPoint &p1, const QPoint &p2)
- static double **calculateAngle** (int x1, int y1, int x2, int y2)
- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }



## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.675.1 Member Function Documentation

### 6.675.1.1 filterAction()

`FilterAction Digikam::FreeRotationFilter::filterAction ( ) [override], [virtual]`

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.675.1.2 filterIdentifier()

`QString Digikam::FreeRotationFilter::filterIdentifier ( ) const [inline], [override], [virtual]`

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

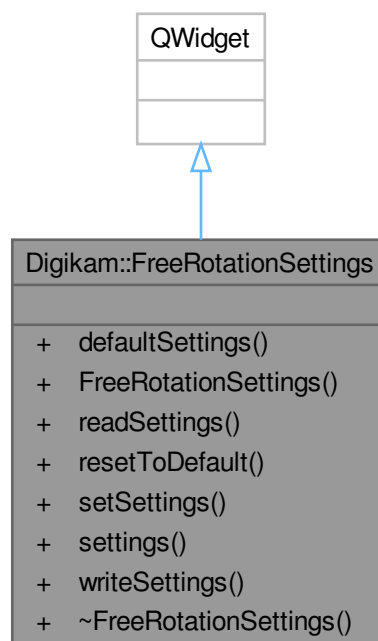
### 6.675.1.3 readParameters()

`void Digikam::FreeRotationFilter::readParameters ( const FilterAction & action ) [override], [virtual]`

Implements [Digikam::DImgThreadedFilter](#).

## 6.676 Digikam::FreeRotationSettings Class Reference

Inheritance diagram for Digikam::FreeRotationSettings:



## Signals

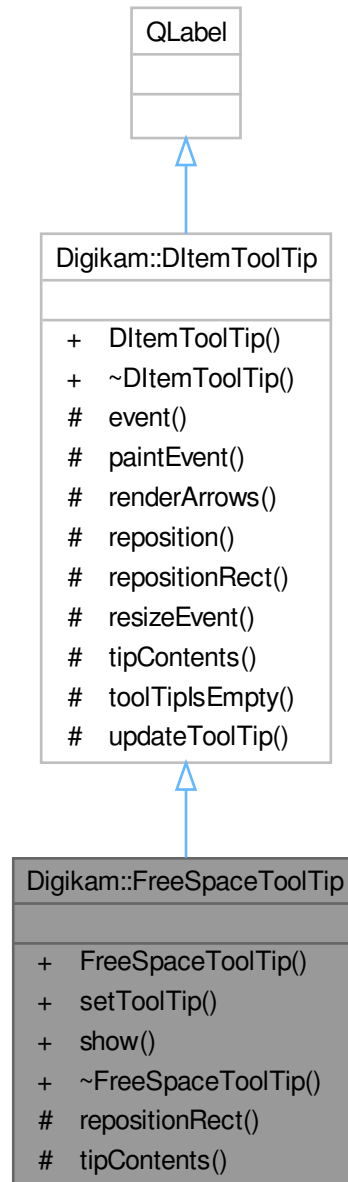
- void **signalSettingsChanged** ()

## Public Member Functions

- [FreeRotationContainer](#) **defaultSettings** () const
- **FreeRotationSettings** (QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const [FreeRotationContainer](#) &settings)
- [FreeRotationContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.677 Digikam::FreeSpaceToolTip Class Reference

Inheritance diagram for Digikam::FreeSpaceToolTip:



### Public Member Functions

- **FreeSpaceToolTip** (QWidget \*const parent)
- void **setToolTip** (const QString &tip)
- void **show** ()

## Public Member Functions inherited from [Digikam::DItemToolTip](#)

- **DItemToolTip** (QWidget \*const parent=nullptr)

## Protected Member Functions

- QRect [repositionRect](#) () override
- QString [tipContents](#) () override

## Protected Member Functions inherited from [Digikam::DItemToolTip](#)

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- void **resizeEvent** (QResizeEvent \*) override
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()

## 6.677.1 Member Function Documentation

### 6.677.1.1 [repositionRect\(\)](#)

QRect Digikam::FreeSpaceToolTip::repositionRect ( ) [override], [protected], [virtual]

Implements [Digikam::DItemToolTip](#).

### 6.677.1.2 [tipContents\(\)](#)

QString Digikam::FreeSpaceToolTip::tipContents ( ) [override], [protected], [virtual]

Implements [Digikam::DItemToolTip](#).

## 6.678 Digikam::FreeSpaceWidget Class Reference

Inheritance diagram for Digikam::FreeSpaceWidget:



### Public Types

- enum **FreeSpaceMode** { **AlbumLibrary** = 0 , **UMSCamera** , **GPhotoCamera** }

### Public Member Functions

- void **addInformation** (qint64 bytesSize, qint64 bytesUsed, qint64 bytesAvail, const QString &mountPoint)
- qint64 **bytesAvail** () const
- qint64 **bytesAvail** (const QString &path) const

- qint64 **bytesSize** () const
- qint64 **bytesUsed** () const
- qint64 **estimatedDSizeBytes** () const
- **FreeSpaceWidget** (QWidget \*const parent, int width)
- bool **isValid** () const
- int **percentUsed** () const
- void **refresh** ()
- void **setEstimatedDSizeBytes** (qint64 dSize)
- void **setMode** (FreeSpaceMode mode)
- void **setPath** (const QString &path)
- void **setPaths** (const QStringList &paths)

#### Protected Member Functions

- void **enterEvent** (QEnterEvent \*) override
- void **leaveEvent** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **updateToolTip** ()

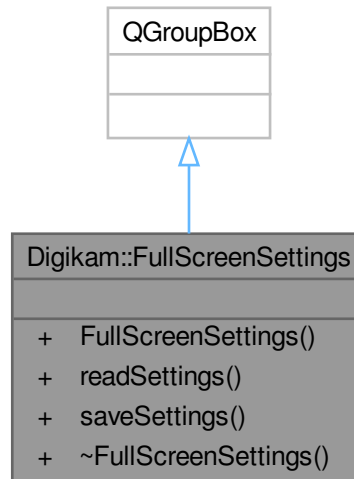
## 6.679 Digikam::FullObjectDetection Class Reference

#### Public Member Functions

- **FullObjectDetection** (const cv::Rect &rect\_)
- **FullObjectDetection** (const cv::Rect &rect\_, const std::vector< std::vector< float > > &parts\_)
- cv::Rect & **get\_rect** ()
- const cv::Rect & **get\_rect** () const
- unsigned long **num\_parts** () const
- std::vector< float > & **part** (unsigned long idx)
- const std::vector< float > & **part** (unsigned long idx) const

## 6.680 Digikam::FullScreenSettings Class Reference

Inheritance diagram for Digikam::FullScreenSettings:



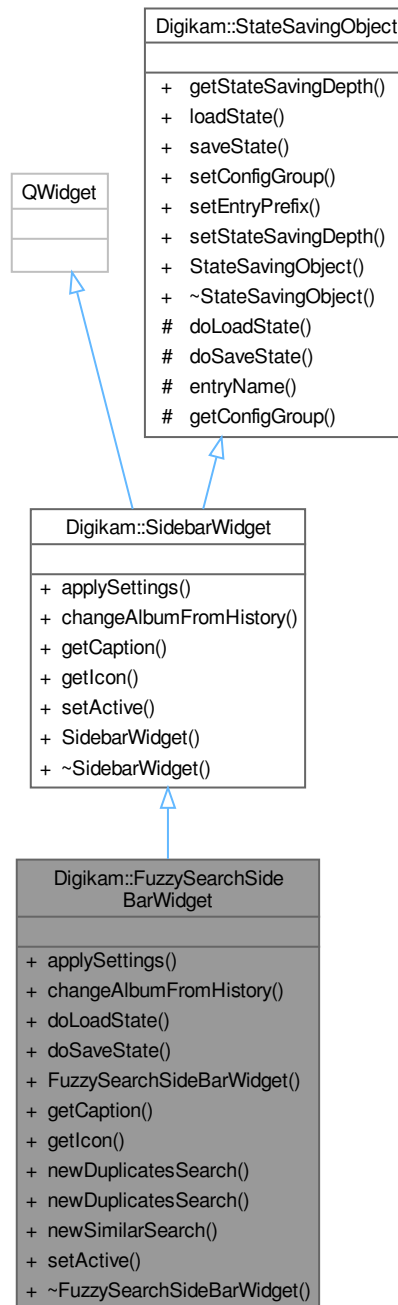
### Public Member Functions

- **FullScreenSettings** (int options, QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **saveSettings** (KConfigGroup &group)



## 6.681 Digikam::FuzzySearchSideBarWidget Class Reference

Inheritance diagram for Digikam::FuzzySearchSideBarWidget:



### Signals

- void **signalActive** (bool)

## Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget \\*](#))
- void [signalNotificationError](#) (const QString &message, int type)

## Public Member Functions

- void [applySettings](#) () override
- void [changeAlbumFromHistory](#) (const QList< [Album \\*](#) > &album) override
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- **FuzzySearchSideBarWidget** (QWidget \*const parent, [searchModel \\*](#)const searchModel, [SearchModificationHelper \\*](#)const searchModificationHelper)
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- void **newDuplicatesSearch** (const QList< [PAlbum \\*](#) > &albums)
- void **newDuplicatesSearch** (const QList< [TAlbum \\*](#) > &albums)
- void **newSimilarSearch** (const [ItemInfo](#) &imageInfo)
- void [setActive](#) (bool active) override

## Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.681.1 Member Function Documentation

### 6.681.1.1 applySettings()

```
void Digikam::FuzzySearchSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).

### 6.681.1.2 changeAlbumFromHistory()

```
void Digikam::FuzzySearchSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.681.1.3 doLoadState()

```
void Digikam::FuzzySearchSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.681.1.4 doSaveState()

```
void Digikam::FuzzySearchSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.681.1.5 getCaption()

```
const QString Digikam::FuzzySearchSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.681.1.6 `getIcon()`

```
const QIcon Digikam::FuzzySearchSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.681.1.7 `setActive()`

```
void Digikam::FuzzySearchSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

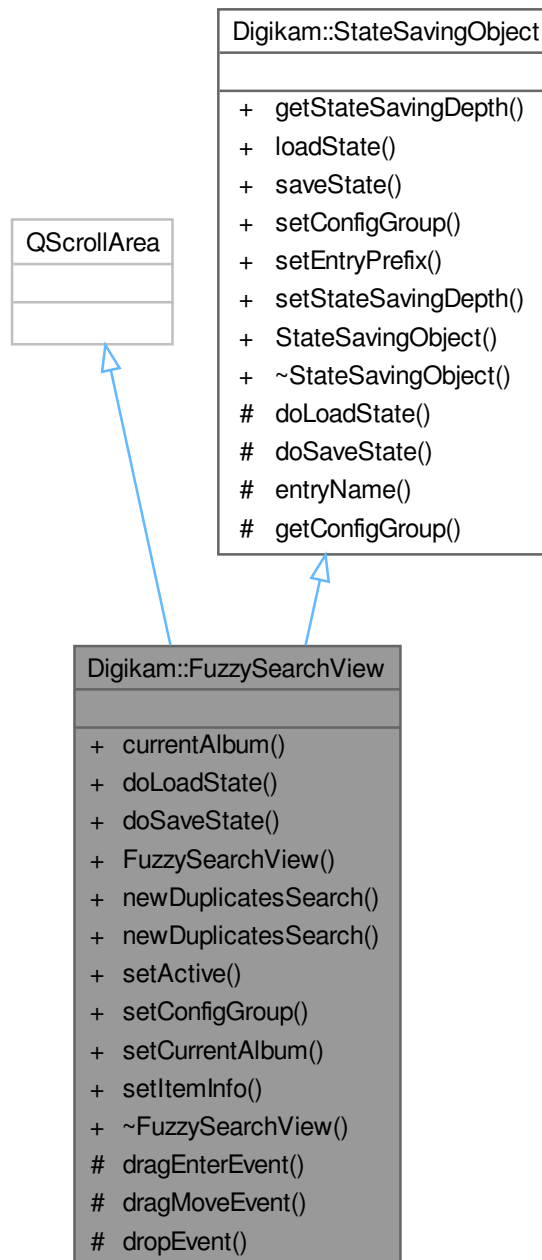
## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.682 Digikam::FuzzySearchView Class Reference

Inheritance diagram for Digikam::FuzzySearchView:



## Classes

- class [Private](#)

## Signals

- void **signalNotificationError** (const QString &message, int type)

## Public Member Functions

- [SAlbum](#) \* **currentAlbum** () const
- void **doLoadState** () override
- void **doSaveState** () override
- **FuzzySearchView** ([searchModel](#) \*const searchModel, [SearchModificationHelper](#) \*const searchModificationHelper, QWidget \*const parent=nullptr)
- void **newDuplicatesSearch** (const QList< [PAAlbum](#) \* > &albums)
- void **newDuplicatesSearch** (const QList< [TAAlbum](#) \* > &albums)
- void **setActive** (bool val)
- void **setConfigGroup** (const KConfigGroup &group) override
- void **setCurrentAlbum** ([SAlbum](#) \*const album)
- void **setItemInfo** (const [ItemInfo](#) &info)

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Protected Member Functions

- void **dragEnterEvent** (QDragEnterEvent \*e) override
- void **dragMoveEvent** (QDragMoveEvent \*e) override
- void **dropEvent** (QDropEvent \*e) override

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString **entryName** (const QString &base) const
- KConfigGroup **getConfigGroup** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## 6.682.1 Member Function Documentation

### 6.682.1.1 doLoadState()

```
void Digikam::FuzzySearchView::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.682.1.2 doSaveState()

```
void Digikam::FuzzySearchView::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.682.1.3 setConfigGroup()

```
void Digikam::FuzzySearchView::setConfigGroup (
    const KConfigGroup & group ) [override], [virtual]
```

Sets a dedicated config group that will be used to store and reload the state from. If this method is not called, a group based on the object name is used.

You can re-implement this method to pass the group set here to child objects. Don't forget to call this method in your implementation.

#### Parameters

<i>group</i>	config group to use for state saving and restoring
--------------	--

Reimplemented from [Digikam::StateSavingObject](#).

## 6.683 Digikam::FuzzySearchView::Private Class Reference

### Public Types

- enum **FuzzySearchTab** { **DUPLICATES** = 0 , **SIMILARS** , **SKETCH** }

### Public Attributes

- bool **active** = false  
*initially be active to update sketch panel when the search list is restored*
- const QString **configPenSketchHueEntry** = QLatin1String("Pen Sketch Hue")
- const QString **configPenSketchSaturationEntry** = QLatin1String("Pen Sketch Saturation")

- const QString **configPenSketchSizeEntry** = QLatin1String("Pen Sketch Size")
- const QString **configPenSkethValueEntry** = QLatin1String("Pen Sketch Value")
- const QString **configResultSketchItemsEntry** = QLatin1String("Result Sketch items")
- const QString **configSimilarMaxThresholdEntry** = QLatin1String("Similar Maximum Threshold")
- const QString **configSimilarThresholdEntry** = QLatin1String("Similar Threshold")
- const QString **configTabEntry** = QLatin1String("FuzzySearch Tab")
- [FindDuplicatesView](#) \* **findDuplicatesPanel** = nullptr
- bool **fingerprintsChecked** = false
- [DVBox](#) \* **folderView** = nullptr
- [AlbumSelectors](#) \* **fuzzySearchAlbumSelectors** = nullptr
- [DHueSaturationSelector](#) \* **hsSelector** = nullptr
- [ItemInfo](#) **imageInfo**
- [AlbumPointer](#)< [SAlbum](#) > **imageSAlbum**
- [QUrl](#) **imageUrl**
- [QLabel](#) \* **imageWidget** = nullptr
- [DAdjustableLabel](#) \* **labelFile** = nullptr
- [DAdjustableLabel](#) \* **labelFolder** = nullptr
- [DTextEdit](#) \* **nameEditImage** = nullptr
- [DTextEdit](#) \* **nameEditSketch** = nullptr
- [QSpinBox](#) \* **penSize** = nullptr
- [QToolButton](#) \* **redoBtnSketch** = nullptr
- [QToolButton](#) \* **resetButton** = nullptr
- [QSpinBox](#) \* **resultsSketch** = nullptr
- [QToolButton](#) \* **saveBtnImage** = nullptr
- [QToolButton](#) \* **saveBtnSketch** = nullptr
- [SearchTextBarDb](#) \* **searchFuzzyBar** = nullptr
- [SearchModel](#) \* **searchModel** = nullptr
- [SearchModificationHelper](#) \* **searchModificationHelper** = nullptr
- [EditableSearchTreeView](#) \* **searchTreeView** = nullptr
- [QColor](#) **selColor**
- [ApplicationSettings](#) \* **settings** = nullptr
- [DIntRangeBox](#) \* **similarityRange** = nullptr
- [AlbumPointer](#)< [SAlbum](#) > **sketchSAlbum**
- [AlbumSelectors](#) \* **sketchSearchAlbumSelectors** = nullptr
- [SketchWidget](#) \* **sketchWidget** = nullptr
- [QTabWidget](#) \* **tabWidget** = nullptr
- [ThumbnailLoadThread](#) \* **thumbLoadThread** = nullptr
- [QTimer](#) \* **timerImage** = nullptr
- [QTimer](#) \* **timerSketch** = nullptr
- [QToolButton](#) \* **undoBtnSketch** = nullptr
- [DColorValueSelector](#) \* **vSelector** = nullptr

## 6.684 Digikam::GeoCoordinates Class Reference

### Public Types

- enum **HasFlag** {  
**HasNothing** = 0 , **HasLatitude** = 1 , **HasLongitude** = 2 , **HasCoordinates** = 3 ,  
**HasAltitude** = 4 }
- typedef QList< [GeoCoordinates](#) > **List**
- typedef QPair< [GeoCoordinates](#), [GeoCoordinates](#) > **Pair**
- typedef QList< [GeoCoordinates::Pair](#) > **PairList**



**Public Member Functions**

- double **alt** () const
- QString **altString** () const
- void **clear** ()
- void **clearAlt** ()
- **GeoCoordinates** (const double inLat, const double inLon)
- **GeoCoordinates** (const double inLat, const double inLon, const double inAlt)
- QString **geoUrl** () const
- bool **hasAltitude** () const
- bool **hasCoordinates** () const
- HasFlags **hasFlags** () const
- bool **hasLatitude** () const
- bool **hasLongitude** () const
- double **lat** () const
- QString **latString** () const
- double **lon** () const
- QString **lonString** () const
- bool **operator==** (const [GeoCoordinates](#) &other) const
- bool **sameLonLatAs** (const [GeoCoordinates](#) &other) const
- void **setAlt** (const double inAlt)
- void **setLatLon** (const double inLat, const double inLon)

**Static Public Member Functions**

- static [GeoCoordinates](#) **fromGeoUrl** (const QString &url, bool \*const parsedOkay=nullptr)
- static Pair **makePair** (const qreal lat1, const qreal lon1, const qreal lat2, const qreal lon2)

**6.685 Digikam::GeodeticCalculator Class Reference****Public Member Functions**

- double [azimuth](#) ()
- bool [checkOrthodromicDistance](#) ()
- bool [computeDestinationPoint](#) ()
- bool [computeDirection](#) ()
- QPointF **destinationGeographicPoint** ()
- bool [destinationGeographicPoint](#) (double \*longitude, double \*latitude)
- [Ellipsoid](#) [ellipsoid](#) () const
- [GeodeticCalculator](#) (const [Ellipsoid](#) &e=[Ellipsoid::WGS84](#)())
- double [meridianArcLength](#) (double latitude1, double latitude2)
- double [meridianArcLengthRadians](#) (double P1, double P2)
- double [orthodromicDistance](#) ()
- void [setDestinationGeographicPoint](#) (double longitude, double latitude)
- void [setDirection](#) (double [azimuth](#), double distance)
- void [setStartingGeographicPoint](#) (double longitude, double latitude)

**Protected Member Functions**

- double **castToAngleRange** (const double alpha)
- bool [checkAzimuth](#) (double \*azimuth)
- bool [checkLatitude](#) (double \*latitude)
- bool [checkLongitude](#) (double \*longitude)
- bool [checkOrthodromicDistance](#) (const double distance)

## Protected Attributes

- double **a01** = 0.0
- double **a02** = 0.0
- double **a03** = 0.0
- double **a21** = 0.0
- double **a22** = 0.0
- double **a23** = 0.0
- double **a42** = 0.0
- double **a43** = 0.0
- double **a63** = 0.0
- double **f** = 0.0
- double **f2** = 0.0
- double **f3** = 0.0
- double **f4** = 0.0
- double **fo** = 0.0
- double **m\_A** = 0.0
- double **m\_azimuth** = 0.0
- double **m\_B** = 0.0
- double **m\_C** = 0.0
- double **m\_D** = 0.0
- bool **m\_destinationValid** = false
- bool **m\_directionValid** = false
- double **m\_distance** = 0.0
- double **m\_E** = 0.0
- double **m\_eccentricitySquared** = 0.0
- **Ellipsoid m\_ellipsoid**
- double **m\_F** = 0.0
- double **m\_lat1** = 0.0
- double **m\_lat2** = 0.0
- double **m\_long1** = 0.0
- double **m\_long2** = 0.0
- double **m\_maxOrthodromicDistance** = 0.0
- double **m\_semiMajorAxis** = 0.0
- double **m\_semiMinorAxis** = 0.0
- double **m\_TOLERANCE\_0** = 5.0e-15
- double **m\_TOLERANCE\_1** = 5.0e-14
- double **m\_TOLERANCE\_2** = 5.0e-13
- double **m\_TOLERANCE\_3** = 7.0e-3
- double **m\_TOLERANCE\_CHECK** = 1E-8
- double **T1** = 1.0
- double **T2** = 0.0
- double **T4** = 0.0
- double **T6** = 0.0

## 6.685.1 Constructor & Destructor Documentation

### 6.685.1.1 GeodeticCalculator()

```
Digikam::GeodeticCalculator::GeodeticCalculator (
    const Ellipsoid & e = Ellipsoid::WGS84() ) [explicit]
```

Performs geodetic calculations on an ellipsoid. This class encapsulates a generic ellipsoid and calculates the following properties:

Distance and azimuth between two points. Point located at a given distance and azimuth from an other point.

The calculation use the following information:

The starting position (`setStartingPosition`), which is always considered valid. It is initially set at (0,0) and can only be changed to another legitimate value. Only one of the following:

```
The destination position (setDestinationPosition), or  
An azimuth and distance (setDirection).
```

The latest one set overrides the other and determines what will be calculated.

## 6.685.2 Member Function Documentation

### 6.685.2.1 azimuth()

```
double Digikam::GeodeticCalculator::azimuth ( )
```

Returns the azimuth. This method returns the value set by the last call to `setDirection(double, double)` `setDirection(azimuth, distance)`, except if `setDestinationGeographicPoint(double, double)` `setDestinationGeographicPoint(...)` has been invoked after. In this later case, the azimuth will be computed from the startingGeographicPoint starting point to the destination point.

#### Returns

The azimuth, in decimal degrees from -180° to +180°.

### 6.685.2.2 checkAzimuth()

```
bool Digikam::GeodeticCalculator::checkAzimuth (
    double * azimuth ) [protected]
```

Checks the azimuth validity. The argument `azimuth` should be greater or equal than -180 degrees and lower or equals than +180 degrees. As a convenience, this method converts the azimuth to radians.

#### Parameters

<code>azimuth</code>	The azimuth value in decimal degrees.
----------------------	---------------------------------------

### 6.685.2.3 checkLatitude()

```
bool Digikam::GeodeticCalculator::checkLatitude (
    double * latitude ) [protected]
```

Checks the latitude validity. The argument `latitude` should be greater or equal than -90 degrees and lower or equals than +90 degrees. As a convenience, this method converts the latitude to radians.

## Parameters

<i>latitude</i>	The latitude value in decimal degrees.
-----------------	--

**6.685.2.4 checkLongitude()**

```
bool Digikam::GeodeticCalculator::checkLongitude (
    double * longitude ) [protected]
```

Checks the longitude validity. The argument `longitude` should be greater or equal than -180 degrees and lower or equals than +180 degrees. As a convenience, this method converts the longitude to radians.

## Parameters

<i>longitude</i>	The longitude value in decimal degrees.
------------------	---

**6.685.2.5 checkOrthodromicDistance() [1/2]**

```
bool Digikam::GeodeticCalculator::checkOrthodromicDistance ( )
```

Computes the orthodromic distance using the algorithm implemented in the Geotools's ellipsoid class (if available), and check if the error is smaller than some tolerance error.

**6.685.2.6 checkOrthodromicDistance() [2/2]**

```
bool Digikam::GeodeticCalculator::checkOrthodromicDistance (
    const double distance ) [protected]
```

Checks the orthodromic distance validity. Arguments `orthodromicDistance` should be greater or equal than 0 and lower or equals than the maximum orthodromic distance.

## Parameters

<i>distance</i>	The orthodromic distance value.
-----------------	---------------------------------

**6.685.2.7 computeDestinationPoint()**

```
bool Digikam::GeodeticCalculator::computeDestinationPoint ( )
```

Computes the destination point from the starting point, the azimuth and the orthodromic distance.

**6.685.2.8 computeDirection()**

```
bool Digikam::GeodeticCalculator::computeDirection ( )
```

Computes the azimuth and orthodromic distance from the startingGeographicPoint starting point and the destinationGeographicPoint destination point.

Computes the azimuth and orthodromic distance from the startingGeographicPoint() and the destinationGeographicPoint().

### 6.685.2.9 destinationGeographicPoint()

```
bool Digikam::GeodeticCalculator::destinationGeographicPoint (
    double * longitude,
    double * latitude )
```

Returns the destination point. This method returns the point set by the last call to a setDestinationGeographicPoint(...) method, except if setDirection(...) has been invoked after. In this later case, the destination point will be computed from the starting point to the azimuth and distance specified. Coordinates positive North and East.

#### Returns

The destination point. The x and y coordinates are the longitude and latitude in decimal degrees, respectively.

### 6.685.2.10 ellipsoid()

```
Ellipsoid Digikam::GeodeticCalculator::ellipsoid ( ) const
```

Returns the referenced ellipsoid.

### 6.685.2.11 meridianArcLength()

```
double Digikam::GeodeticCalculator::meridianArcLength (
    double latitude1,
    double latitude2 )
```

Calculates the meridian arc length between two points in the same meridian in the referenced ellipsoid.

#### Parameters

<i>latitude1</i>	The latitude of the first point (in decimal degrees).
<i>latitude2</i>	The latitude of the second point (in decimal degrees).

#### Returns

Returned the meridian arc length between latitude1 and latitude2

### 6.685.2.12 meridianArcLengthRadians()

```
double Digikam::GeodeticCalculator::meridianArcLengthRadians (
    double P1,
    double P2 )
```

Calculates the meridian arc length between two points in the same meridian in the referenced ellipsoid.

**Parameters**

<i>P1</i>	The latitude of the first point (in radians).
<i>P2</i>	The latitude of the second point (in radians).

**Returns**

Returned the meridian arc length between P1 and P2

**6.685.2.13 orthodromicDistance()**

```
double Digikam::GeodeticCalculator::orthodromicDistance ( )
```

Returns the orthodromic distance. This method returns the value set by the last call to `setDirection(double, double)` `setDirection(azimuth, distance)`, except if `setDestinationGeographicPoint(double, double)` `setDestinationGeographicPoint(...)` has been invoked after. In this later case, the distance will be computed from the startingGeographicPoint starting point to the destination point.

**Returns**

The orthodromic distance, in the same units as the getEllipsoid ellipsoid axis.

**6.685.2.14 setDestinationGeographicPoint()**

```
void Digikam::GeodeticCalculator::setDestinationGeographicPoint (
    double longitude,
    double latitude )
```

Set the destination point in geographic coordinates. The azimuth and distance values will be updated as a side effect of this call. They will be recomputed the next time `getAzimuth()` or `getOrthodromicDistance()` are invoked. Coordinates positive North and East.

**Parameters**

<i>longitude</i>	The longitude in decimal degrees between -180 and +180°
<i>latitude</i>	The latitude in decimal degrees between -90 and +90°

**6.685.2.15 setDirection()**

```
void Digikam::GeodeticCalculator::setDirection (
    double azimuth,
    double distance )
```

Set the azimuth and the distance from the startingGeographicPoint starting point. The destination point will be updated as a side effect of this call. It will be recomputed the next time `destinationGeographicPoint()` is invoked. Azimuth 0° North.

## Parameters

<i>azimuth</i>	The azimuth in decimal degrees from -180° to 180°.
<i>distance</i>	The orthodromic distance in the same units as the ellipsoid axis.

**6.685.2.16 setStartingGeographicPoint()**

```
void Digikam::GeodeticCalculator::setStartingGeographicPoint (
    double longitude,
    double latitude )
```

Set the starting point in geographic coordinates. The azimuth, the orthodromic distance and the destination point are discarded. They will need to be specified again. Coordinates positive North and East.

## Parameters

<i>longitude</i>	The longitude in decimal degrees between -180 and +180°
<i>latitude</i>	The latitude in decimal degrees between -90 and +90°

**6.685.3 Member Data Documentation****6.685.3.1 a01**

```
double Digikam::GeodeticCalculator::a01 = 0.0 [protected]
```

Parameters computed from the ellipsoid.

**6.685.3.2 fo**

```
double Digikam::GeodeticCalculator::fo = 0.0 [protected]
```

GPNHRI parameters computed from the ellipsoid.

$f$  if the flattening of the referenced ellipsoid.  $f_2$ ,  $f_3$  and  $f_4$  are  $f^2$ ,  $f^3$  and  $f^4$  respectively.

**6.685.3.3 m\_A**

```
double Digikam::GeodeticCalculator::m_A = 0.0 [protected]
```

GPNARC parameters computed from the ellipsoid.

**6.685.3.4 m\_destinationValid**

```
bool Digikam::GeodeticCalculator::m_destinationValid = false [protected]
```

Tell if the destination point is valid. `false` if long2 and lat2 need to be computed.

### 6.685.3.5 m\_directionValid

```
bool Digikam::GeodeticCalculator::m_directionValid = false [protected]
```

Tell if the azimuth and the distance are valids. `false` if distance and azimuth need to be computed.

### 6.685.3.6 m\_distance

```
double Digikam::GeodeticCalculator::m_distance = 0.0 [protected]
```

The distance and azimuth (in radians) from the starting point (long1, lat1) to the destination point (long2, lat2).

### 6.685.3.7 m\_eccentricitySquared

```
double Digikam::GeodeticCalculator::m_eccentricitySquared = 0.0 [protected]
```

The eccentricity squared of the referenced ellipsoid.

### 6.685.3.8 m\_ellipsoid

```
Ellipsoid Digikam::GeodeticCalculator::m_ellipsoid [protected]
```

The encapsulated ellipsoid.

### 6.685.3.9 m\_lat1

```
double Digikam::GeodeticCalculator::m_lat1 = 0.0 [protected]
```

The (*latitude*, *longitude*) coordinate of the first point in radians. This point is set by `setStartingGeographicPoint`.

### 6.685.3.10 m\_lat2

```
double Digikam::GeodeticCalculator::m_lat2 = 0.0 [protected]
```

The (*latitude*, *longitude*) coordinate of the destination point in radians. This point is set by `setDestinationGeographicPoint`.

### 6.685.3.11 m\_maxOrthodromicDistance

```
double Digikam::GeodeticCalculator::m_maxOrthodromicDistance = 0.0 [protected]
```

The maximum orthodromic distance that could be calculated onto the referenced ellipsoid.



**6.685.3.12 m\_semiMajorAxis**

```
double Digikam::GeodeticCalculator::m_semiMajorAxis = 0.0 [protected]
```

The semi major axis of the referenced ellipsoid.

**6.685.3.13 m\_semiMinorAxis**

```
double Digikam::GeodeticCalculator::m_semiMinorAxis = 0.0 [protected]
```

The semi minor axis of the referenced ellipsoid.

**6.685.3.14 m\_TOLERANCE\_0**

```
double Digikam::GeodeticCalculator::m_TOLERANCE_0 = 5.0e-15 [protected]
```

Tolerance factors from the strictest (TOLERANCE\_0) to the most relax one (TOLERANCE\_3).

**6.685.3.15 m\_TOLERANCE\_CHECK**

```
double Digikam::GeodeticCalculator::m_TOLERANCE_CHECK = 1E-8 [protected]
```

Tolerance factor for assertions. It has no impact on computed values.

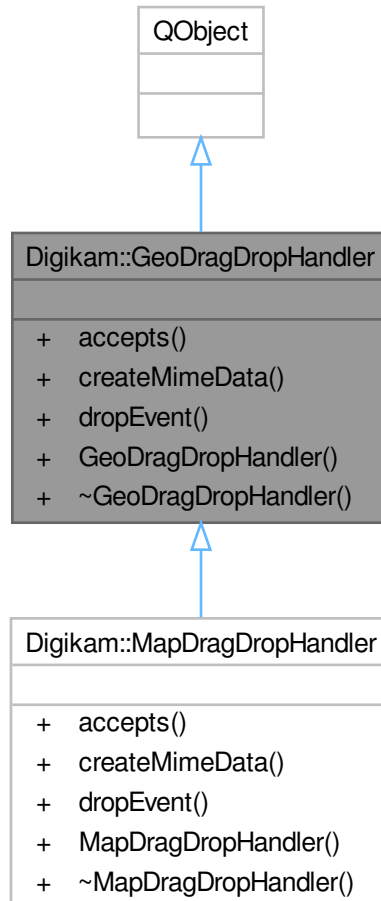
**6.685.3.16 T1**

```
double Digikam::GeodeticCalculator::T1 = 1.0 [protected]
```

Parameters computed from the ellipsoid.

## 6.686 Digikam::GeoDragDropHandler Class Reference

Inheritance diagram for Digikam::GeoDragDropHandler:



### Public Member Functions

- virtual Qt::DropAction **accepts** (const QDropEvent \*e)=0
- virtual QMimeData \* **createMimeData** (const QList< QPersistentModelIndex > &modelIndices)=0
- virtual bool **dropEvent** (const QDropEvent \*e, const [GeoCoordinates](#) &dropCoordinates)=0
- **GeoDragDropHandler** (QObject \*const parent=nullptr)

## 6.687 Digikam::GeofaceCluster Class Reference

### Public Types

- typedef QList< [GeofaceCluster](#) > **List**
- enum **PixmapType** { **PixmapMarker** , **PixmapCircle** , **PixmapImage** }

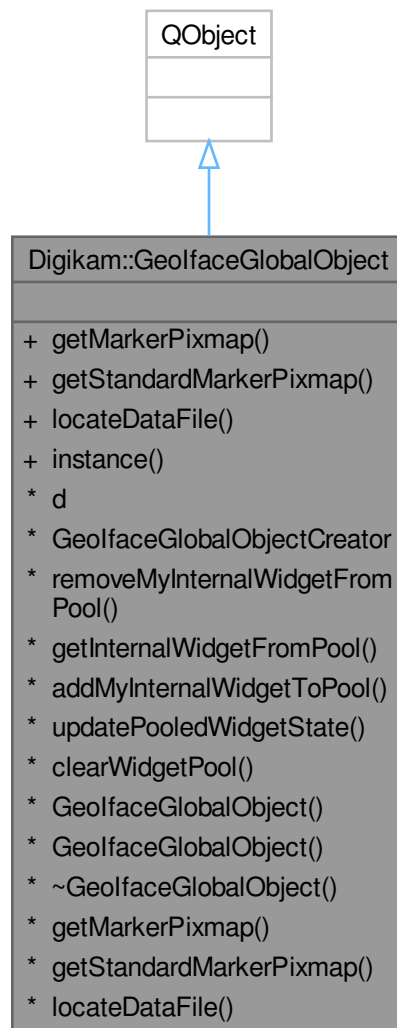
### Public Attributes

- [GeoCoordinates](#) **coordinates**
- GeoGroupState **groupState**
- int **markerCount**
- int **markerSelectedCount**
- QPoint **pixelPos**
- QPoint **pixmapOffset**  
*anchor point of the image, measured from bottom-left*
- QSize **pixmapSize**
- enum Digikam::GeolfaceCluster::PixmapType **pixmapType**
- QMap< int, QVariant > **representativeMarkers**
- QList< [TileIndex](#) > **tileIndicesList**

## 6.688 Digikam::GeolfaceGlobalObject Class Reference

Global object for geolocation interface to hold items common to all geolocation interface Widget instances.

Inheritance diagram for Digikam::GeolfaceGlobalObject:



## Public Member Functions

### Shared pixmaps

- QPixmap **getMarkerPixmap** (const QString &pixmapId)
- QPixmap **getStandardMarkerPixmap** ()
- QUrl **locateDataFile** (const QString &filename)

## Static Public Member Functions

- static [GeolfaceGlobalObject](#) \* **instance** ()

### Shared internal map widgets

- class **GeolfaceGlobalObjectCreator**
- void **removeMyInternalWidgetFromPool** (const [MapBackend](#) \*const mapBackend)
- bool **getInternalWidgetFromPool** (const [MapBackend](#) \*const mapBackend, [GeolfaceInternalWidgetInfo](#) \*const targetInfo)
- void **addMyInternalWidgetToPool** (const [GeolfaceInternalWidgetInfo](#) &info)
- void **updatePooledWidgetState** (const QWidget \*const widget, const [GeolfaceInternalWidgetInfo](#)::[InternalWidgetState](#) newState)
- void **clearWidgetPool** ()

## 6.689 Digikam::GeolfaceInternalWidgetInfo Class Reference

Class to hold information about map widgets stored in the [GeolfaceGlobalObject](#).

### Public Types

- typedef void(\* **DeleteFunction**) ([GeolfaceInternalWidgetInfo](#) \*const info)
- enum **InternalWidgetState** { **InternalWidgetReleased** = 1 , **InternalWidgetUndocked** = 2 , **InternalWidgetStillDocked** = 4 }

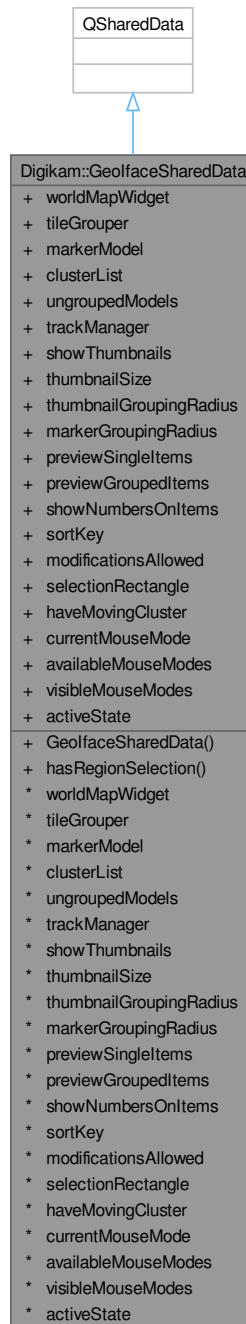
### Public Attributes

- QVariant **backendData**
- QString **backendName**
- QPointer< QObject > **currentOwner**
- DeleteFunction **deleteFunction**
- InternalWidgetStates **state**
- QPointer< QWidget > **widget**

## 6.689.1 Detailed Description

## 6.690 Digikam::GeolfaceSharedData Class Reference

Inheritance diagram for Digikam::GeolfaceSharedData:



### Public Member Functions

- bool [hasRegionSelection](#) () const

## Public Attributes

### Objects

- [MapWidget](#) \* **worldMapWidget**
- [TileGrouper](#) \* **tileGrouper**
- [AbstractMarkerTiler](#) \* **markerModel**
- [GeofaceCluster::List](#) **clusterList**
- [QList< GeoModelHelper \\* >](#) **ungroupedModels**
- [TrackManager](#) \* **trackManager**

### Display options

- bool **showThumbnails**
- int **thumbnailSize**
- int **thumbnailGroupingRadius**
- int **markerGroupingRadius**
- bool **previewSingleItems**
- bool **previewGroupedItems**
- bool **showNumbersOnItems**
- int **sortKey**
- bool **modificationsAllowed**

### Current map state

- [GeoCoordinates::Pair](#) **selectionRectangle**
- bool **haveMovingCluster**
- [GeoMouseModes](#) **currentMouseMode**
- [GeoMouseModes](#) **availableMouseModes**
- [GeoMouseModes](#) **visibleMouseModes**
- bool **activeState**

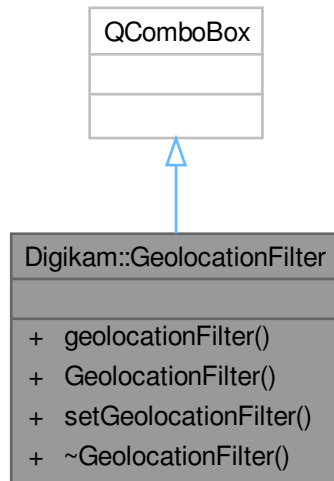
## 6.690.1 Member Function Documentation

### 6.690.1.1 hasRegionSelection()

```
bool Digikam::GeoIfaceSharedData::hasRegionSelection ( ) const [inline]
```

## 6.691 Digikam::GeolocationFilter Class Reference

Inheritance diagram for Digikam::GeolocationFilter:



### Signals

- void **signalFilterChanged** (const [ItemFilterSettings::GeolocationCondition](#) &condition)

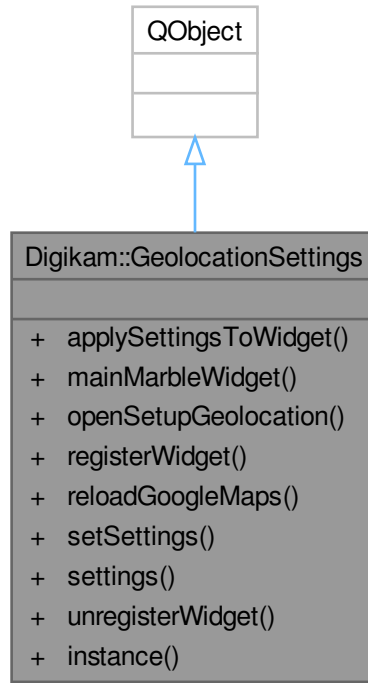
### Public Member Functions

- [ItemFilterSettings::GeolocationCondition](#) **geolocationFilter** () const
- **GeolocationFilter** (QWidget \*const parent)
- void **setGeolocationFilter** (const [ItemFilterSettings::GeolocationCondition](#) &condition)



## 6.692 Digikam::GeolocationSettings Class Reference

Inheritance diagram for Digikam::GeolocationSettings:



### Signals

- void **signalGeolocationSettingsChanged** (const [GeolocationSettingsContainer](#) &current, const [GeolocationSettingsContainer](#) &previous)
- void **signalSettingsChanged** ()
- void **signalSetupGeolocation** (int tab)

### Public Member Functions

- void **applySettingsToWidget** ([MapWidget](#) \*const widget)
- [MarbleWidget](#) \* **mainMarbleWidget** () const
- void **openSetupGeolocation** ([SetupGeolocation::GeolocationTab](#) tab)
- void **registerWidget** ([MapWidget](#) \*const widget)
- void **reloadGoogleMaps** ()
- void **setSettings** (const [GeolocationSettingsContainer](#) &settings)
- [GeolocationSettingsContainer](#) **settings** () const
- void **unregisterWidget** ([MapWidget](#) \*const widget)

### Static Public Member Functions

- static [GeolocationSettings](#) \* **instance** ()

## Friends

- class **GeolocationSettingsCreator**

## 6.692.1 Member Function Documentation

### 6.692.1.1 applySettingsToWidget()

```
void Digikam::GeolocationSettings::applySettingsToWidget (
    MapWidget *const widget )
```

Apply the current settings to a previously registered [MapWidget](#).

### 6.692.1.2 instance()

```
GeolocationSettings * Digikam::GeolocationSettings::instance ( ) [static]
```

Global container for Metadata settings. All accessor methods are thread-safe.

### 6.692.1.3 mainMarbleWidget()

```
MarbleWidget * Digikam::GeolocationSettings::mainMarbleWidget ( ) const
```

Return the first registered MarbleWidget instance stored in the collection. If no valid instance is found, nullptr is returned.

### 6.692.1.4 registerWidget()

```
void Digikam::GeolocationSettings::registerWidget (
    MapWidget *const widget )
```

Store one [MapWidget](#) instance in the collection.

### 6.692.1.5 setSettings()

```
void Digikam::GeolocationSettings::setSettings (
    const GeolocationSettingsContainer & settings )
```

Sets the current Metadata settings and writes them to config.

### 6.692.1.6 settings()

```
GeolocationSettingsContainer Digikam::GeolocationSettings::settings ( ) const
```

Returns the current Metadata settings.

### 6.692.1.7 unregisterWidget()

```
void Digikam::GeolocationSettings::unregisterWidget (
    MapWidget *const widget )
```

Remove one [MapWidget](#) instance in the collection.

## 6.693 Digikam::GeolocationSettingsContainer Class Reference

### Public Member Functions

- void **readFromConfig** (const KConfigGroup &group)
- void **writeToConfig** (KConfigGroup &group) const

### Public Attributes

- Marble::AngleUnit **angleUnit** = Marble::DecimalDegree
- Marble::MapQuality **animationQuality** = Marble::LowQuality
- MarbleLocale::MeasurementSystem **distanceUnit** = MarbleLocale::MetricSystem
- bool **inertialRotation** = true
- QFont **mapFont**
- bool **mouseRotation** = true
- int **persistentTileCacheLimit** = 999999
- bool **showAtmos** = false
- bool **showCities** = true
- bool **showCross** = true
- bool **showGrid** = true
- bool **showRelief** = true
- bool **showSunShading** = false
- Marble::MapQuality **stillQuality** = Marble::HighQuality
- int **volatileTileCacheLimit** = 100

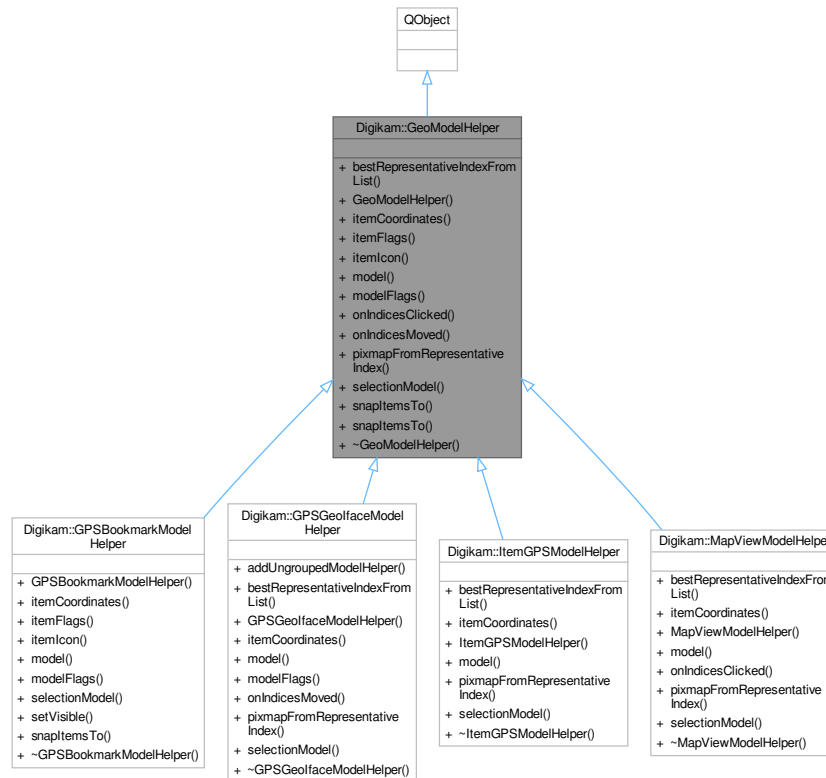
### 6.693.1 Detailed Description

The class [GeolocationSettingsContainer](#) encapsulates all Marble related settings.

## 6.694 Digikam::GeoModelHelper Class Reference

Helper class to access data in models.

Inheritance diagram for Digikam::GeoModelHelper:



### Public Types

- enum **PropertyFlag** { **FlagNull** = 0 , **FlagVisible** = 1 , **FlagMovable** = 2 , **FlagSnaps** = 4 }

### Signals

- void **signalModelChangedDrastically** ()
- void **signalThumbnailAvailableForIndex** (const QPersistentModelIndex &index, const QPixmap &pixmap)
- void **signalVisibilityChanged** ()

### Public Member Functions

- virtual QPersistentModelIndex **bestRepresentativeIndexFromList** (const QList< QPersistentModelIndex > &list, const int sortKey)
- **GeoModelHelper** (QObject \*const parent=nullptr)
- virtual bool **itemCoordinates** (const QModelIndex &index, GeoCoordinates \*const coordinates) const =0
- virtual PropertyFlags **itemFlags** (const QModelIndex &index) const

- virtual bool [itemIcon](#) (const QModelIndex &index, QPoint \*const offset, QSize \*const size, QPixmap \*const pixmap, QUrl \*const url) const  
*these are necessary for ungrouped models*
- virtual QAbstractItemModel \* [model](#) () const =0  
*these are necessary for grouped and ungrouped models*
- virtual PropertyFlags [modelFlags](#) () const
- virtual void [onIndicesClicked](#) (const QList< QPersistentModelIndex > &clickedIndices)
- virtual void [onIndicesMoved](#) (const QList< QPersistentModelIndex > &movedIndices, const [GeoCoordinates](#) &targetCoordinates, const QPersistentModelIndex &targetSnapIndex)
- virtual QPixmap [pixmapFromRepresentativeIndex](#) (const QPersistentModelIndex &index, const QSize &size)  
*these are used by MarkerModel for grouped models*
- virtual QItemSelectionModel \* [selectionModel](#) () const =0
- virtual void [snapItemsTo](#) (const QModelIndex &targetIndex, const QList< QModelIndex > &snappedIndices)
- void [snapItemsTo](#) (const QModelIndex &targetIndex, const QList< QPersistentModelIndex > &snappedIndices)

## 6.694.1 Detailed Description

[GeoModelHelper](#) is used to access data held in models, which is not suitable for transfer using the the Qt-style API, like coordinates or custom sized thumbnails.

The basic functions which have to be implemented are:

- [model\(\)](#): Returns a pointer to the model
- [selectionModel\(\)](#): Returns a pointer to the selection model. It may return a null-pointer if no selection model is used.
- [itemCoordinates\(\)](#): Returns the coordinates for a given item index, if it has any.
- [modelFlags\(\)](#): Returns flags for the model.

For ungrouped models, the following functions should also be implemented:

- [itemIcon\(\)](#): Returns an icon for an index, and an offset to the 'center' of the item.
- [itemFlags\(\)](#): Returns flags for individual items.
- [snapItemsTo\(\)](#): Grouped items have been moved and should snap to an index.

For grouped models which are accessed by [MarkerModel](#), the following functions should be implemented:

- [bestRepresentativeIndexFromList\(\)](#): Find the item that should represent a group of items.
- [pixmapFromRepresentativeIndex\(\)](#): Find a thumbnail for an item.

## 6.694.2 Member Function Documentation

### 6.694.2.1 [bestRepresentativeIndexFromList\(\)](#)

```
QPersistentModelIndex Digikam::GeoModelHelper::bestRepresentativeIndexFromList (
    const QList< QPersistentModelIndex > & list,
    const int sortKey ) [virtual]
```

Reimplemented in [Digikam::MapViewModelHelper](#).

### 6.694.2.2 itemCoordinates()

```
virtual bool Digikam::GeoModelHelper::itemCoordinates (
    const QModelIndex & index,
    GeoCoordinates *const coordinates ) const [pure virtual]
```

Implemented in [Digikam::MapViewModelHelper](#).

### 6.694.2.3 itemIcon()

```
bool Digikam::GeoModelHelper::itemIcon (
    const QModelIndex & index,
    QPoint *const offset,
    QSize *const size,
    QPixmap *const pixmap,
    QUrl *const url ) const [virtual]
```

Returns the icon for an ungrouped marker.

The icon can either be returned as a URL to an image, or as a QPixmap. If the caller can handle URLs (for example, to display them in HTML), he can provide the URL parameter. However, the [GeoModelHelper](#) may still choose to return a QPixmap instead, if no URL is available.

#### Parameters

<i>index</i>	Modelindex of the marker.
<i>offset</i>	Offset of the zero point in the icon, given from the top-left.
<i>size</i>	the size of the icon, only populated if a URL is returned.
<i>pixmap</i>	Holder for the pixmap of the icon.
<i>url</i>	URL of the icon if available.

Reimplemented in [Digikam::GPSBookmarkModelHelper](#).

### 6.694.2.4 model()

```
virtual QAbstractItemModel * Digikam::GeoModelHelper::model ( ) const [pure virtual]
```

Implemented in [Digikam::MapViewModelHelper](#), [Digikam::ItemGPSModelHelper](#), [Digikam::GPSBookmarkModelHelper](#), and [Digikam::GPSGeofaceModelHelper](#).

### 6.694.2.5 onIndicesClicked()

```
void Digikam::GeoModelHelper::onIndicesClicked (
    const QList< QPersistentModelIndex > & clickedIndices ) [virtual]
```

Reimplemented in [Digikam::MapViewModelHelper](#).

### 6.694.2.6 pixmapFromRepresentativeIndex()

```
QPixmap Digikam::GeoModelHelper::pixmapFromRepresentativeIndex (
    const QPersistentModelIndex & index,
    const QSize & size ) [virtual]
```

Reimplemented in [Digikam::MapViewModelHelper](#), [Digikam::ItemGPSModelHelper](#), and [Digikam::GPSGeofaceModelHelper](#).

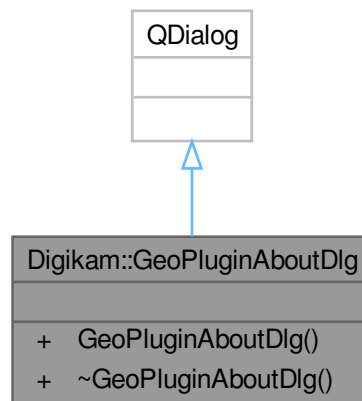
### 6.694.2.7 selectionModel()

```
virtual QItemSelectionModel * Digikam::GeoModelHelper::selectionModel ( ) const [pure virtual]
```

Implemented in [Digikam::MapViewModelHelper](#).

## 6.695 Digikam::GeoPluginAboutDlg Class Reference

Inheritance diagram for Digikam::GeoPluginAboutDlg:

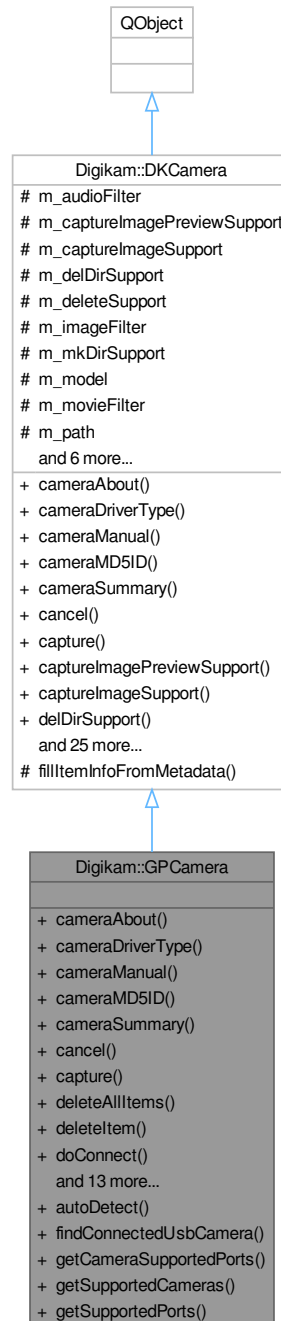


### Public Member Functions

- **GeoPluginAboutDlg** (PluginInterface \*const tool, QWidget \*const parent=nullptr)

## 6.696 Digikam::GPCamera Class Reference

Inheritance diagram for Digikam::GPCamera:



### Public Member Functions

- bool [cameraAbout](#) (QString &about) override
- DKCamera::CameraDriverType [cameraDriverType](#) () override



- bool [cameraManual](#) (QString &manual) override
- QByteArray [cameraMD5ID](#) () override
- bool [cameraSummary](#) (QString &summary) override
- void [cancel](#) () override
- bool [capture](#) (CamItemInfo &itemInfo) override
- bool [deleteAllItems](#) (const QString &folder)
  - recursively delete all items*
- bool [deleteItem](#) (const QString &folder, const QString &itemName) override
- bool [doConnect](#) () override
- bool [downloadItem](#) (const QString &folder, const QString &itemName, const QString &saveFile) override
- bool [getFolders](#) (const QString &folder) override
- bool [getFreeSpace](#) (qint64 &bytesSize, qint64 &bytesAvail) override
- void [getItemInfo](#) (const QString &folder, const QString &itemName, [CamItemInfo](#) &info, bool useMetadata) override
- bool [getItemsInfoList](#) (const QString &folder, bool useMetadata, [CamItemInfoList](#) &items) override
  - If getImageDimensions is false, the camera shall set width and height to -1 if the values are not immediately available.*
- bool [getItemsList](#) (const QString &folder, QStringList &itemsList)
- bool [getMetadata](#) (const QString &folder, const QString &itemName, [DMetadata](#) &meta) override
- bool [getPreview](#) (QImage &preview) override
- bool [getThumbnail](#) (const QString &folder, const QString &itemName, QImage &thumbnail) override
- [GPCamera](#) (const QString &title, const QString &model, const QString &port, const QString &path)
- bool [setLockItem](#) (const QString &folder, const QString &itemName, bool lock) override
- bool [uploadItem](#) (const QString &folder, const QString &itemName, const QString &localFile, [CamItemInfo](#) &itemInfo) override

## Public Member Functions inherited from [Digikam::DKCamera](#)

- bool [captureImagePreviewSupport](#) () const
- bool [captureImageSupport](#) () const
- bool [delDirSupport](#) () const
- bool [deleteSupport](#) () const
- [DKCamera](#) (const QString &title, const QString &model, const QString &port, const QString &path)
- QString [mimeType](#) (const QString &fileext) const
- bool [mkDirSupport](#) () const
- QString [model](#) () const
- QString [path](#) () const
- QString [port](#) () const
- void [printSupportedFeatures](#) ()
- bool [thumbnailSupport](#) () const
- QString [title](#) () const
- bool [uploadSupport](#) () const
- QString [uuid](#) () const

## Static Public Member Functions

- static int [autoDetect](#) (QString &model, QString &port)
- static bool [findConnectedUsbCamera](#) (int vendorId, int productId, QString &model, QString &port)
- static void [getCameraSupportedPorts](#) (const QString &model, QStringList &plist)
- static void [getSupportedCameras](#) (int &count, QStringList &cList)
- static void [getSupportedPorts](#) (QStringList &plist)

### Additional Inherited Members

### Public Types inherited from [Digikam::DKCamera](#)

- enum **CameraDriverType** { **GPhotoDriver** = 0 , **UMSDriver** }

### Signals inherited from [Digikam::DKCamera](#)

- void **signalFolderList** (const QStringList &)

### Protected Member Functions inherited from [Digikam::DKCamera](#)

- void **fillItemInfoFromMetadata** ([CamItemInfo](#) &item, const [DMetadata](#) &meta) const

### Protected Attributes inherited from [Digikam::DKCamera](#)

- QString **m\_audioFilter**
- bool **m\_captureImagePreviewSupport** = false
- bool **m\_captureImageSupport** = false
- bool **m\_delDirSupport** = false
- bool **m\_deleteSupport** = false
- QString **m\_imageFilter**
- bool **m\_mkDirSupport** = false
- QString **m\_model**
- QString **m\_movieFilter**
- QString **m\_path**
- QString **m\_port**
- QString **m\_rawFilter**
- bool **m\_thumbnailSupport** = false
- QString **m\_title**
- bool **m\_uploadSupport** = false
- QString **m\_uuid**

## 6.696.1 Detailed Description

Gphoto2 camera Implementation of abstract type [DKCamera](#)

## 6.696.2 Member Function Documentation

### 6.696.2.1 cameraAbout()

```
bool Digikam::GPCamera::cameraAbout (
    QString & about ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.2 cameraDriverType()

```
DKCamera::CameraDriverType Digikam::GPCamera::cameraDriverType ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.3 cameraManual()

```
bool Digikam::GPCamera::cameraManual (
    QString & manual ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.4 cameraMD5ID()

```
QByteArray Digikam::GPCamera::cameraMD5ID ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.5 cameraSummary()

```
bool Digikam::GPCamera::cameraSummary (
    QString & summary ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.6 cancel()

```
void Digikam::GPCamera::cancel ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.7 capture()

```
bool Digikam::GPCamera::capture (
    CamItemInfo & itemInfo ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.8 deleteItem()

```
bool Digikam::GPCamera::deleteItem (
    const QString & folder,
    const QString & itemName ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.9 doConnect()

```
bool Digikam::GPCamera::doConnect ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.10 downloadItem()

```
bool Digikam::GPCamera::downloadItem (
    const QString & folder,
    const QString & itemName,
    const QString & saveFile ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.11 getFolders()

```
bool Digikam::GPCamera::getFolders (
    const QString & folder ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.12 getFreeSpace()

```
bool Digikam::GPCamera::getFreeSpace (
    quint64 & bytesSize,
    quint64 & bytesAvail ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.13 getItemInfo()

```
void Digikam::GPCamera::getItemInfo (
    const QString & folder,
    const QString & itemName,
    CamItemInfo & info,
    bool useMetadata ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.14 getItemsInfoList()

```
bool Digikam::GPCamera::getItemsInfoList (
    const QString & folder,
    bool useMetadata,
    CamItemInfoList & infoList ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.15 getMetadata()

```
bool Digikam::GPCamera::getMetadata (
    const QString & folder,
    const QString & itemName,
    DMetadata & meta ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.16 getPreview()

```
bool Digikam::GPCamera::getPreview (
    QImage & preview ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.17 getThumbnail()

```
bool Digikam::GPCamera::getThumbnail (
    const QString & folder,
    const QString & itemName,
    QImage & thumbnail ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.696.2.18 setLockItem()

```
bool Digikam::GPCamera::setLockItem (
    const QString & folder,
    const QString & itemName,
    bool lock ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

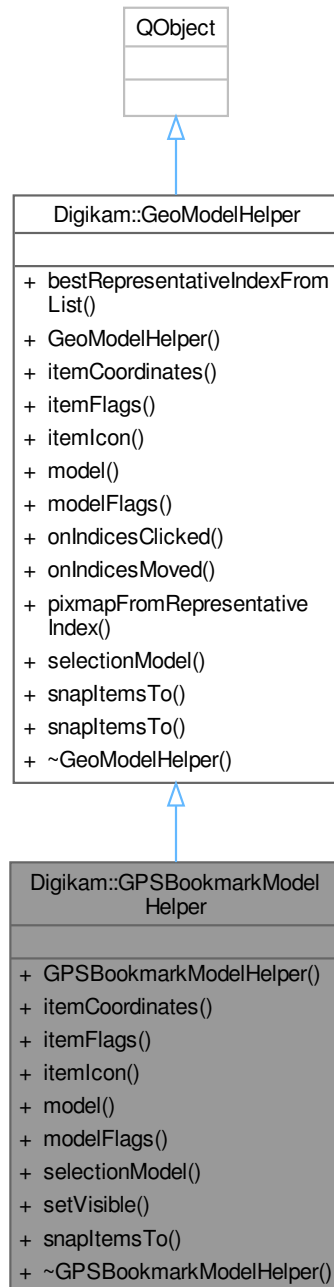
### 6.696.2.19 uploadItem()

```
bool Digikam::GPCamera::uploadItem (
    const QString & folder,
    const QString & itemName,
    const QString & localFile,
    CamItemInfo & itemInfo ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

## 6.697 Digikam::GPSBookmarkModelHelper Class Reference

Inheritance diagram for Digikam::GPSBookmarkModelHelper:



### Public Types

- enum **Constants** { **CoordinatesRole** = Qt::UserRole + 1 }

## Public Types inherited from Digikam::GeoModelHelper

- enum **PropertyFlag** { **FlagNull** = 0 , **FlagVisible** = 1 , **FlagMovable** = 2 , **FlagSnaps** = 4 }

## Signals

- void **signalUndoCommand** ([GPSUndoCommand](#) \*undoCommand)

## Signals inherited from Digikam::GeoModelHelper

- void **signalModelChangedDrastically** ()
- void **signalThumbnailAvailableForIndex** (const [QPersistentModelIndex](#) &index, const [QPixmap](#) &pixmap)
- void **signalVisibilityChanged** ()

## Public Member Functions

- **GPSBookmarkModelHelper** ([BookmarksManager](#) \*const bookmarkManager, [GPSItemModel](#) \*const imageModel, [QObject](#) \*const parent=nullptr)
- bool **itemCoordinates** (const [QModelIndex](#) &index, [GeoCoordinates](#) \*const coordinates) const override
- PropertyFlags **itemFlags** (const [QModelIndex](#) &index) const override
- bool **itemIcon** (const [QModelIndex](#) &index, [QPoint](#) \*const offset, [QSize](#) \*const size, [QPixmap](#) \*const pixmap, [QUrl](#) \*const url) const override
- these are necessary for ungrouped models*
- [QAbstractItemModel](#) \* **model** () const override
- these are necessary for grouped and ungrouped models*
- PropertyFlags **modelFlags** () const override
- [QItemSelectionModel](#) \* **selectionModel** () const override
- void **setVisible** (const bool state)
- void **snapItemsTo** (const [QModelIndex](#) &targetIndex, const [QList](#)< [QModelIndex](#) > &snappedIndices) override

## Public Member Functions inherited from Digikam::GeoModelHelper

- virtual [QPersistentModelIndex](#) **bestRepresentativeIndexFromList** (const [QList](#)< [QPersistentModelIndex](#) > &list, const int sortKey)
- **GeoModelHelper** ([QObject](#) \*const parent=nullptr)
- virtual void **onIndicesClicked** (const [QList](#)< [QPersistentModelIndex](#) > &clickedIndices)
- virtual void **onIndicesMoved** (const [QList](#)< [QPersistentModelIndex](#) > &movedIndices, const [GeoCoordinates](#) &targetCoordinates, const [QPersistentModelIndex](#) &targetSnapIndex)
- virtual [QPixmap](#) **pixmapFromRepresentativeIndex** (const [QPersistentModelIndex](#) &index, const [QSize](#) &size)
- these are used by MarkerModel for grouped models*
- void **snapItemsTo** (const [QModelIndex](#) &targetIndex, const [QList](#)< [QPersistentModelIndex](#) > &snappedIndices)

## 6.697.1 Member Function Documentation

### 6.697.1.1 itemCoordinates()

```
bool Digikam::GPSBookmarkModelHelper::itemCoordinates (
    const QModelIndex & index,
    GeoCoordinates *const coordinates ) const [override], [virtual]
```

Implements [Digikam::GeoModelHelper](#).

### 6.697.1.2 itemFlags()

```
GeoModelHelper::PropertyFlags Digikam::GPSBookmarkModelHelper::itemFlags (
    const QModelIndex & index ) const [override], [virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.697.1.3 itemIcon()

```
bool Digikam::GPSBookmarkModelHelper::itemIcon (
    const QModelIndex & index,
    QPoint *const offset,
    QSize *const size,
    QPixmap *const pixmap,
    QUrl *const url ) const [override], [virtual]
```

Returns the icon for an ungrouped marker.

The icon can either be returned as a URL to an image, or as a QPixmap. If the caller can handle URLs (for example, to display them in HTML), he can provide the URL parameter. However, the [GeoModelHelper](#) may still choose to return a QPixmap instead, if no URL is available.

#### Parameters

<i>index</i>	Modelindex of the marker.
<i>offset</i>	Offset of the zero point in the icon, given from the top-left.
<i>size</i>	the size of the icon, only populated if a URL is returned.
<i>pixmap</i>	Holder for the pixmap of the icon.
<i>url</i>	URL of the icon if available.

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.697.1.4 model()

```
QAbstractItemModel * Digikam::GPSBookmarkModelHelper::model ( ) const [override], [virtual]
```

Implements [Digikam::GeoModelHelper](#).

### 6.697.1.5 modelFlags()

```
GeoModelHelper::PropertyFlags Digikam::GPSBookmarkModelHelper::modelFlags ( ) const [override],
[virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.697.1.6 selectionModel()

```
QItemSelectionModel * Digikam::GPSBookmarkModelHelper::selectionModel ( ) const [override],
[virtual]
```

Implements [Digikam::GeoModelHelper](#).



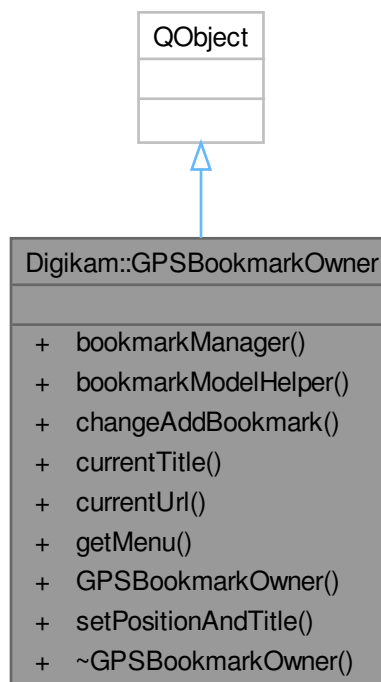
### 6.697.1.7 snapItemsTo()

```
void Digikam::GPSBookmarkModelHelper::snapItemsTo (
    const QModelIndex & targetIndex,
    const QList< QModelIndex > & snappedIndices ) [override], [virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

## 6.698 Digikam::GPSBookmarkOwner Class Reference

Inheritance diagram for Digikam::GPSBookmarkOwner:



### Signals

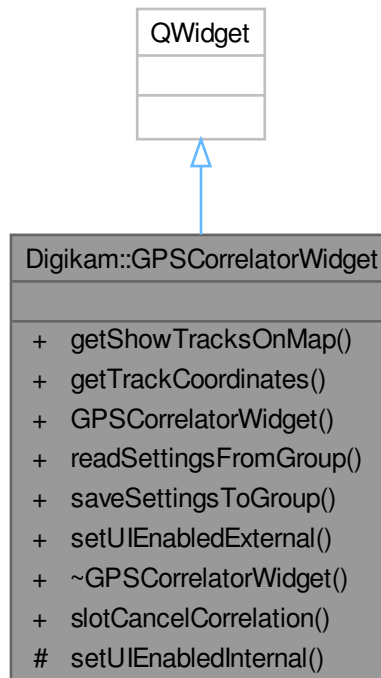
- void **positionSelected** (const [GPSDataContainer](#) &position)

### Public Member Functions

- [BookmarksManager](#) \* **bookmarkManager** () const
- [GPSBookmarkModelHelper](#) \* **bookmarkModelHelper** () const
- void **changeAddBookmark** (const bool state)
- QString **currentTitle** () const
- QString **currentUrl** () const
- QMenu \* **getMenu** () const
- [GPSBookmarkOwner](#) ([GPSItemModel](#) \*const gpsItemModel, QWidget \*const parent)
- void **setPositionAndTitle** (const [GeoCoordinates](#) &coordinates, const QString &title)

## 6.699 Digikam::GPSCorrelatorWidget Class Reference

Inheritance diagram for Digikam::GPSCorrelatorWidget:



### Public Slots

- void **slotCancelCorrelation** ()

### Signals

- void **signalAllTrackFilesReady** ()
- void **signalProgressChanged** (const int currentProgress)
- void **signalProgressSetup** (const int maxProgress, const QString &progressText)
- void **signalSetUIEnabled** (const bool enabledState)
- void **signalSetUIEnabled** (const bool enabledState, QObject \*const cancelObject, const QString &cancelText, const QString &cancelText) (Slot)
- void **signalTrackListChanged** (const [Digikam::GeoCoordinates](#) &coordinate)
- void **signalUndoCommand** ([GPSUndoCommand](#) \*undoCommand)

### Public Member Functions

- bool **getShowTracksOnMap** () const
- [QList](#)< [GeoCoordinates::List](#) > **getTrackCoordinates** () const
- **GPSCorrelatorWidget** ([QWidget](#) \*const parent, [GPSItemModel](#) \*const imageModel, [QItemSelectionModel](#) \*const selectionModel, [TrackManager](#) \*const trackManager)
- void **readSettingsFromGroup** (const [KConfigGroup](#) \*const group)
- void **saveSettingsToGroup** ([KConfigGroup](#) \*const group)
- void **setUIEnabledExternal** (const bool state)

### Protected Member Functions

- void **setUIEnabledInternal** (const bool state)

## 6.700 Digikam::GPSDataContainer Class Reference

### Public Types

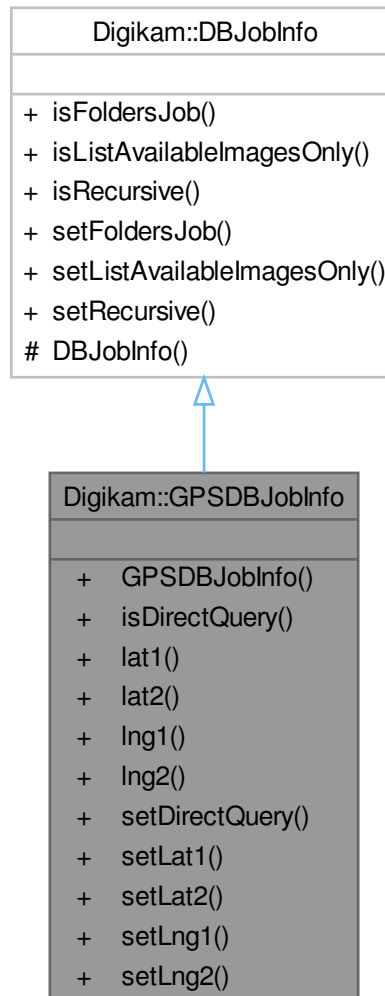
- enum **HasFlagsEnum** {  
    **HasCoordinates** = 1 , **HasAltitude** = 2 , **HasIsInterpolated** = 4 , **HasNSatellites** = 8 ,  
    **HasDop** = 16 , **HasFixType** = 32 , **HasSpeed** = 64 }

### Public Member Functions

- void **clear** ()
- void **clearAltitude** ()
- void **clearDop** ()
- void **clearFixType** ()
- void **clearNonCoordinates** ()
- void **clearNSatellites** ()
- void **clearSpeed** ()
- HasFlags **flags** () const
- [GeoCoordinates](#) **getCoordinates** () const
- qreal **getDop** () const
- qreal **getFixType** () const
- int **getNSatellites** () const
- qreal **getSpeed** () const  
    *Return the speed in m/s.*
- bool **hasAltitude** () const
- bool **hasCoordinates** () const
- bool **hasDop** () const
- bool **hasFixType** () const
- bool **hasNSatellites** () const
- bool **hasSpeed** () const
- bool **operator==** (const [GPSDataContainer](#) &b) const
- void **setAltitude** (const qreal alt)
- void **setCoordinates** (const [GeoCoordinates](#) &coordinates)
- void **setDop** (const qreal dop)
- void **setFixType** (const int fixType)
- void **setLatLon** (const qreal lat, const qreal lon)
- void **setNSatellites** (const int nSatellites)
- void **setSpeed** (const qreal speed)  
    *Set the speed in m/s.*

## 6.701 Digikam::GPSDBJobInfo Class Reference

Inheritance diagram for Digikam::GPSDBJobInfo:



### Public Member Functions

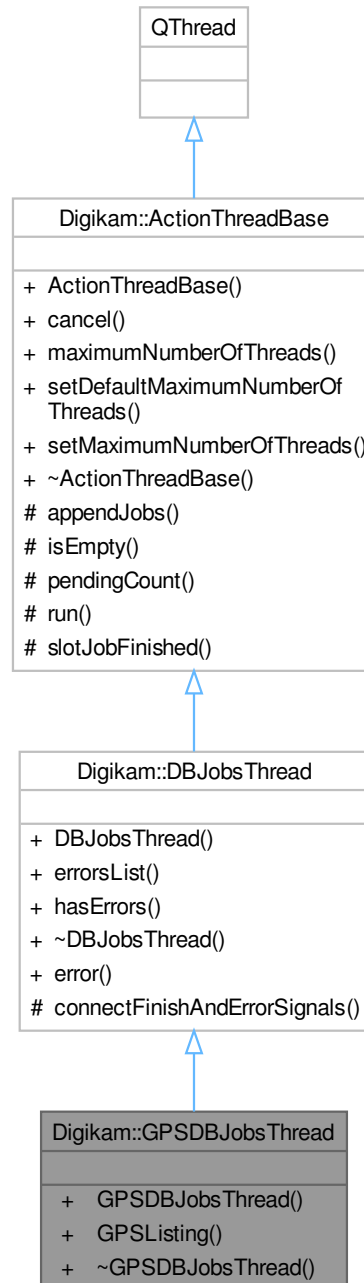
- bool **isDirectQuery** () const
- qreal **lat1** () const
- qreal **lat2** () const
- qreal **lng1** () const
- qreal **lng2** () const
- void **setDirectQuery** ()
- void **setLat1** (qreal lat)
- void **setLat2** (qreal lat)
- void **setLng1** (qreal lng)
- void **setLng2** (qreal lng)

**Public Member Functions inherited from [Digikam::DBJobInfo](#)**

- bool **isFoldersJob** () const
- bool **isListAvailableImagesOnly** () const
- bool **isRecursive** () const
- void **setFoldersJob** ()
- void **setListAvailableImagesOnly** ()
- void **setRecursive** ()

## 6.702 Digikam::GPSDBJobsThread Class Reference

Inheritance diagram for Digikam::GPSDBJobsThread:



### Signals

- void **directQueryData** (const QList< QVariant > &data)

## Signals inherited from [Digikam::DBJobsThread](#)

- void **data** (const QList< [ItemListerRecord](#) > &records)
- void **finished** ()

## Public Member Functions

- **GPSDBJobsThread** (QObject \*const parent)
- void **GPSListing** (const [GPSDBJobInfo](#) &info)  
*Starts GPS listing and scanning.*

## Public Member Functions inherited from [Digikam::DBJobsThread](#)

- **DBJobsThread** (QObject \*const parent)
- QList< QString > & **errorsList** ()  
*A method to get all errors reported from jobs.*
- bool **hasErrors** ()  
*hasErrors: a method to check for jobs errors*

## Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

## Additional Inherited Members

## Public Slots inherited from [Digikam::DBJobsThread](#)

- void **error** (const QString &errString)  
*Appends the error string to m\_errorsList.*

## Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void **slotJobFinished** ()

## Protected Member Functions inherited from [Digikam::DBJobsThread](#)

- void **connectFinishAndErrorSignals** ([DBJob](#) \*const j)  
*Connects the signals of job to the signals of the thread.*

## Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void **appendJobs** (const [ActionJobCollection](#) &jobs)
- bool **isEmpty** () const
- int **pendingCount** () const
- void **run** () override

## 6.702.1 Member Function Documentation

### 6.702.1.1 GPSListing()

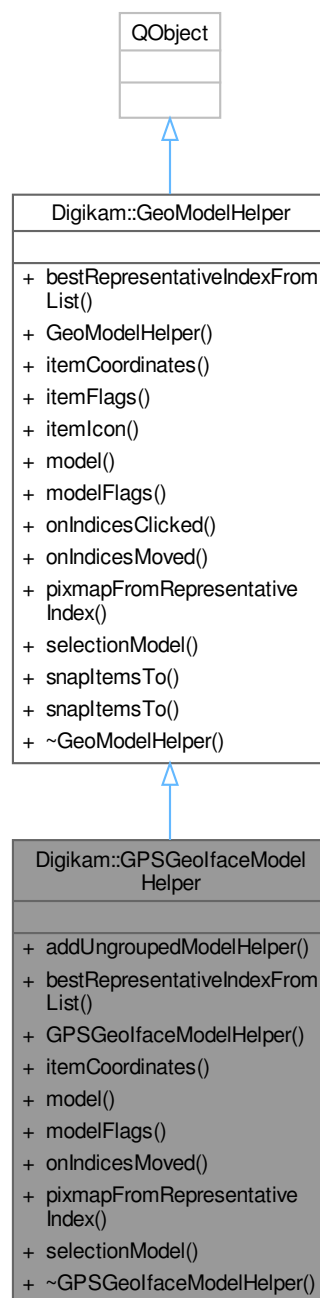
```
void Digikam::GPSDBJobsThread::GPSListing (
    const GPSDBJobInfo & info )
```

## Parameters

<i>info</i>	represents the GPS job info
-------------	-----------------------------

## 6.703 Digikam::GPSGeofaceModelHelper Class Reference

Inheritance diagram for Digikam::GPSGeofaceModelHelper:





## Signals

- void **signalUndoCommand** ([GPSUndoCommand](#) \*undoCommand)

## Signals inherited from [Digikam::GeoModelHelper](#)

- void **signalModelChangedDrastically** ()
- void **signalThumbnailAvailableForIndex** (const [QPersistentModelIndex](#) &index, const [QPixmap](#) &pixmap)
- void **signalVisibilityChanged** ()

## Public Member Functions

- void **addUngroupedModelHelper** ([GeoModelHelper](#) \*const newModelHelper)
- [QPersistentModelIndex](#) **bestRepresentativeIndexFromList** (const [QList](#)< [QPersistentModelIndex](#) > &list, const int sortKey) override
- **GPSGeofaceModelHelper** ([GPSItemModel](#) \*const model, [QItemSelectionModel](#) \*const selectionModel, [QObject](#) \*const parent=nullptr)
- bool **itemCoordinates** (const [QModelIndex](#) &index, [GeoCoordinates](#) \*const coordinates) const override
- [QAbstractItemModel](#) \* **model** () const override
  - these are necessary for grouped and ungrouped models*
- PropertyFlags **modelFlags** () const override
- void **onIndicesMoved** (const [QList](#)< [QPersistentModelIndex](#) > &movedMarkers, const [GeoCoordinates](#) &targetCoordinates, const [QPersistentModelIndex](#) &targetSnapIndex) override
- [QPixmap](#) **pixmapFromRepresentativeIndex** (const [QPersistentModelIndex](#) &index, const [QSize](#) &size) override
  - these are used by MarkerModel for grouped models*
- [QItemSelectionModel](#) \* **selectionModel** () const override

## Public Member Functions inherited from [Digikam::GeoModelHelper](#)

- **GeoModelHelper** ([QObject](#) \*const parent=nullptr)
- virtual PropertyFlags **itemFlags** (const [QModelIndex](#) &index) const
- virtual bool **itemIcon** (const [QModelIndex](#) &index, [QPoint](#) \*const offset, [QSize](#) \*const size, [QPixmap](#) \*const pixmap, [QUrl](#) \*const url) const
  - these are necessary for ungrouped models*
- virtual void **onIndicesClicked** (const [QList](#)< [QPersistentModelIndex](#) > &clickedIndices)
- virtual void **snapItemsTo** (const [QModelIndex](#) &targetIndex, const [QList](#)< [QModelIndex](#) > &snappedIndices)
- void **snapItemsTo** (const [QModelIndex](#) &targetIndex, const [QList](#)< [QPersistentModelIndex](#) > &snappedIndices)

## Additional Inherited Members

## Public Types inherited from [Digikam::GeoModelHelper](#)

- enum **PropertyFlag** { **FlagNull** = 0 , **FlagVisible** = 1 , **FlagMovable** = 2 , **FlagSnaps** = 4 }

## 6.703.1 Member Function Documentation

### 6.703.1.1 bestRepresentativeIndexFromList()

```
QPersistentModelIndex Digikam::GPSGeoIfaceModelHelper::bestRepresentativeIndexFromList (
    const QList< QPersistentModelIndex > & list,
    const int sortKey ) [override], [virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.703.1.2 itemCoordinates()

```
bool Digikam::GPSGeoIfaceModelHelper::itemCoordinates (
    const QModelIndex & index,
    GeoCoordinates *const coordinates ) const [override], [virtual]
```

Implements [Digikam::GeoModelHelper](#).

### 6.703.1.3 model()

```
QAbstractItemModel * Digikam::GPSGeoIfaceModelHelper::model ( ) const [override], [virtual]
```

Implements [Digikam::GeoModelHelper](#).

### 6.703.1.4 modelFlags()

```
GeoModelHelper::PropertyFlags Digikam::GPSGeoIfaceModelHelper::modelFlags ( ) const [override],
[virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.703.1.5 onIndicesMoved()

```
void Digikam::GPSGeoIfaceModelHelper::onIndicesMoved (
    const QList< QPersistentModelIndex > & movedMarkers,
    const GeoCoordinates & targetCoordinates,
    const QPersistentModelIndex & targetSnapIndex ) [override], [virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.703.1.6 pixmapFromRepresentativeIndex()

```
QPixmap Digikam::GPSGeoIfaceModelHelper::pixmapFromRepresentativeIndex (
    const QPersistentModelIndex & index,
    const QSize & size ) [override], [virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

## 6.703.1.7 selectionModel()

```
QItemSelectionModel * Digikam::GPSGeoIfaceModelHelper::selectionModel ( ) const [override],
[virtual]
```

Implements [Digikam::GeoModelHelper](#).

## 6.704 Digikam::GPSItemContainer Class Reference

Inheritance diagram for Digikam::GPSItemContainer:



## Public Member Functions

- **GPSItemContainer** (const [QUrl](#) &url)

### Loading and saving

- virtual [QString](#) **saveChanges** ()
- virtual bool **loadImageData** ()
- bool **isDirty** () const
- [QUrl](#) **url** () const
- [QDateTime](#) **dateTime** () const

### GPS related functions

- void **setCoordinates** (const [GeoCoordinates](#) &newCoordinates)
- [GeoCoordinates](#) **coordinates** () const
- [GPSDataContainer](#) **gpsData** () const
- void **setGPSData** (const [GPSDataContainer](#) &container)
- void **restoreGPSData** (const [GPSDataContainer](#) &container)

*Restore the gps data to container. Sets m\_dirty to false if container equals savedState.*

## Static Public Attributes

- static const int **ColumnAccuracy** = 6
- static const int **ColumnAltitude** = 5
- static const int **ColumnDateTime** = 2
- static const int **ColumnDOP** = 9
- static const int **ColumnFilename** = 1
- static const int **ColumnFixType** = 10
- static const int **ColumnGPSItemContainerCount** = 13
- static const int **ColumnLatitude** = 3
- static const int **ColumnLongitude** = 4
- static const int **ColumnNSatellites** = 11
- static const int **ColumnSpeed** = 12
- static const int **ColumnStatus** = 8
- static const int **ColumnTags** = 7
- static const int **ColumnThumbnail** = 0
- static const int **RoleCoordinates** = [Qt::UserRole](#) + 1

## Tag related functions

- [GPSItemModel](#) \* **m\_model** = nullptr
- [QUrl](#) **m\_url**
- [QDateTime](#) **m\_dateTime**
- bool **m\_dirty** = false
- [GPSDataContainer](#) **m\_gpsData**
- [GPSDataContainer](#) **m\_savedState**
- bool **m\_tagListDirty** = false
- [QList](#)< [QList](#)< [TagData](#) > > **m\_tagList**
- [QList](#)< [QList](#)< [TagData](#) > > **m\_savedTagList**
- bool **m\_writeXmpTags** = true
- bool **m\_writeMetaLoc** = true
- class **GPSItemModel**
- void **setTagList** (const [QList](#)< [QList](#)< [TagData](#) > > &externalTagList)

- bool `isTagListDirty` () const
- QList< QList< TagData > > `getTagList` () const
- void `restoreRGTagList` (const QList< QList< TagData > > &tagList)
- void `writeTagsToXmp` (const bool writeXmpTags)
- void `writeLocations` (const bool writeMetaLoc)
- void `setLocationInfo` (const TagData &tagData, lptcCoreLocationInfo &locationInfo)
- QVariant `data` (const int column, const int role) const  
*these are only to be called by the GPSItemModel*
- void `setModel` (GPSItemModel \*const model)
- void `emitDataChanged` ()
- DMetadata \* `getMetadataForFile` () const
- SaveProperties `saveProperties` () const

### Functions used by the model

- bool `lessThan` (const GPSItemContainer \*const otherItem, const int column) const
- static void `setHeaderData` (GPSItemModel \*const model)

## 6.704.1 Member Function Documentation

### 6.704.1.1 getTagList()

```
QList< QList< TagData > > Digikam::GPSItemContainer::getTagList ( ) const
```

Returns the tag list of the current image.

### 6.704.1.2 isTagListDirty()

```
bool Digikam::GPSItemContainer::isTagListDirty ( ) const
```

Returns

Returns true if the current image has been modified and not saved.

### 6.704.1.3 loadImageData()

```
bool Digikam::GPSItemContainer::loadImageData ( ) [virtual]
```

### 6.704.1.4 restoreRGTagList()

```
void Digikam::GPSItemContainer::restoreRGTagList (
    const QList< QList< TagData > > & tagList )
```

Replaces the current tag list with the one contained in tagList.

### 6.704.1.5 saveChanges()

```
QString Digikam::GPSItemContainer::saveChanges ( ) [virtual]
```

### 6.704.1.6 setTagList()

```
void Digikam::GPSItemContainer::setTagList (
    const QList< QList< TagData > > & externalTagList )
```

The tags added in reverse geocoding process are stored in each image, before they end up in external tag model. This function adds them.

## Parameters

<i>externalTagList</i>	A list containing tags.
------------------------	-------------------------

**6.704.1.7 writeLocations()**

```
void Digikam::GPSItemContainer::writeLocations (
    const bool writeMetaLoc ) [inline]
```

Writes the current tags to the metadata location fields.

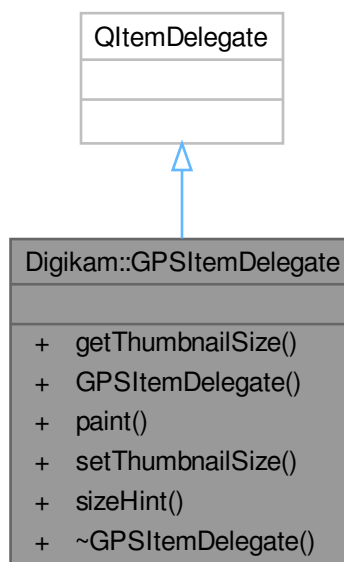
**6.704.1.8 writeTagsToXmp()**

```
void Digikam::GPSItemContainer::writeTagsToXmp (
    const bool writeXmpTags ) [inline]
```

Writes the current tags to XMP metadata.

**6.705 Digikam::GPSItemDelegate Class Reference**

Inheritance diagram for Digikam::GPSItemDelegate:



### Public Member Functions

- int **getThumbnailSize** () const
- **GPSItemDelegate** ([GPSItemList](#) \*const imageList, QObject \*const parent=nullptr)
- void **paint** (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &sortMappedindex) const override
- void **setThumbnailSize** (const int size)
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &sortMappedindex) const override

## 6.706 Digikam::GPSItemInfo Class Reference

### Public Types

- typedef QList< [GPSItemInfo](#) > **List**

### Static Public Member Functions

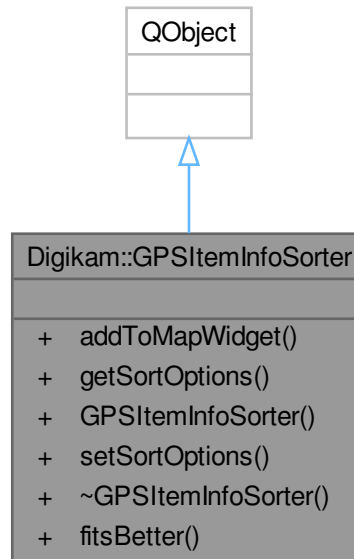
- static [GPSItemInfo](#) **fromIdCoordinatesRatingDateTime** (const qlonglong p\_id, const [GeoCoordinates](#) &p\_coordinates, const int p\_rating, const QDateTime &p\_creationDate)

### Public Attributes

- [GeoCoordinates](#) **coordinates**
- QDateTime **dateTime**
- qlonglong **id** = -2
- int **rating** = -1
- QUrl **url**

## 6.707 Digikam::GPSItemInfoSorter Class Reference

Inheritance diagram for Digikam::GPSItemInfoSorter:



### Public Types

- enum **SortOption** { **SortYoungestFirst** = 0 , **SortOldestFirst** = 1 , **SortRating** = 2 }

### Public Member Functions

- void **addToMapWidget** ([MapWidget](#) \*const mapWidget)
- [SortOptions](#) **getSortOptions** () const
- **GPSItemInfoSorter** ([QObject](#) \*const parent)
- void **setSortOptions** (const [SortOptions](#) sortOptions)

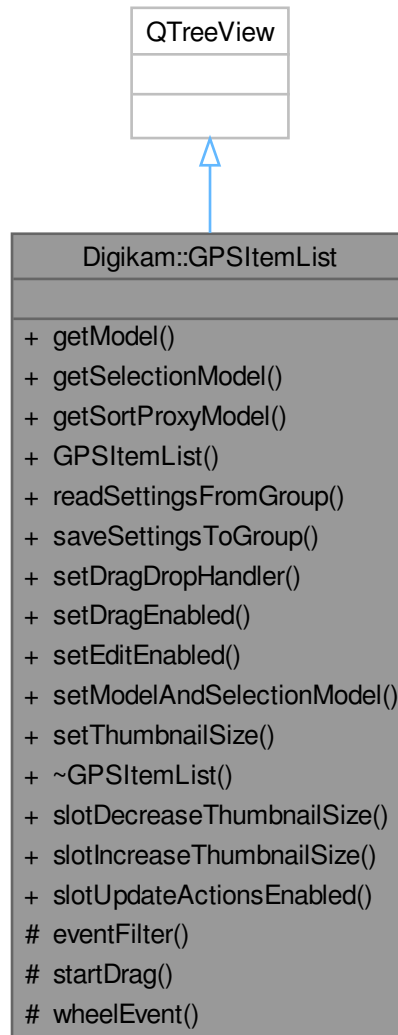
### Static Public Member Functions

- static bool **fitsBetter** (const [GPSItemInfo](#) &oldInfo, const [GeoGroupState](#) oldState, const [GPSItemInfo](#) &newInfo, const [GeoGroupState](#) newState, const [GeoGroupState](#) globalGroupState, const [SortOptions](#) sortOptions)



## 6.708 Digikam::GPSItemList Class Reference

Inheritance diagram for Digikam::GPSItemList:



### Public Slots

- void **slotDecreaseThumbnailSize** ()
- void **slotIncreaseThumbnailSize** ()
- void **slotUpdateActionsEnabled** ()

### Signals

- void **signalImageActivated** (const QModelIndex &index)

**Public Member Functions**

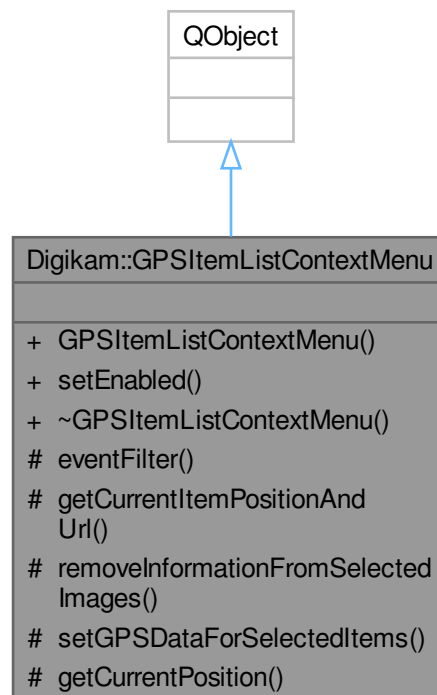
- [GPSItemModel](#) \* **getModel** () const
- QItemSelectionModel \* **getSelectionModel** () const
- [GPSItemSortProxyModel](#) \* **getSortProxyModel** () const
- **GPSItemList** (QWidget \*const parent=nullptr)
- void **readSettingsFromGroup** (const KConfigGroup \*const group)
- void **saveSettingsToGroup** (KConfigGroup \*const group)
- void **setDragDropHandler** ([ItemListDragDropHandler](#) \*const dragDropHandler)
- void **setDragEnabled** (const bool state)
- void **setEditEnabled** (const bool state)
- void **setModelAndSelectionModel** ([GPSItemModel](#) \*const model, QItemSelectionModel \*const selectionModel)
- void **setThumbnailSize** (const int size)

**Protected Member Functions**

- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- void **startDrag** (Qt::DropActions supportedActions) override
- void **wheelEvent** (QWheelEvent \*we) override

**6.709 Digikam::GPSItemListContextMenu Class Reference**

Inheritance diagram for Digikam::GPSItemListContextMenu:



## Signals

- void **signalProgressChanged** (const int currentProgress)
- void **signalProgressSetup** (const int maxProgress, const QString &progressText)
- void **signalSetUIEnabled** (const bool enabledState)
- void **signalSetUIEnabled** (const bool enabledState, QObject \*const cancelObject, const QString &cancelSlot)
- void **signalUndoCommand** ([GPSUndoCommand](#) \*undoCommand)

## Public Member Functions

- **GPSItemListContextMenu** ([GPSItemList](#) \*const imagesList, [GPSBookmarkOwner](#) \*const bookmarkOwner=nullptr)
- void **setEnabled** (const bool state)

## Protected Member Functions

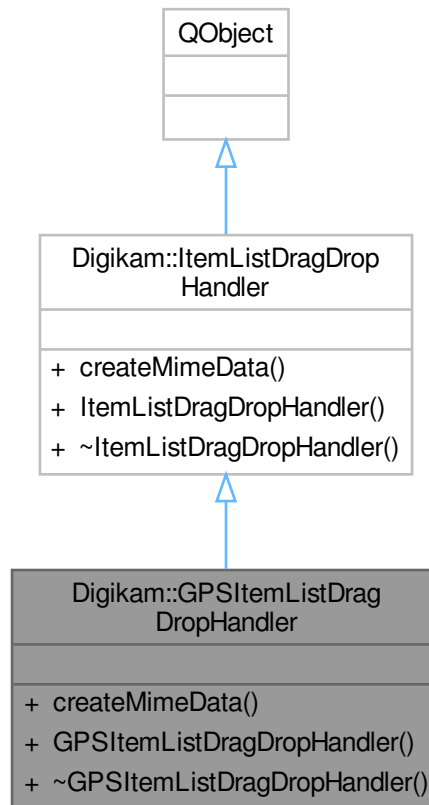
- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- bool **getCurrentItemPositionAndUrl** ([GPSDataContainer](#) \*const gpsInfo, QUrl \*const itemUrl)
- void **removeInformationFromSelectedImages** (const [GPSDataContainer::HasFlags](#) flagsToClear, const QString &undoDescription)
- void **setGPSDataForSelectedItems** (const [GPSDataContainer](#) &gpsData, const QString &undoDescription)

## Static Protected Member Functions

- static bool **getCurrentPosition** ([GPSDataContainer](#) \*position, void \*mydata)

## 6.710 Digikam::GPSItemListDragDropHandler Class Reference

Inheritance diagram for Digikam::GPSItemListDragDropHandler:



### Public Member Functions

- `QMimeData * createMimeData (const QList< QPersistentModelIndex > &modelIndices)` override
- `GPSItemListDragDropHandler (QObject *const parent=nullptr)`

### Public Member Functions inherited from [Digikam::ItemListDragDropHandler](#)

- `ItemListDragDropHandler (QObject *const parent=nullptr)`

### 6.710.1 Member Function Documentation

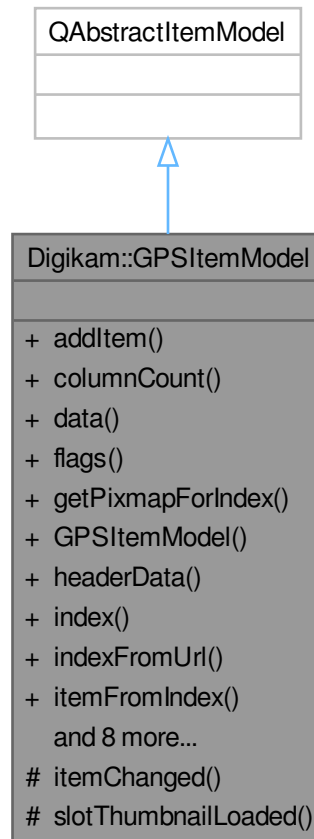
#### 6.710.1.1 createMimeData()

```
QMimeData * Digikam::GPSItemListDragDropHandler::createMimeData (
    const QList< QPersistentModelIndex > & modelIndices ) [override], [virtual]
```

Implements [Digikam::ItemListDragDropHandler](#).

## 6.711 Digikam::GPSItemModel Class Reference

Inheritance diagram for Digikam::GPSItemModel:



### Signals

- void **signalThumbnailForIndexAvailable** (const `QPersistentModelIndex` &index, const `QPixmap` &pixmap)

### Public Member Functions

- void **addItem** (`GPSItemContainer` \*const newItem)
- int **columnCount** (const `QModelIndex` &parent=`QModelIndex()`) const override
- `QVariant` **data** (const `QModelIndex` &index, int role=`Qt::DisplayRole`) const override
- `Qt::ItemFlags` **flags** (const `QModelIndex` &index) const override
- `QPixmap` **getPixmapForIndex** (const `QPersistentModelIndex` &itemIndex, const int size)
- `GPSItemModel` (`QObject` \*const parent=`nullptr`)
- `QVariant` **headerData** (int section, `Qt::Orientation` orientation, int role) const override
- `QModelIndex` **index** (int row, int column, const `QModelIndex` &parent=`QModelIndex()`) const override
- `QModelIndex` **indexFromUrl** (const `QUrl` &url) const
- `GPSItemContainer` \* **itemFromIndex** (const `QModelIndex` &index) const

- [GPSItemContainer](#) \* **itemFromUrl** (const QUrl &url) const
- QModelIndex **parent** (const QModelIndex &index) const override
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setColumnCount** (const int nColumns)
- bool **setData** (const QModelIndex &index, const QVariant &value, int role) override
- bool **setHeaderData** (int section, Qt::Orientation orientation, const QVariant &value, int role) override
- Qt::DropActions **supportedDragActions** () const override

### Protected Slots

- void **slotThumbnailLoaded** (const [LoadingDescription](#) &, const QPixmap &)

### Protected Member Functions

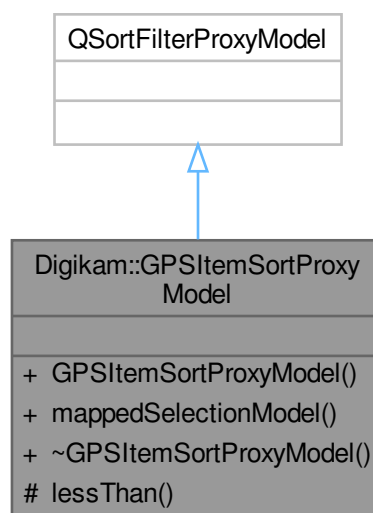
- void **itemChanged** ([GPSItemContainer](#) \*const changedItem)

### Friends

- class [GPSItemContainer](#)

## 6.712 Digikam::GPSItemSortProxyModel Class Reference

Inheritance diagram for Digikam::GPSItemSortProxyModel:



**Public Member Functions**

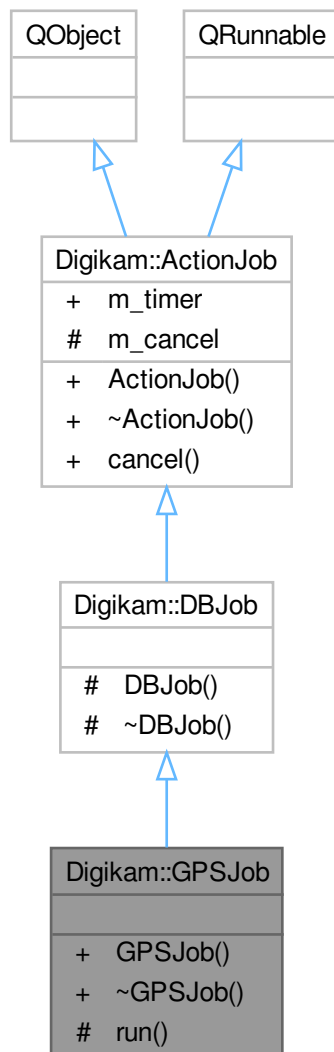
- **GPSItemSortProxyModel** ([GPSItemModel](#) \*const imageModel, [QItemSelectionModel](#) \*const source↔  
SelectionModel)
- [QItemSelectionModel](#) \* **mappedSelectionModel** () const

**Protected Member Functions**

- bool **lessThan** (const [QModelIndex](#) &left, const [QModelIndex](#) &right) const override

**6.713 Digikam::GPSJob Class Reference**

Inheritance diagram for Digikam::GPSJob:



## Signals

- void **directQueryData** (const QList< QVariant > &data)

## Signals inherited from [Digikam::DBJob](#)

- void **data** (const QList< [ItemLISTERRecord](#) > &records)
- void **error** (const QString &err)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Member Functions

- **GPSJob** (const [GPSDBJobInfo](#) &jobInfo)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void **run** () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

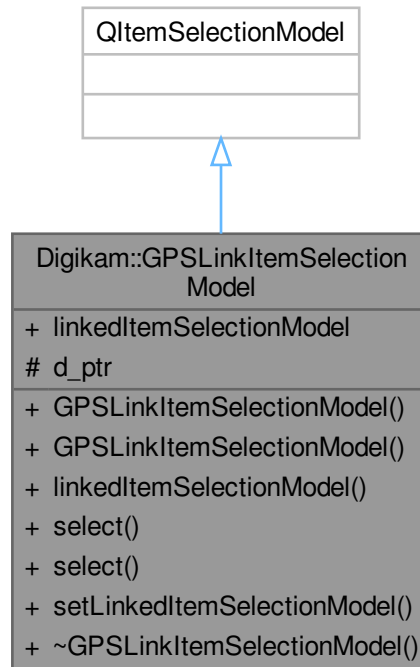
## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false



## 6.714 Digikam::GPSLinkItemSelectionModel Class Reference

Inheritance diagram for Digikam::GPSLinkItemSelectionModel:



### Signals

- void **linkedItemSelectionModelChanged** ()

### Public Member Functions

- **GPSLinkItemSelectionModel** (QAbstractItemModel \*const targetModel, QItemSelectionModel \*const linkedItemSelectionModel, QObject \*const parent=nullptr)
- **GPSLinkItemSelectionModel** (QObject \*const parent=nullptr)
- QItemSelectionModel \* **linkedItemSelectionModel** () const
- void **select** (const QItemSelection &selection, QItemSelectionModel::SelectionFlags command) override
- void **select** (const QModelIndex &index, QItemSelectionModel::SelectionFlags command) override
- void **setLinkedItemSelectionModel** (QItemSelectionModel \*const selectionModel)

### Protected Attributes

- GPSLinkItemSelectionModelPrivate \*const **d\_ptr**

## Properties

- QItemSelectionModel \* **linkedItemSelectionModel**

### 6.714.1 Detailed Description

Makes it possible to share a selection in multiple views which do not have the same source model

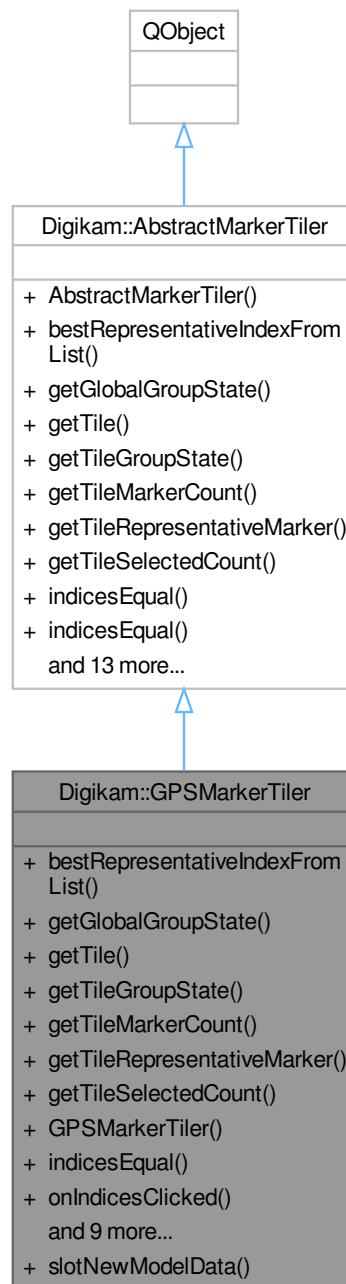
Although Qt documentation, multiple views can share the same QItemSelectionModel, the views then need to have the same source model.

If there is a proxy model between the model and one of the views, or different proxy models in each, this class makes it possible to share the selection between the views.

## 6.715 Digikam::GPSTiler Class Reference

Marker model for storing data needed to display markers on the map. The data is retrieved from [Digikam's](#) database.

Inheritance diagram for Digikam::GPSTiler:



### Public Slots

- void `slotNewModelData` (const `QList< ItemInfo >` &infoList)  
*Receives notifications from the album model about new items.*

### Signals

- void `signalModelFilteredImages` (const `QList< qlonglong >` &imagesId)

## Signals inherited from [Digikam::AbstractMarkerTiler](#)

- void **signalThumbnailAvailableForIndex** (const QVariant &index, const QPixmap &pixmap)
- void **signalTilesOrSelectionChanged** ()

## Public Member Functions

- QVariant **bestRepresentativeIndexFromList** (const QList< QVariant > &indices, const int sortKey) override  
*This function finds the best representative marker from a group of markers. This is needed to display a thumbnail for a marker group.*
- GeoGroupState **getGlobalGroupState** () override
- [AbstractMarkerTiler::Tile](#) \* **getTile** (const [TileIndex](#) &tileIndex, const bool stopIfEmpty) override  
*Returns a pointer to a tile.*
- GeoGroupState **getTileGroupState** (const [TileIndex](#) &tileIndex) override
- int **getTileMarkerCount** (const [TileIndex](#) &tileIndex) override
- QVariant **getTileRepresentativeMarker** (const [TileIndex](#) &tileIndex, const int sortKey) override  
*This function finds the best representative marker from a tile of markers.*
- int **getTileSelectedCount** (const [TileIndex](#) &tileIndex) override
- [GPSMarkerTiler](#) (QObject \*const parent, [ItemFilterModel](#) \*const imageFilterModel, [QItemSelectionModel](#) \*const selectionModel)  
*Constructor.*
- bool **indicesEqual** (const QVariant &a, const QVariant &b) const override  
*This function compares two marker indices.*
- void **onIndicesClicked** (const [ClickInfo](#) &clickInfo) override  
*These can be implemented if you want to react to actions in geolocation interface.*
- QPixmap **pixmapFromRepresentativeIndex** (const QVariant &index, const QSize &size) override  
*This function retrieves the thumbnail for an index.*
- void **prepareTiles** (const [GeoCoordinates](#) &upperLeft, const [GeoCoordinates](#) &lowerRight, int level) override  
*Requests all images inside a given rectangle from the database.*
- void **regenerateTiles** () override
- void **removeCurrentRegionSelection** ()
- void **setActive** (const bool state) override  
*Sets the map active/inactive.*
- void **setPositiveFilterIsActive** (const bool state)
- void **setRegionSelection** (const [GeoCoordinates::Pair](#) &sel)
- [Tile](#) \* **tileNew** () override
- ~[GPSMarkerTiler](#) () override  
*Destructor.*

## Public Member Functions inherited from [Digikam::AbstractMarkerTiler](#)

- [AbstractMarkerTiler](#) (QObject \*const parent=nullptr)
- bool **indicesEqual** (const QList &a, const QList &b, const int upToLevel) const
- bool **isDirty** () const
- virtual void **onIndicesMoved** (const [TileIndex::List](#) &tileIndicesList, const [GeoCoordinates](#) &target←Coordinates, const [QPersistentModelIndex](#) &targetSnapIndex)
- void **resetRootTile** ()
- [Tile](#) \* **rootTile** ()
- void **setDirty** (const bool state=true)
- virtual [TilerFlags](#) **tilerFlags** () const  
*These have to be implemented.*

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractMarkerTiler](#)

- enum **TilerFlag** { **FlagNull** = 0 , **FlagMovable** = 1 }

## 6.715.1 Constructor & Destructor Documentation

### 6.715.1.1 GPSMarkerTiler()

```
Digikam::GPSMarkerTiler::GPSMarkerTiler (
    QObject *const parent,
    ItemFilterModel *const imageFilterModel,
    QItemSelectionModel *const selectionModel ) [explicit]
```

#### Parameters

<i>parent</i>	The parent object
<i>imageFilterModel</i>	The image filter instance
<i>selectionModel</i>	The selection model instance

## 6.715.2 Member Function Documentation

### 6.715.2.1 bestRepresentativeIndexFromList()

```
QVariant Digikam::GPSMarkerTiler::bestRepresentativeIndexFromList (
    const QList< QVariant > & indices,
    const int sortKey ) [override], [virtual]
```

#### Parameters

<i>indices</i>	A list containing markers, obtained by <code>getTileRepresentativeMarker</code> .
<i>sortKey</i>	Sets the criteria for selecting the representative thumbnail, a combination of the <code>SortOptions</code> bits.

#### Returns

Returns the internally used index of the marker.

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.2 getGlobalGroupState()

```
GeoGroupState Digikam::GPSMarkerTiler::getGlobalGroupState ( ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.3 `getTile()`

```
AbstractMarkerTiler::Tile * Digikam::GPSTiler::getTile (
    const TileIndex & tileIndex,
    const bool stopIfEmpty ) [override], [virtual]
```

#### Parameters

<i>tileIndex</i>	The index of a tile.
<i>stopIfEmpty</i>	Determines whether child tiles are also created for empty tiles.

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.4 `getTileGroupState()`

```
GeoGroupState Digikam::GPSTiler::getTileGroupState (
    const TileIndex & tileIndex ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.5 `getTileMarkerCount()`

```
int Digikam::GPSTiler::getTileMarkerCount (
    const TileIndex & tileIndex ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.6 `getTileRepresentativeMarker()`

```
QVariant Digikam::GPSTiler::getTileRepresentativeMarker (
    const TileIndex & tileIndex,
    const int sortKey ) [override], [virtual]
```

#### Parameters

<i>tileIndex</i>	Index of the tile from which the best marker should be found.
<i>sortKey</i>	Sets the criteria for selecting the representative thumbnail, a combination of the SortOptions bits.

#### Returns

Returns the internally used index of the marker.

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.7 `getTileSelectedCount()`

```
int Digikam::GPSTiler::getTileSelectedCount (
    const TileIndex & tileIndex ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.8 indicesEqual()

```
bool Digikam::GPSTiler::indicesEqual (
    const QVariant & a,
    const QVariant & b ) const [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.9 onIndicesClicked()

```
void Digikam::GPSTiler::onIndicesClicked (
    const ClickInfo & clickInfo ) [override], [virtual]
```

Reimplemented from [Digikam::AbstractMarkerTiler](#).

### 6.715.2.10 pixmapFromRepresentativeIndex()

```
QPixmap Digikam::GPSTiler::pixmapFromRepresentativeIndex (
    const QVariant & index,
    const QSize & size ) [override], [virtual]
```

#### Parameters

<i>index</i>	The marker's index.
<i>size</i>	The size of the thumbnail.

#### Returns

If the thumbnail has been loaded in the [ThumbnailLoadThread](#) instance, it is returned. If not, a QPixmap is returned and [ThumbnailLoadThread](#)'s signal named `signalThumbnailLoaded` is emitted when the thumbnail becomes available.

Implements [Digikam::AbstractMarkerTiler](#).

### 6.715.2.11 prepareTiles()

```
void Digikam::GPSTiler::prepareTiles (
    const GeoCoordinates & upperLeft,
    const GeoCoordinates & lowerRight,
    int level ) [override], [virtual]
```

This function calls the database for the images found inside a rectangle defined by `upperLeft` and `lowerRight` points. The images are returned from the database in batches.

#### Parameters

<i>upperLeft</i>	The North-West point.
<i>lowerRight</i>	The South-East point.
<i>level</i>	The requested tiling level.

Implements [Digikam::AbstractMarkerTiler](#).

#### 6.715.2.12 regenerateTiles()

```
void Digikam::GPSTiler::regenerateTiles ( ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

#### 6.715.2.13 setActive()

```
void Digikam::GPSTiler::setActive (
    const bool state ) [override], [virtual]
```

##### Parameters

<i>state</i>	New state of the map, true means active.
--------------	--

Implements [Digikam::AbstractMarkerTiler](#).

#### 6.715.2.14 setPositiveFilterIsActive()

```
void Digikam::GPSTiler::setPositiveFilterIsActive (
    const bool state )
```

#### 6.715.2.15 slotNewModelData

```
void Digikam::GPSTiler::slotNewModelData (
    const QList< ItemInfo > & infoList ) [slot]
```

#### 6.715.2.16 tileNew()

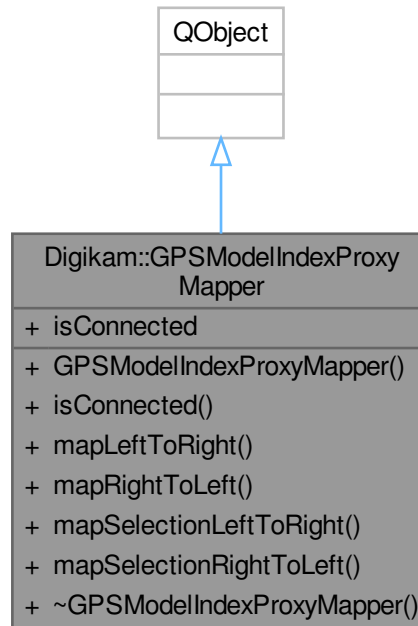
```
AbstractMarkerTiler::Tile * Digikam::GPSTiler::tileNew ( ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).



## 6.716 Digikam::GPSModelIndexProxyMapper Class Reference

Inheritance diagram for Digikam::GPSModelIndexProxyMapper:



### Signals

- void **isConnectedChanged** ()

### Public Member Functions

- **GPSModelIndexProxyMapper** (const QAbstractItemModel \*const leftModel, const QAbstractItemModel \*const rightModel, QObject \*const parent=nullptr)
- bool **isConnected** () const
- QModelIndex [mapLeftToRight](#) (const QModelIndex &index) const
- QModelIndex [mapRightToLeft](#) (const QModelIndex &index) const
- QItemSelection [mapSelectionLeftToRight](#) (const QItemSelection &selection) const
- QItemSelection [mapSelectionRightToLeft](#) (const QItemSelection &selection) const

### Properties

- bool [isConnected](#)

### 6.716.1 Detailed Description

This class facilitates easy mapping of indexes and selections through proxy models.

In a complex system of proxy models there can be a need to map indexes and selections between them, and sometimes to do so without knowledge of the path from one model to another.

If there is a need to map indexes between proxy 2 and proxy 4, a [GPSModelIndexProxyMapper](#) can be created to facilitate mapping of indexes between them.

Note that the aim is to achieve black box connections so that there is no need for application code to know the structure of proxy models in the path between left and right and attempt to manually map them.

The `isConnected` property indicates whether there is a path from the left side to the right side.

### 6.716.2 Member Function Documentation

#### 6.716.2.1 `mapLeftToRight()`

```
QModelIndex Digikam::GPSModelIndexProxyMapper::mapLeftToRight (
    const QModelIndex & index ) const
```

Maps the `index` from the left model to the right model.

#### 6.716.2.2 `mapRightToLeft()`

```
QModelIndex Digikam::GPSModelIndexProxyMapper::mapRightToLeft (
    const QModelIndex & index ) const
```

Maps the `index` from the right model to the left model.

#### 6.716.2.3 `mapSelectionLeftToRight()`

```
QItemSelection Digikam::GPSModelIndexProxyMapper::mapSelectionLeftToRight (
    const QItemSelection & selection ) const
```

Maps the `selection` from the left model to the right model.

#### 6.716.2.4 `mapSelectionRightToLeft()`

```
QItemSelection Digikam::GPSModelIndexProxyMapper::mapSelectionRightToLeft (
    const QItemSelection & selection ) const
```

Maps the `selection` from the right model to the left model.

## 6.716.3 Property Documentation

### 6.716.3.1 isConnected

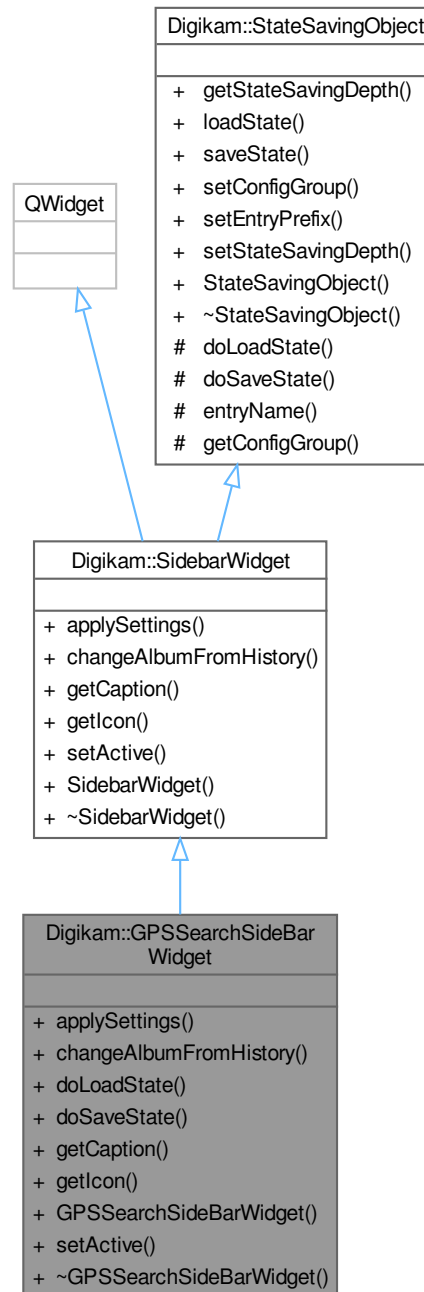
```
bool Digikam::GPSModelIndexProxyMapper::isConnected [read]
```

Indicates whether there is a chain that can be followed from leftModel to rightModel.

This value can change if the sourceModel of an intermediate proxy is changed.

## 6.717 Digikam::GPSSearchSideBarWidget Class Reference

Inheritance diagram for Digikam::GPSSearchSideBarWidget:



### Signals

- void **signalMapSololtems** (const QList< qlonglong > &, const QString &)

## Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

## Public Member Functions

- void [applySettings](#) () override
- void [changeAlbumFromHistory](#) (const QList< [Album](#) \* > &album) override
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- **GPSSearchSideBarWidget** (QWidget \*const parent, [SearchModel](#) \*const searchModel, [SearchModificationHelper](#) \*const searchModificationHelper, [ItemFilterModel](#) \*const imageFilterModel, [QItemSelectionModel](#) \*const itemSelectionModel)
- void [setActive](#) (bool active) override

## Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.717.1 Member Function Documentation

### 6.717.1.1 applySettings()

```
void Digikam::GPSSearchSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).

### 6.717.1.2 changeAlbumFromHistory()

```
void Digikam::GPSSearchSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.717.1.3 doLoadState()

```
void Digikam::GPSSearchSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.717.1.4 doSaveState()

```
void Digikam::GPSSearchSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.717.1.5 getCaption()

```
const QString Digikam::GPSSearchSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.717.1.6 `getIcon()`

```
const QIcon Digikam::GPSSearchSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

QPixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.717.1.7 `setActive()`

```
void Digikam::GPSSearchSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

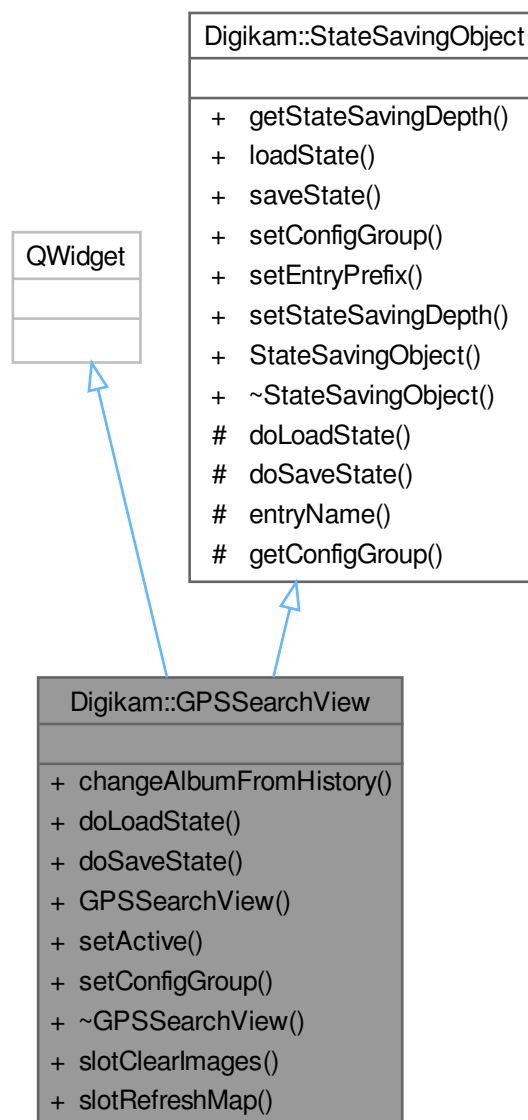
## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.718 Digikam::GPSSearchView Class Reference

Inheritance diagram for Digikam::GPSSearchView:





**Public Slots**

- void **slotClearImages** ()
- void **slotRefreshMap** ()

**Signals**

- void **signalMapSololtems** (const QList< qlonglong > &idList, const QString &id)

**Public Member Functions**

- void **changeAlbumFromHistory** (SAlbum \*const album)
- void **doLoadState** () override
- void **doSaveState** () override
- **GPSSearchView** (QWidget \*const parent, SearchModel \*const searchModel, SearchModificationHelper \*const searchModificationHelper, ItemFilterModel \*const imageFilterModel, QItemSelectionModel \*const itemSelectionModel)  
*Constructor.*
- void **setActive** (bool state)  
*Sets the widget active or inactive.*
- void **setConfigGroup** (const KConfigGroup &group) override

**Public Member Functions inherited from Digikam::StateSavingObject**

- StateSavingDepth **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const StateSavingDepth depth)
- StateSavingObject (QObject \*const host)
- virtual **~StateSavingObject** ()

**Additional Inherited Members****Public Types inherited from Digikam::StateSavingObject**

- enum StateSavingDepth { INSTANCE , DIRECT\_CHILDREN , RECURSIVE }

**Protected Member Functions inherited from Digikam::StateSavingObject**

- QString **entryName** (const QString &base) const
- KConfigGroup **getConfigGroup** () const

**6.718.1 Constructor & Destructor Documentation****6.718.1.1 GPSSearchView()**

```
Digikam::GPSSearchView::GPSSearchView (
    QWidget *const parent,
    SearchModel *const searchModel,
    SearchModificationHelper *const searchModificationHelper,
    ItemFilterModel *const imageFilterModel,
    QItemSelectionModel *const itemSelectionModel ) [explicit]
```

## Parameters

<i>parent</i>	The parent object.
<i>searchModel</i>	The model that stores the searches.
<i>searchModificationHelper</i>	The helper instance to perform the searches.
<i>imageFilterModel</i>	The image model used by displaying the selected images on map.
<i>itemSelectionModel</i>	The selection model corresponding to the imageFilterModel.

## 6.718.2 Member Function Documentation

### 6.718.2.1 doLoadState()

```
void Digikam::GPSSearchView::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.718.2.2 doSaveState()

```
void Digikam::GPSSearchView::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.718.2.3 setActive()

```
void Digikam::GPSSearchView::setActive (
    bool state )
```

Called when the GPSSearch tab becomes the current/not current tab.

## Parameters

<i>state</i>	When true, the widget is enabled.
--------------	-----------------------------------

### 6.718.2.4 setConfigGroup()

```
void Digikam::GPSSearchView::setConfigGroup (
    const KConfigGroup & group ) [override], [virtual]
```

Sets a dedicated config group that will be used to store and reload the state from. If this method is not called, a group based on the object name is used.

You can re-implement this method to pass the group set here to child objects. Don't forget to call this method in your implementation.

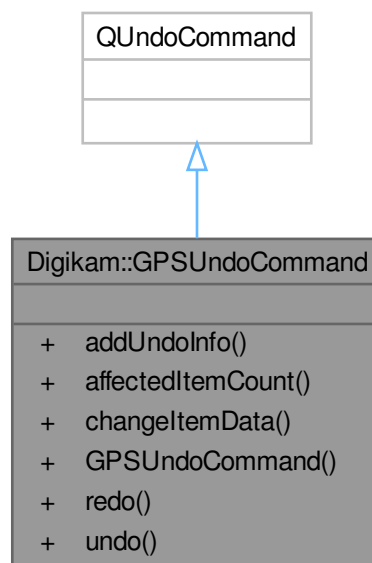
## Parameters

<i>group</i>	config group to use for state saving and restoring
--------------	--

Reimplemented from [Digikam::StateSavingObject](#).

## 6.719 Digikam::GPSUndoCommand Class Reference

Inheritance diagram for Digikam::GPSUndoCommand:



### Classes

- class [UndoInfo](#)

### Public Member Functions

- void **addUndoInfo** (const [UndoInfo](#) &info)
- int **affectedItemCount** () const
- void **changeItemData** (const bool redoIt)
- **GPSUndoCommand** (QUndoCommand \*const parent=nullptr)
- void **redo** () override
- void **undo** () override

## 6.720 Digikam::GPSUndoCommand::UndoInfo Class Reference

### Public Types

- typedef QList< [UndoInfo](#) > **List**

### Public Member Functions

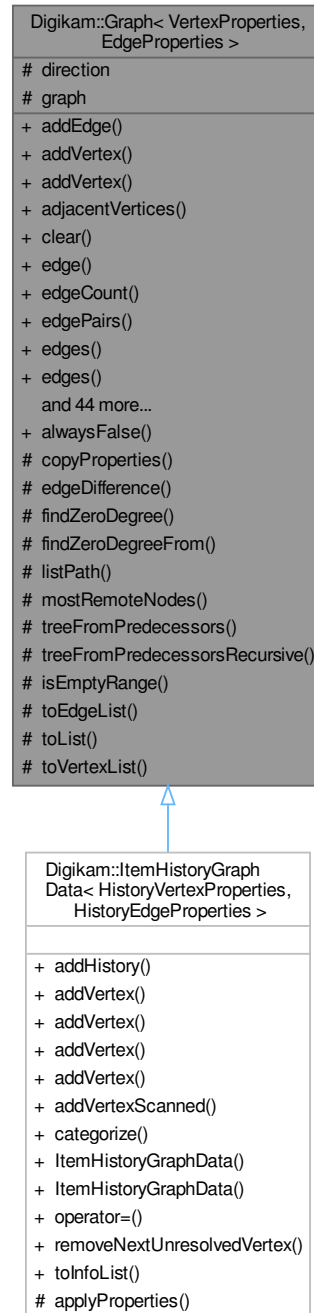
- void **readNewDataFromItem** (const [GPSItemContainer](#) \*const imageItem)
- void **readOldDataFromItem** (const [GPSItemContainer](#) \*const imageItem)
- **UndoInfo** (const QPersistentModelIndex &pModelIndex)

### Public Attributes

- [GPSDataContainer](#) **dataAfter**
- [GPSDataContainer](#) **dataBefore**
- QPersistentModelIndex **modelIndex**
- QList< QList< [TagData](#) > > **newTagList**
- QList< QList< [TagData](#) > > **oldTagList**

## 6.721 Digikam::Graph< VertexProperties, EdgeProperties > Class Template Reference

Inheritance diagram for Digikam::Graph< VertexProperties, EdgeProperties >:



### Classes

- class [DominatorTree](#)

- class [Edge](#)
- class [GraphSearch](#)
- class [Path](#)
- class [Vertex](#)

## Public Types

- typedef graph\_traits::adjacency\_iterator **adjacency\_iter**
- typedef std::pair< adjacency\_iter, adjacency\_iter > **adjacency\_vertex\_range\_t**
- enum [AdjacencyFlags](#) {  
**OutboundEdges** = 1 << 0 , **InboundEdges** = 1 << 1 , **EdgesToLeaf** = 1 << 2 , **EdgesToRoot** = 1 << 3 ,  
**AllEdges** = InboundEdges | OutboundEdges }
- typedef boost::property\_map< GraphContainer, edge\_properties\_t >::const\_type **const\_edge\_property\_↔\_map\_t**
- typedef boost::property\_map< GraphContainer, boost::vertex\_index\_t >::const\_type **const\_vertex\_index\_↔\_map\_t**
- typedef boost::property\_map< GraphContainer, vertex\_properties\_t >::const\_type **const\_vertex\_↔\_property\_map\_t**
- typedef graph\_traits::degree\_size\_type **degree\_t**
- typedef graph\_traits::edge\_iterator **edge\_iter**
- typedef boost::property\_map< GraphContainer, edge\_properties\_t >::type **edge\_property\_map\_t**
- typedef std::pair< edge\_iter, edge\_iter > **edge\_range\_t**
- typedef graph\_traits::edge\_descriptor **edge\_t**
- typedef QPair< [Edge](#), [Edge](#) > **EdgePair**
- typedef boost::graph\_traits< GraphContainer > [graph\\_traits](#)
- typedef boost::adjacency\_list< boost::vecS, boost::vecS, boost::bidirectionalS, boost::property< boost::↔\_vertex\_index\_t, int, boost::property< vertex\_properties\_t, VertexProperties > >, boost::property< edge\_↔\_properties\_t, EdgeProperties > > > **GraphContainer**
- enum **GraphCopyFlags** { **CopyVertexProperties** = 1 << 0 , **CopyEdgeProperties** = 1 << 1 , **CopyAll\_↔\_Properties** = CopyVertexProperties | CopyEdgeProperties }
- typedef graph\_traits::in\_edge\_iterator **in\_edge\_iter**
- typedef boost::inv\_adjacency\_iterator\_generator< GraphContainer, vertex\_t, in\_edge\_iter >::type **inv\_↔\_adjacency\_iter**
- typedef std::pair< inv\_adjacency\_iter, inv\_adjacency\_iter > **inv\_adjacency\_vertex\_range\_t**
- typedef graph\_traits::out\_edge\_iterator **out\_edge\_iter**
- typedef std::pair< out\_edge\_iter, out\_edge\_iter > **out\_edge\_range\_t**
- enum **ReturnOrder** { **BreadthFirstOrder** , **DepthFirstOrder** }
- typedef boost::property\_map< GraphContainer, boost::vertex\_index\_t >::type **vertex\_index\_map\_t**
- typedef graph\_traits::vertex\_iterator **vertex\_iter**
- typedef boost::property\_map< GraphContainer, vertex\_properties\_t >::type **vertex\_property\_map\_t**
- typedef std::pair< vertex\_iter, vertex\_iter > **vertex\_range\_t**
- typedef graph\_traits::vertex\_descriptor **vertex\_t**
- typedef [QMapForAdaptors](#)< [Vertex](#), int > **VertexIntMap**
- typedef boost::associative\_property\_map< [VertexIntMap](#) > **VertexIntMapAdaptor**
- typedef QPair< [Vertex](#), [Vertex](#) > **VertexPair**
- typedef [QMapForAdaptors](#)< [Vertex](#), [Vertex](#) > **VertexVertexMap**
- typedef boost::associative\_property\_map< [VertexVertexMap](#) > **VertexVertexMapAdaptor**

## Public Member Functions

- [Edge](#) **addEdge** (const [Vertex](#) &v1, const [Vertex](#) &v2)
- [Vertex](#) **addVertex** ()
- [Vertex](#) **addVertex** (const [VertexProperties](#) &properties)
- [QList](#)< [Vertex](#) > **adjacentVertices** (const [Vertex](#) &v, [AdjacencyFlags](#) flags=AllEdges) const
- void **clear** ()
- [Edge](#) **edge** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- int **edgeCount** () const
- [QList](#)< [VertexPair](#) > **edgePairs** () const
- [QList](#)< [Edge](#) > **edges** () const
- [QList](#)< [Edge](#) > **edges** (const [Vertex](#) &v, [AdjacencyFlags](#) flags=AllEdges) const
- template<class T >  
[Vertex](#) **findVertexByProperties** (const T &value) const
- const [GraphContainer](#) & **getGraph** () const
- **Graph** (const [Graph](#) &g)
- **Graph** ([MeaningOfDirection](#) dir=ParentToChild)
- bool **hasEdge** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- bool **hasEdges** () const
- bool **hasEdges** (const [Vertex](#) &v, [AdjacencyFlags](#) flags=AllEdges) const
- int **inDegree** (const [Vertex](#) &v) const
- bool **isConnected** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- bool **isEmpty** () const
- bool **isLeaf** (const [Vertex](#) &v) const
- bool **isRoot** (const [Vertex](#) &v) const
- [QList](#)< [Vertex](#) > **leaves** () const
- [QList](#)< [Vertex](#) > **leavesFrom** (const [Vertex](#) &v) const
- [QList](#)< [Vertex](#) > **longestPathTouching** (const [Vertex](#) &v) const
- template<typename LessThan >  
[QList](#)< [Vertex](#) > **longestPathTouching** (const [Vertex](#) &v, LessThan lessThan) const
- [MeaningOfDirection](#) **meaningOfDirection** () const
- [Graph](#) & **operator=** (const [Graph](#) &other)
- int **outDegree** (const [Vertex](#) &v) const
- [EdgeProperties](#) & **properties** (const [Edge](#) &e)
- const [EdgeProperties](#) & **properties** (const [Edge](#) &e) const
- [VertexProperties](#) & **properties** (const [Vertex](#) &v)
- const [VertexProperties](#) & **properties** (const [Vertex](#) &v) const
- [EdgeProperties](#) **properties** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- void **remove** (const [Vertex](#) &v)
- [QList](#)< [Vertex](#) > **roots** () const
- [QList](#)< [Vertex](#) > **rootsOf** (const [Vertex](#) &v) const
- void **setProperties** (const [Edge](#) &e, const [EdgeProperties](#) &props)
- void **setProperties** (const [Vertex](#) &v, const [VertexProperties](#) &props)
- [QMap](#)< [Vertex](#), int > **shortestDistancesFrom** (const [Vertex](#) &v) const
- [QList](#)< [Vertex](#) > **shortestPath** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- [Vertex](#) **source** (const [Edge](#) &e) const
- [Vertex](#) **target** (const [Edge](#) &e) const
- [QList](#)< [Vertex](#) > **topologicalSort** () const
- [Graph](#) **transitiveClosure** ([GraphCopyFlags](#) flags=CopyAllProperties) const
- [Graph](#) **transitiveReduction** ([QList](#)< [Edge](#) > \*removedEdges=0, [GraphCopyFlags](#) flags=CopyAllProperties) const
- int **vertexCount** () const
- *NOTE: for "hasAdjacentVertices", simply use hasEdges(v, flags).*
- [QList](#)< [Vertex](#) > **vertices** () const
- [QList](#)< [Vertex](#) > **verticesBreadthFirst** (const [Vertex](#) &givenRef=[Vertex](#)()) const

- `template<typename LessThan >`  
`QList< Vertex > verticesDepthFirstSorted` (const `Vertex` &givenRef, `LessThan` lessThan) const
- `QList< Vertex > verticesDominatedBy` (const `Vertex` &v, const `Vertex` &root, const `QList< Vertex >` &presortedVertices) const
- `QList< Vertex > verticesDominatedBy` (const `Vertex` &v, const `Vertex` &root, `ReturnOrder` order=`BreadthFirstOrder`) const
- `template<typename LessThan >`  
`QList< Vertex > verticesDominatedByDepthFirstSorted` (const `Vertex` &v, const `Vertex` &root, `LessThan` lessThan) const

### Static Public Member Functions

- `template<typename T >`  
static bool **alwaysFalse** (const T &, const T &)

### Protected Member Functions

- void `copyProperties` (`Graph` &other, `GraphCopyFlags` flags, const `std::vector< vertex_t >` &copiedVertices) const
- `QList< Edge > edgeDifference` (const `Graph` &other, const `std::vector< vertex_t >` &copiedVertices) const
- `QList< Vertex > findZeroDegree` (bool inOrOut) const
- `QList< Vertex > findZeroDegreeFrom` (const `Vertex` &v, bool inOrOut) const
- `QList< Vertex > listPath` (const `Vertex` &root, const `Vertex` &target, const `VertexVertexMap` &predecessors, `MeaningOfDirection` dir=`ParentToChild`) const
- `QList< Vertex > mostRemoteNodes` (const `VertexIntMap` &distances) const
- `QList< Vertex > treeFromPredecessors` (const `Vertex` &v, const `VertexVertexMap` &predecessors) const
- void `treeFromPredecessorsRecursive` (const `Vertex` &v, `QList< Vertex >` &vertices, const `VertexVertexMap` &predecessors) const

### Static Protected Member Functions

- `template<typename range_t >`  
static bool **isEmptyRange** (const range\_t &range)
- `template<typename range_t >`  
static `QList< Edge >` **toEdgeList** (const range\_t &range)
- `template<typename Value , typename range_t >`  
static `QList< Value >` **toList** (const range\_t &range)
- `template<typename range_t >`  
static `QList< Vertex >` **toVertexList** (const range\_t &range)

### Protected Attributes

- `MeaningOfDirection` **direction** = `ParentToChild`
- `GraphContainer` **graph**

## 6.721.1 Detailed Description

```
template<class VertexProperties, class EdgeProperties>
class Digikam::Graph< VertexProperties, EdgeProperties >
```

The graph base class template.



## 6.721.2 Member Typedef Documentation

### 6.721.2.1 graph\_traits

```
template<class VertexProperties , class EdgeProperties >
typedef boost::graph_traits<GraphContainer> Digikam::Graph< VertexProperties, EdgeProperties
>::graph_traits
```

a bunch of graph-specific typedefs that make the long boost types manageable.

## 6.721.3 Member Enumeration Documentation

### 6.721.3.1 AdjacencyFlags

```
template<class VertexProperties , class EdgeProperties >
enum Digikam::Graph::AdjacencyFlags
```

Enumerator

EdgesToLeaf	These resolve to one of the flags above, depending on MeaningOfDirection.
-------------	---

## 6.721.4 Member Function Documentation

### 6.721.4.1 copyProperties()

```
template<class VertexProperties , class EdgeProperties >
void Digikam::Graph< VertexProperties, EdgeProperties >::copyProperties (
    Graph< VertexProperties, EdgeProperties > & other,
    GraphCopyFlags flags,
    const std::vector< vertex_t > & copiedVertices ) const [inline], [protected]
```

According to the given flags and based on the map, copies vertex and edge properties from this to the other graph.

### 6.721.4.2 edgeDifference()

```
template<class VertexProperties , class EdgeProperties >
QList< Edge > Digikam::Graph< VertexProperties, EdgeProperties >::edgeDifference (
    const Graph< VertexProperties, EdgeProperties > & other,
    const std::vector< vertex_t > & copiedVertices ) const [inline], [protected]
```

Returns a list of edges of this graph that have been removed in other. copiedVertices maps the vertices of this graph to other.

### 6.721.4.3 findZeroDegree()

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::findZeroDegree (
    bool inOrOut ) const [inline], [protected]
```

Finds vertex ids of all vertices with zero in- our out-degree.

#### 6.721.4.4 `getGraph()`

```
template<class VertexProperties , class EdgeProperties >
const GraphContainer & Digikam::Graph< VertexProperties, EdgeProperties >::getGraph ( ) const
[inline]
```

Accessing vertices and edges.

#### 6.721.4.5 `leaves()`

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::leaves ( ) const [inline]
```

Returns all leaves, i.e. vertices with no children Takes the graph direction into account.

#### 6.721.4.6 `listPath()`

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::listPath (
    const Vertex & root,
    const Vertex & target,
    const VertexVertexMap & predecessors,
    MeaningOfDirection dir = ParentToChild ) const [inline], [protected]
```

Get a list of vertex ids for the path from root to target, using the given predecessors. Depending on MeaningOfDirection, the ids are listed inverted, from target to root.

#### 6.721.4.7 `longestPathTouching()`

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::longestPathTouching (
    const Vertex & v ) const [inline]
```

Returns the longest path through the graph, starting from a vertex in `roots()`, ending on a vertex in `leaves()`, and passing vertex v. The returned list is given in that order, root - v - leave. If there is more than one candidate for root or leave, `lessThan` is used to determine the first candidate.

#### 6.721.4.8 `mostRemoteNodes()`

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::mostRemoteNodes (
    const VertexIntMap & distances ) const [inline], [protected]
```

Get the list of vertices with the largest value in the given distance map.

#### 6.721.4.9 `roots()`

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::roots ( ) const [inline]
```

Returns all roots, i.e. vertices with no parents. Takes the graph direction into account.

#### 6.721.4.10 rootsOf()

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::rootsOf (
    const Vertex & v ) const [inline]
```

Returns all roots of vertex v. Subset of [roots\(\)](#). I case any leaves have roots that are not roots of v, they will not be contained in this list.

#### 6.721.4.11 shortestDistancesFrom()

```
template<class VertexProperties , class EdgeProperties >
QMap< Vertex, int > Digikam::Graph< VertexProperties, EdgeProperties >::shortestDistancesFrom
(
    const Vertex & v ) const [inline]
```

Returns the shortest distances from [Vertex](#) to all vertices in the graph. If the value is -1, a vertex is not reachable from v.

#### 6.721.4.12 shortestPath()

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::shortestPath (
    const Vertex & v1,
    const Vertex & v2 ) const [inline]
```

Returns the [shortestPath](#) between id1 and id2. If s2 is not reachable from s1, the path is searched from s2 to s1. The returned list always starts with s1, contains the intermediate vertices, and ends with s2. If no path is available, an empty list is returned.

#### 6.721.4.13 toList()

```
template<class VertexProperties , class EdgeProperties >
template<typename Value , typename range_t >
static QList< Value > Digikam::Graph< VertexProperties, EdgeProperties >::toList (
    const range_t & range ) [inline], [static], [protected]
```

Returns a list of vertex ids of vertices in the given range.

#### 6.721.4.14 topologicalSort()

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::topologicalSort ( ) const
[inline]
```

Returns the vertex ids of this graph, in topological order.

**6.721.4.15 transitiveClosure()**

```
template<class VertexProperties , class EdgeProperties >
Graph Digikam::Graph< VertexProperties, EdgeProperties >::transitiveClosure (
    GraphCopyFlags flags = CopyAllProperties ) const [inline]
```

Returns a copy of this graph with all edges added to form the transitive closure.

**6.721.4.16 transitiveReduction()**

```
template<class VertexProperties , class EdgeProperties >
Graph Digikam::Graph< VertexProperties, EdgeProperties >::transitiveReduction (
    QList< Edge > * removedEdges = 0,
    GraphCopyFlags flags = CopyAllProperties ) const [inline]
```

Returns a copy of this graph, with edges removed so that the transitive reduction is formed. Optionally, a list of edges of this graph that have been removed in the returned graph is given.

**6.721.4.17 verticesBreadthFirst()**

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::verticesBreadthFirst (
    const Vertex & givenRef = Vertex() ) const [inline]
```

Orders all vertices of the graph in a breadth-first manner. A single vertex is taken as reference to distinguish main root and side paths. Otherwise the first root is taken as reference.

**6.721.4.18 verticesDepthFirstSorted()**

```
template<class VertexProperties , class EdgeProperties >
template<typename LessThan >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::verticesDepthFirstSorted
(
    const Vertex & givenRef,
    LessThan lessThan ) const [inline]
```

Orders all vertices of the graph in a depth-first manner. When discovering a vertex, the adjacent vertices are sorted with the given lessThan. A single vertex is taken as starting point. If null, the first root is taken as reference.

**6.721.4.19 verticesDominatedBy() [1/2]**

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::verticesDominatedBy (
    const Vertex & v,
    const Vertex & root,
    const QList< Vertex > & presortedVertices ) const [inline]
```

For a vertex *v* reachable from a vertex *root* returns all vertices dominated by *v* starting from *root*. The order is the same as in the given, sorted list of all vertices in this graph (or all vertices expected to be returned. The returned list is the intersection of the dominated vertices and *presortedVertices*, in order of *presortedVertices*). Remove all vertices from the DFS of *v* that are not in the dominated tree.

**6.721.4.20 verticesDominatedBy()** [2/2]

```
template<class VertexProperties , class EdgeProperties >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::verticesDominatedBy (
    const Vertex & v,
    const Vertex & root,
    ReturnOrder order = BreadthFirstOrder ) const [inline]
```

For a vertex v reachable from a vertex root, returns, in depth-first or breadth-first order, all vertices dominated by v starting from root.

**6.721.4.21 verticesDominatedByDepthFirstSorted()**

```
template<class VertexProperties , class EdgeProperties >
template<typename LessThan >
QList< Vertex > Digikam::Graph< VertexProperties, EdgeProperties >::verticesDominatedBy↔
DepthFirstSorted (
    const Vertex & v,
    const Vertex & root,
    LessThan lessThan ) const [inline]
```

For a vertex v reachable from a vertex root all vertices dominated by v starting from root. The returned list is in depth-first order, using root as starting point, and when discovering a vertex, sorting the adjacent vertices with the given lessThan.

**6.722 Digikam::Graph< VertexProperties, EdgeProperties >::DominatorTree Class Reference****Public Member Functions**

- template<class GraphType > void **enter** (const GraphType &graph, const Vertex &v, MeaningOfDirection direction=ParentToChild)

**Public Attributes**

- VertexVertexMap predecessors

**6.723 Digikam::Graph< VertexProperties, EdgeProperties >::Edge Class Reference****Public Member Functions**

- **Edge** (const edge\_t &e)
- bool **isNull** () const
- **operator const edge\_t &** () const
- **operator edge\_t &** ()
- **Edge** & **operator=** (const edge\_t &other)
- bool **operator==** (const edge\_t &other) const
- edge\_t & **toEdge** ()
- const edge\_t & **toEdge** () const

**Protected Attributes**

- edge\_t e
- bool null = true

## 6.724 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch Class Reference

**Classes**

- class [BreadthFirstSearchVisitor](#)
- class [CommonVisitor](#)
- class [DepthFirstSearchVisitor](#)
- class [lessThanMapEdgeToTarget](#)

**Public Member Functions**

- template<class GraphType >  
void **breadthFirstSearch** (const GraphType &graph, const [Vertex](#) &v, bool invertGraph)
- template<class GraphType >  
void **depthFirstSearch** (const GraphType &graph, const [Vertex](#) &v, bool invertGraph)
- template<class GraphType , typename LessThan >  
void **depthFirstSearchSorted** (const GraphType &graph, const [Vertex](#) &v, bool invertGraph, LessThan lessThan)

**Public Attributes**

- QList< [Vertex](#) > vertices

**Protected Member Functions**

- template<class IncidenceGraph , class DFSVisitor , class ColorMap , typename LessThan >  
void **depth\_first\_search\_sorted** (const IncidenceGraph &g, [Vertex](#) u, DFSVisitor &vis, ColorMap color, LessThan lessThan)

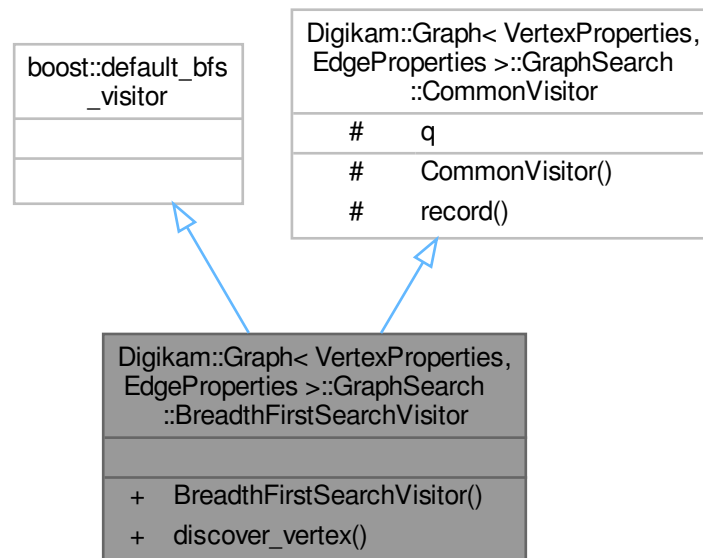
**6.724.1 Member Function Documentation****6.724.1.1 depth\_first\_search\_sorted()**

```
template<class VertexProperties , class EdgeProperties >
template<class IncidenceGraph , class DFSVisitor , class ColorMap , typename LessThan >
void Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::depth_first_search_
sorted (
    const IncidenceGraph & g,
    Vertex u,
    DFSVisitor & vis,
    ColorMap color,
    LessThan lessThan ) [inline], [protected]
```

This is boost's simple, old, recursive DFS algorithm adapted with lessThan. Sort edges. The lessThan we have takes vertices, so we use a lessThan which maps the given edges to their targets, and calls our vertex lessThan.

## 6.725 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::BreadthFirstSearchVisitor Class Reference

Inheritance diagram for Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::BreadthFirstSearchVisitor:



### Public Member Functions

- **BreadthFirstSearchVisitor** ([GraphSearch](#) \*const q)
- `template<typename VertexType , typename GraphType >`  
void **discover\_vertex** (VertexType u, const GraphType &) const

### Additional Inherited Members

#### Protected Member Functions inherited from

#### [Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor](#)

- **CommonVisitor** ([GraphSearch](#) \*const qq)
- void **record** (const [Vertex](#) &v) const

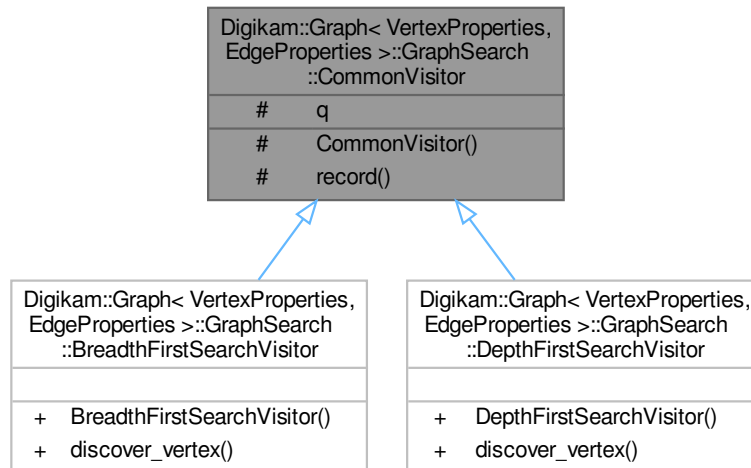
#### Protected Attributes inherited from

#### [Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor](#)

- [GraphSearch](#) \*const q = nullptr

## 6.726 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor Class Reference

Inheritance diagram for Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor:



### Protected Member Functions

- **CommonVisitor** ([GraphSearch](#) \*const qq)
- void **record** (const [Vertex](#) &v) const

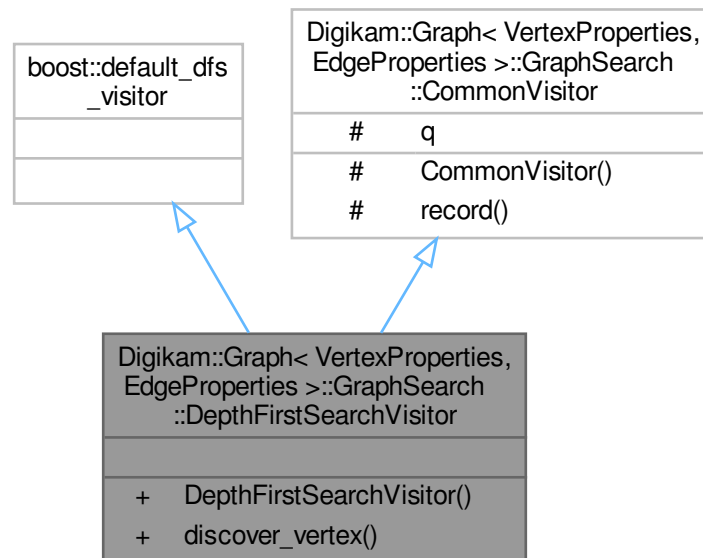
### Protected Attributes

- [GraphSearch](#) \*const **q** = nullptr



## 6.727 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::DepthFirstSearchVisitor Class Reference

Inheritance diagram for Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::DepthFirstSearchVisitor:



### Public Member Functions

- **DepthFirstSearchVisitor** ([GraphSearch](#) \*const q)
- `template<typename VertexType , typename GraphType >`  
void **discover\_vertex** (VertexType u, const GraphType &) const

### Additional Inherited Members

### Protected Member Functions inherited from

#### [Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor](#)

- **CommonVisitor** ([GraphSearch](#) \*const qq)
- void **record** (const [Vertex](#) &v) const

### Protected Attributes inherited from

#### [Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor](#)

- [GraphSearch](#) \*const q = nullptr

## 6.728 Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::lessThanMapEdgeToTarget< GraphType, VertexLessThan > Class Template Reference

### Public Member Functions

- **lessThanMapEdgeToTarget** (const GraphType &gg, VertexLessThan vertexLessThan)
- bool **operator()** (const edge\_descriptor &a, const edge\_descriptor &b)

### Public Attributes

- const GraphType & **g**
- VertexLessThan **vertexLessThan**

## 6.729 Digikam::Graph< VertexProperties, EdgeProperties >::Path Class Reference

### Public Member Functions

- bool **isReachable** (const [Vertex](#) &v) const
- template<class GraphType >  
void **longestPath** (const GraphType &graph, const [Vertex](#) &v)
- template<class GraphType >  
void **shortestPath** (const GraphType &graph, const [Vertex](#) &v)

### Public Attributes

- [VertexIntMap](#) **distances**
- [VertexVertexMap](#) **predecessors**

### 6.729.1 Detailed Description

```
template<class VertexProperties, class EdgeProperties>
class Digikam::Graph< VertexProperties, EdgeProperties >::Path
```

Helper class to find paths through the graph. Call one of the methods and then read the maps.

### 6.729.2 Member Function Documentation

#### 6.729.2.1 longestPath()

```
template<class VertexProperties , class EdgeProperties >
template<class GraphType >
void Digikam::Graph< VertexProperties, EdgeProperties >::Path::longestPath (
    const GraphType & graph,
    const Vertex & v ) [inline]
```

We provide a constant weight of 1.

Invert the default compare method: With greater, we get the longest path.

Will be returned if a node is unreachable.

Store distance and predecessors in QMap, wrapped to serve as property maps.

### 6.729.2.2 shortestPath()

```
template<class VertexProperties , class EdgeProperties >
template<class GraphType >
void Digikam::Graph< VertexProperties, EdgeProperties >::Path::shortestPath (
    const GraphType & graph,
    const Vertex & v ) [inline]
```

we provide a constant weight of 1.

Store distance and predecessors in QMap, wrapped to serve as property maps.

## 6.730 Digikam::Graph< VertexProperties, EdgeProperties >::Vertex Class Reference

### Public Member Functions

- bool **isNull** () const
- **operator const vertex\_t &** () const
- **operator vertex\_t &** ()
- bool **operator!=** (const vertex\_t &other) const
- **Vertex & operator=** (const vertex\_t &other)
- bool **operator==** (const vertex\_t &other) const
- **Vertex** (const vertex\_t &vv)

### Protected Attributes

- vertex\_t v

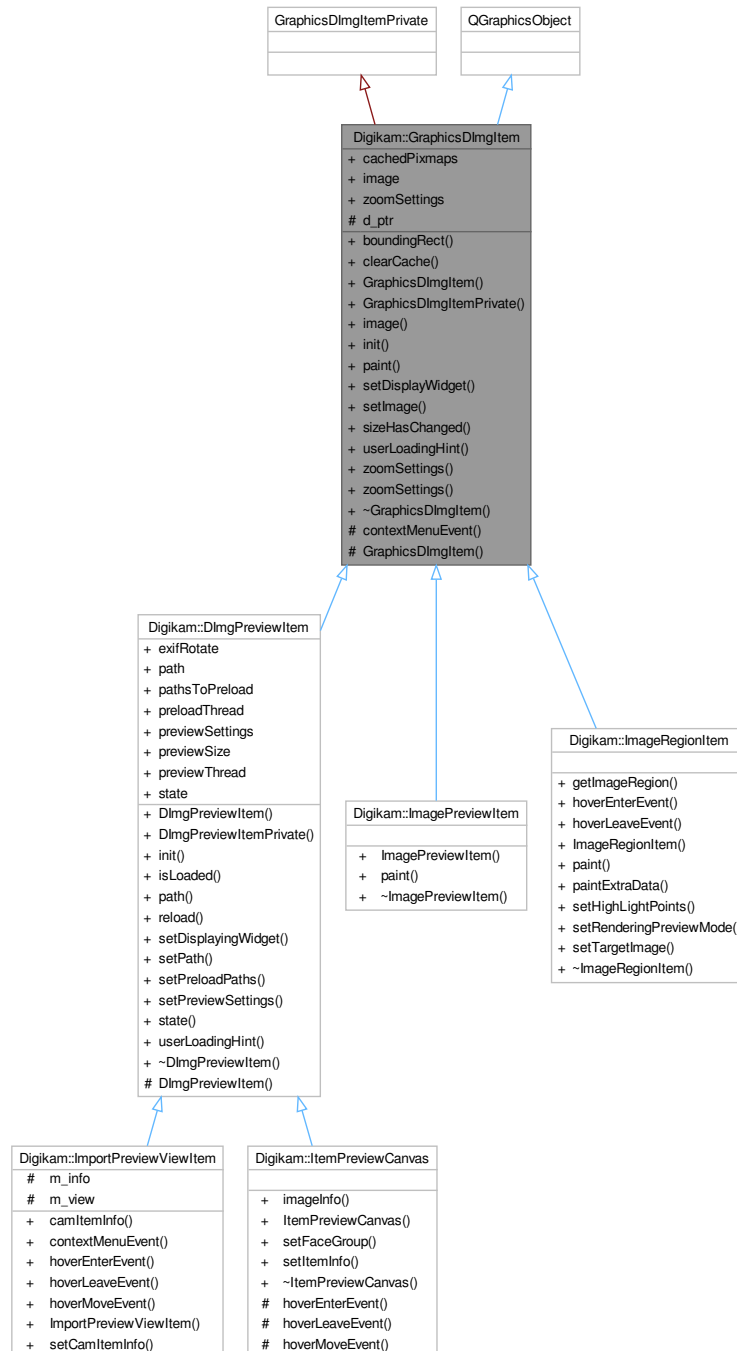
### 6.730.1 Detailed Description

```
template<class VertexProperties, class EdgeProperties>
class Digikam::Graph< VertexProperties, EdgeProperties >::Vertex
```

These two classes provide source-compatible wrappers for the vertex and edge descriptors, providing default construction to null and the isNull() method.

## 6.731 Digikam::GraphicsDImgItem Class Reference

Inheritance diagram for Digikam::GraphicsDImgItem:



### Signals

- void **imageChanged** ()
- void **imageSizeChanged** (const QSizeF &size)
- void **showContextMenu** (QGraphicsSceneContextMenuEvent \*e)

## Public Member Functions

- QRectF **boundingRect** () const override
- void **clearCache** ()
- **GraphicsDImgItem** (QGraphicsItem \*const parent=nullptr)
- **GraphicsDImgItemPrivate** ()=default
- **DImg image** () const
- void **init** ([GraphicsDImgItem](#) \*const q)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override
- void **setDisplayWidget** (QWidget \*const widget)
- void **setImage** (const [DImg](#) &img)
- void **sizeHasChanged** ()
- virtual QString **userLoadingHint** () const
- [ImageZoomSettings](#) \* **zoomSettings** ()
- const [ImageZoomSettings](#) \* **zoomSettings** () const

## Public Attributes

- [CachedPixmap](#) **cachedPixmap**
- [DImg](#) **image**
- [ImageZoomSettings](#) **zoomSettings**

## Protected Member Functions

- void **contextMenuEvent** (QGraphicsSceneContextMenuEvent \*e) override
- **GraphicsDImgItem** ([GraphicsDImgItemPrivate](#) &dd, QGraphicsItem \*const parent)

## Protected Attributes

- [GraphicsDImgItemPrivate](#) \*const **d\_ptr**

## 6.731.1 Member Function Documentation

### 6.731.1.1 setImage()

```
void Digikam::GraphicsDImgItem::setImage (  
    const DImg & img )
```

Sets the [DImg](#) to be drawn by this item. Note: [DImg](#) is explicitly shared, and no copy is automatically taken here.

## 6.732 Digikam::GraphicsDImgView Class Reference

Inheritance diagram for Digikam::GraphicsDImgView:



### Signals

- void **activated** ()
- void **contentsMoved** (bool panningFinished)

- void **contentsMoving** (int, int)
- void **leftButtonClicked** ()
- void **leftButtonDoubleClicked** ()
- void **resized** ()
- void **rightButtonClicked** ()
- void **toNextImage** ()
- void **toPreviousImage** ()
- void **viewportRectChanged** (const QRectF &viewportRect)

### Public Member Functions

- int **contentsX** () const
- int **contentsY** () const
- void **drawText** (QPainter \*p, const QRectF &rect, const QString &text)
- void **fitToWindow** ()
- **GraphicsDImgView** (QWidget \*const parent=nullptr)
- [GraphicsDImgItem](#) \* **item** () const
- [SinglePhotoPreviewLayout](#) \* **layout** () const
- [DImgPreviewItem](#) \* **previewItem** () const
- void **scrollPointOnPoint** (const QPointF &scenePos, const QPoint &viewportPos)
- void **setContentPos** (int x, int y)
- void **setItem** ([GraphicsDImgItem](#) \*const item)
- void **toggleFullScreen** (bool set)
- QRect **visibleArea** () const

### Protected Slots

- void **slotContentsMoved** ()
- void **slotCornerButtonPressed** ()
- void **slotPanIconHidden** ()
- virtual void **slotPanIconSelectionMoved** (const QRect &, bool)

### Protected Member Functions

- virtual bool **acceptsMouseEvent** (QMouseEvent \*e)
- void **continuePanning** (const QPoint &pos)
- void **drawForeground** (QPainter \*painter, const QRectF &rect) override
- void **finishPanning** ()
- void **installPanIcon** ()
- void **mouseDoubleClickEvent** (QMouseEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override
- void **scrollContentsBy** (int dx, int dy) override
- void **setScaleFitToWindow** (bool value)
- void **setShowText** (bool value)
- void **startPanning** (const QPoint &pos)
- void **wheelEvent** (QWheelEvent \*) override

## 6.732.1 Member Function Documentation

### 6.732.1.1 item()

```
GraphicsDImgItem * Digikam::GraphicsDImgView::item ( ) const
```

Return the instance of item set by [setItem\(\)](#).

### 6.732.1.2 previewItem()

```
DImgPreviewItem * Digikam::GraphicsDImgView::previewItem ( ) const
```

Return a cast of item instance of item set by [setItem\(\)](#) as [DImgPreviewItem](#) Note: if you store a [GraphicsDImgItem](#) object using [setItem\(\)](#), this method will return 0.

### 6.732.1.3 scrollPointOnPoint()

```
void Digikam::GraphicsDImgView::scrollPointOnPoint (
    const QPointF & scenePos,
    const QPoint & viewportPos )
```

Scrolls the view such that scenePos (in scene coordinates) is displayed on the viewport at viewportPos (in viewport coordinates). E.g., calling `scrollPointOnPoint(scenePos, viewport()->rect().center())` is equivalent to calling `centerOn(scenePos)`.

### 6.732.1.4 setItem()

```
void Digikam::GraphicsDImgView::setItem (
    GraphicsDImgItem *const item )
```

Store internal instance of item as [GraphicsDImgItem](#). You can store [DImgPreviewItem](#) object also by this method. Use [item\(\)](#) or [previewItem\(\)](#) to get right version. Note: if you store a [GraphicsDImgItem](#) object, [previewItem\(\)](#) will return 0.

## 6.733 Digikam::GreycstorageContainer Class Reference

### Public Types

- enum **INTERPOLATION** { **NearestNeighbor** = 0 , **Linear** , **RungeKutta** }

### Public Member Functions

- void **setInpaintingDefaultSettings** ()
- void **setResizeDefaultSettings** ()
- void **setRestorationDefaultSettings** ()



### Public Attributes

- float **alpha** = 0.6F
- float **amplitude** = 60.0F
- float **anisotropy** = 0.3F
- int **btile** = 4
- float **da** = 30.0F
- float **dl** = 0.6F
- bool **fastApprox** = true
- float **gaussPrec** = 2.0F
- uint **interp** = NearestNeighbor
- uint **nbIter** = 1
- float **sharpness** = 0.7F
- float **sigma** = 1.1F
- int **tile** = 256

## 6.734 Digikam::GreycstorationFilter Class Reference

Inheritance diagram for Digikam::GreycstorationFilter:



### Public Types

- enum `MODE` { `Restore = 0`, `InPainting`, `Resize`, `SimpleResize` }

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Member Functions

- void [cancelFilter](#) () override
- [FilterAction](#) [filterAction](#) () override
- QString [filterIdentifier](#) () const override
- [GreycstorationFilter](#) ([DImg](#) \*const orgImage, const [GreycstorationContainer](#) &settings, int mode=Restore, int newWidth=0, int newHeight=0, const QImage &inPaintingMask=QImage(), QObject \*const parent=nullptr)
- [GreycstorationFilter](#) (QObject \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override
- void [setInPaintingMask](#) (const QImage &inPaintingMask)
- void [setMode](#) (int mode, int newWidth=0, int newHeight=0)
- void [setSettings](#) (const [GreycstorationContainer](#) &settings)
- void [setup](#) ()

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static QString **cimgVersionString** ()
- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

### Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

### Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

### Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int **progress**)
- void **postProgress** (int **progress**)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- [DImg m\\_destImage](#)
- [DImgThreadedFilter \\* m\\_master](#) = nullptr
- [QString m\\_name](#)
- [DImg m\\_orgImage](#)
- [int m\\_progressBegin](#) = 0
- [int m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- [int m\\_progressSpan](#) = 0
- [DImgThreadedFilter \\* m\\_slave](#) = nullptr
- [int m\\_version](#) = 1
- [bool m\\_wasCancelled](#) = false

### 6.734.1 Member Enumeration Documentation

#### 6.734.1.1 MODE

```
enum Digikam::GreycstorationFilter::MODE
```

Enumerator

SimpleResize	Mode to resize image without to use Greycstoration algorithm.
--------------	---

### 6.734.2 Constructor & Destructor Documentation

#### 6.734.2.1 GreycstorationFilter() [1/2]

```
Digikam::GreycstorationFilter::GreycstorationFilter (
    QObject *const parent = nullptr ) [explicit]
```

Constructor without argument. Before to use it, you need to call in order: [setSettings\(\)](#), [setMode\(\)](#), optionally [setInPaintingMask\(\)](#), [setOriginalImage\(\)](#), and necessary [setup\(\)](#) at end.

#### 6.734.2.2 GreycstorationFilter() [2/2]

```
Digikam::GreycstorationFilter::GreycstorationFilter (
    DImg *const orgImage,
    const GreycstorationContainer & settings,
    int mode = Restore,
    int newWidth = 0,
    int newHeight = 0,
    const QImage & inPaintingMask = QImage(),
    QObject *const parent = nullptr )
```

Constructor with all arguments. Ready to use.

## 6.734.3 Member Function Documentation

### 6.734.3.1 `cancelFilter()`

```
void Digikam::GreycstorationFilter::cancelFilter ( ) [override], [virtual]
```

Cancel the threaded computation.

Reimplemented from [Digikam::DImgThreadedFilter](#).

### 6.734.3.2 `filterAction()`

```
FilterAction Digikam::GreycstorationFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.734.3.3 `filterIdentifier()`

```
QString Digikam::GreycstorationFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

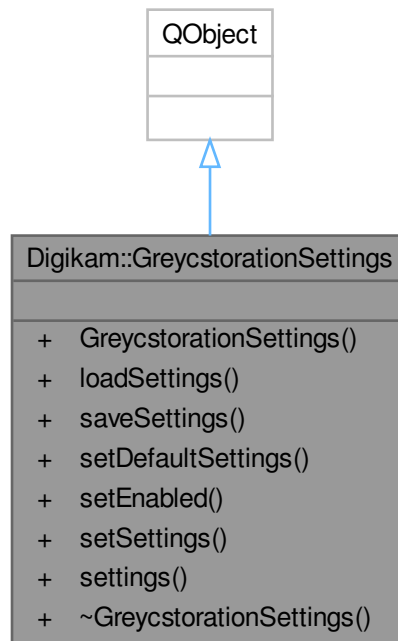
### 6.734.3.4 `readParameters()`

```
void Digikam::GreycstorationFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.735 Digikam::GreycstorationSettings Class Reference

Inheritance diagram for Digikam::GreycstorationSettings:



### Public Member Functions

- **GreycstorationSettings** (QTabWidget \*const parent)
- bool **loadSettings** (QFile &file, const QString &header)
- void **saveSettings** (QFile &file, const QString &header)
- void **setDefaultSettings** (const [GreycstorationContainer](#) &settings)
- void **setEnabled** (bool)
- void **setSettings** (const [GreycstorationContainer](#) &settings)
- [GreycstorationContainer](#) **settings** () const

## 6.736 Digikam::GroupedImagesFinder Class Reference

### Public Member Functions

- [GroupedImagesFinder](#) (const QList< [ItemInfo](#) > &source)

### Public Attributes

- QList< [ItemInfo](#) > **infos**

## 6.736.1 Constructor & Destructor Documentation

### 6.736.1.1 GroupedImagesFinder()

```
Digikam::GroupedImagesFinder::GroupedImagesFinder (  
    const QList< ItemInfo > & source ) [explicit]
```

TODO: Groups should not be resolved in dio, it should be handled in views. This is already done for most things except for drag&drop, which is hard :)



## 6.737 Digikam::GroupIndicatorOverlay Class Reference

Inheritance diagram for Digikam::GroupIndicatorOverlay:



### Signals

- void **showButtonContextMenu** (const QModelIndex &index, QContextMenuEvent \*event)
- void **toggleGroupOpen** (const QModelIndex &index)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- [GroupIndicatorOverlayWidget](#) \* **buttonWidget** () const
- [GroupIndicatorOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotButtonClicked** ()
- void **slotButtonContextMenu** (QContextMenuEvent \*event)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- QWidget \* **createWidget** () override
- void **setActive** (bool) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **updateRating** ()
- void **visualChange** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- virtual void [widgetEnterEvent](#) ()
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- virtual void [widgetLeaveEvent](#) ()
- void [widgetLeaveNotifyMultiple](#) ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

## Protected Attributes

- QPersistentModelIndex [m\\_index](#)

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.737.1 Member Function Documentation

### 6.737.1.1 [checkIndex\(\)](#)

```
bool Digikam::GroupIndicatorOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.737.1.2 createWidget()

```
QWidget * Digikam::GroupIndicatorOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.737.1.3 setActive()

```
void Digikam::GroupIndicatorOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.737.1.4 slotEntered()

```
void Digikam::GroupIndicatorOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and [checkIndex](#) returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.737.1.5 visualChange()

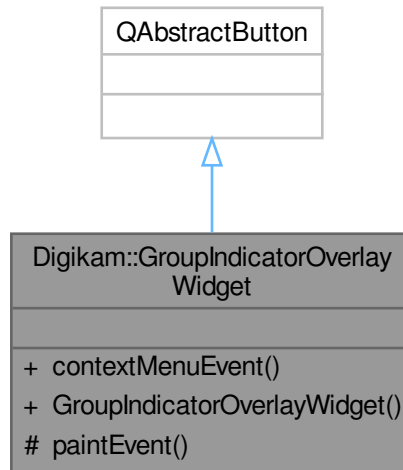
```
void Digikam::GroupIndicatorOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

## 6.738 Digikam::GroupIndicatorOverlayWidget Class Reference

Inheritance diagram for Digikam::GroupIndicatorOverlayWidget:



### Signals

- void **contextMenu** (QContextMenuEvent \*event)

### Public Member Functions

- void **contextMenuEvent** (QContextMenuEvent \*event) override
- **GroupIndicatorOverlayWidget** (QWidget \*const parent=nullptr)

### Protected Member Functions

- void **paintEvent** (QPaintEvent \*) override

## 6.739 Digikam::GroupingViewImplementation Class Reference

Inheritance diagram for Digikam::GroupingViewImplementation:



### Public Member Functions

- [ItemInfoList](#) **getHiddenGroupedInfos** (const [ItemInfoList](#) &infos) const
- virtual bool **hasHiddenGroupedImages** (const [ItemInfo](#) &) const

*must be implemented by parent view*

- bool **needGroupResolving** ([OperationType](#) type, const [ItemInfoList](#) &infos) const
- [ItemInfoList](#) **resolveGrouping** (const [ItemInfoList](#) &infos) const

## 6.739.1 Member Function Documentation

### 6.739.1.1 hasHiddenGroupedImages()

```
virtual bool Digikam::GroupingViewImplementation::hasHiddenGroupedImages (
    const ItemInfo & ) const [inline], [virtual]
```

Reimplemented in [Digikam::ItemThumbnailBar](#), [Digikam::DigikamItemView](#), and [Digikam::TableViewTreeView](#).

## 6.740 Digikam::GroupItemFilterSettings Class Reference

### Public Member Functions

- bool **isAllOpen** () const
- bool **isFiltering** () const
- *Returns if images will be filtered by these criteria at all.*
- bool **isOpen** (qulonglong group) const
- bool **matches** (const [ItemInfo](#) &info) const
- bool **operator==** (const [GroupItemFilterSettings](#) &other) const
- void **setAllOpen** (bool open)
- void **setOpen** (qulonglong group, bool open)
- [DatabaseFields::Set](#) **watchFlags** () const

### Protected Attributes

- bool **m\_allOpen** = false
- [QSet](#)< qulonglong > **m\_openGroups**

## 6.740.1 Member Function Documentation

### 6.740.1.1 matches()

```
bool Digikam::GroupItemFilterSettings::matches (
    const ItemInfo & info ) const
```

Returns true if the given [ItemInfo](#) matches the filter criteria.

### 6.740.1.2 setAllOpen()

```
void Digikam::GroupItemFilterSettings::setAllOpen (
    bool open )
```

Open all groups

### 6.740.1.3 setOpen()

```
void Digikam::GroupItemFilterSettings::setOpen (
    qlonglong group,
    bool open )
```

Open or close a group.

## 6.741 Digikam::GroupStateComputer Class Reference

### Public Member Functions

- void **addFilteredPositiveState** (const GeoGroupState state)
- void **addRegionSelectedState** (const GeoGroupState state)
- void **addSelectedState** (const GeoGroupState state)
- void **addState** (const GeoGroupState state)
- void **clear** ()
- GeoGroupState **getState** () const

## 6.742 Digikam::Haar::Calculator Class Reference

### Public Member Functions

- int **calcHaar** (ImageData \*const imageData, SignatureData \*const sigData)
- void **transform** (ImageData \*const data)

### 6.742.1 Member Function Documentation

#### 6.742.1.1 calcHaar()

```
int Digikam::Haar::Calculator::calcHaar (
    ImageData *const data,
    SignatureData *const sigData )
```

Determines a total of NUM\_COEFS positions in the image that have the largest magnitude (absolute value) in color value. Returns linearized coordinates in sig1, sig2, and sig3. avgl are the [0,0] values. The order of occurrence of the coordinates in sig doesn't matter. Complexity is  $3 \times \text{NUM\_PIXELS}^2 \times 2\log(\text{NUM\_COEFS})$ .

#### 6.742.1.2 transform()

```
void Digikam::Haar::Calculator::transform (
    ImageData *const data )
```

Do the Haar tensorial 2d transform itself. Here input is RGB data [0..255] in Unit arrays. Results are available in a, b, and c. Fully inplace calculation; order of result is interleaved though, but we don't care about that.



## 6.743 Digikam::Haar::ImageData Class Reference

### Public Member Functions

- void `fillImageData` (const `DImg` &image)
- void `fillImageData` (const `QImage` &image)

### Public Attributes

- Unit `data1` [NumberOfPixelsSquared] = { 0.0 }
- Unit `data2` [NumberOfPixelsSquared] = { 0.0 }
- Unit `data3` [NumberOfPixelsSquared] = { 0.0 }

### 6.743.1 Member Function Documentation

#### 6.743.1.1 `fillImageData()` [1/2]

```
void Digikam::Haar::ImageData::fillImageData (
    const DImg & im )
```

Write pixels of a `DImg` in three arrays (one per color channel, pixels linearly)

#### 6.743.1.2 `fillImageData()` [2/2]

```
void Digikam::Haar::ImageData::fillImageData (
    const QImage & im )
```

Write pixels of a `QImage` in three arrays (one per color channel, pixels linearly)

## 6.744 Digikam::Haar::SignatureData Class Reference

### Public Attributes

- double `avg` [3] = { 0.0 }
- Haar::Idx `sig` [3][Haar::NumberOfCoefficients] = { { 0 } }

### 6.744.1 Member Data Documentation

#### 6.744.1.1 `avg`

```
double Digikam::Haar::SignatureData::avg[3] = { 0.0 }
```

YIQ for position [0,0]

### 6.744.1.2 sig

```
Haar::Idx Digikam::Haar::SignatureData::sig[3][Haar::NumberOfCoefficients] = { { 0 } }
```

Y/I/Q positions with largest magnitude

## 6.745 Digikam::Haar::SignatureMap Class Reference

### Public Types

- typedef bool **MapIndexType**

### Public Member Functions

- void **fill** (const Haar::Idx \*const coefs)  
*Load a set of coefficients.*
- bool **operator[]** (Haar::Idx index) const  
*Query if the given index is set. Index must be in the range -16383..16383.*

### Public Attributes

- MapIndexType \* **m\_indexList** = nullptr

### 6.745.1 Detailed Description

This class provides very fast lookup if a certain pixel is set (positive or negative) in the loaded coefficient set.

## 6.746 Digikam::Haar::WeightBin Class Reference

### Public Member Functions

- unsigned char **bin** (int index) const
- unsigned char **binAbs** (int index) const
- [WeightBin](#) ()

### Public Attributes

- unsigned char **m\_bin** [16384] = { 0 }

### 6.746.1 Constructor & Destructor Documentation

#### 6.746.1.1 WeightBin()

```
Digikam::Haar::WeightBin::WeightBin ( )
```

[Setup](#) initial fixed Haar weights that each coefficient represents

## 6.746.2 Member Data Documentation

### 6.746.2.1 m\_bin

```
unsigned char Digikam::Haar::WeightBin::m_bin[16384] = { 0 }
```

Fixed weight mask for pixel positions (i,j). Each entry  $x = i * \text{NUM\_PIXELS} + j$ , gets value  $\max(i,j)$  saturated at 5. To be treated as a constant.

## 6.747 Digikam::Haar::Weights Class Reference

### Public Types

- enum **SketchType** { **ScannedSketch** = 0 , **PaintedSketch** = 1 }

### Public Member Functions

- float **weight** (int weight, int channel) const
- float **weightForAverage** (int channel) const
- **Weights** (SketchType type=ScannedSketch)

## 6.748 Digikam::HaarIface Class Reference

### Classes

- class [Private](#)

### Public Types

- enum **AlbumTagRelation** { **NoMix** = 0 , **Union** = 1 , **Intersection** = 2 , **AlbumExclusive** = 3 , **TagExclusive** = 4 }
- using **DuplicatesResultsMap** = QMap< qlonglong, QPair< double, QList< qlonglong > > >
- enum **DuplicatesSearchRestrictions** { **None** = 0 , **SameAlbum** = 1 , **DifferentAlbum** = 2 }
- enum class **RefImageSelMethod** : unsigned int { **OlderOrLarger** = 0 , **PreferFolder** = 1 , **ExcludeFolder** = 2 , **NewerCreationDate** = 3 , **NewerModificationDate** = 4 }

*The RefImageSelMethod enum Selection method to determine which image will be the reference in the duplicate search.*

- enum **SketchType** { **ScannedSketch** = 0 , **HanddrawnSketch** = 1 }

## Public Member Functions

- QPair< double, QMap< qlonglong, double > > [bestMatchesForImageWithThreshold](#) (const QString &imagePath, double requiredPercentage, double maximumPercentage, const QList< int > &targetAlbums, DuplicatesSearchRestrictions searchResultRestriction=DuplicatesSearchRestrictions::None, SketchType type=ScannedSketch)
- QPair< double, QMap< qlonglong, double > > [bestMatchesForImageWithThreshold](#) (qlonglong imageid, double requiredPercentage, double maximumPercentage, const QList< int > &targetAlbums, DuplicatesSearchRestrictions searchResultRestriction=DuplicatesSearchRestrictions::None, SketchType type=ScannedSketch)
- QMap< qlonglong, double > [bestMatchesForSignature](#) (const QString &signature, const QList< int > &targetAlbums, int numberOfResults=20, SketchType type=ScannedSketch)
- DuplicatesResultsMap [findDuplicates](#) (const QSet< qlonglong > &images2Scan, const QSet< qlonglong >::const\_iterator &rangeBegin, const QSet< qlonglong >::const\_iterator &rangeEnd, [RefImageSelMethod](#) refImageSelectionMethod, const QSet< qlonglong > &refs, double requiredPercentage, double maximumPercentage, DuplicatesSearchRestrictions searchResultRestriction=DuplicatesSearchRestrictions::None, [HaarProgressObserver](#) \*const observer=nullptr)
- bool [fulfillsRestrictions](#) (qlonglong imageid, int albumId, qlonglong originalImageid, int originalAlbumId, const QList< int > &targetAlbums, DuplicatesSearchRestrictions searchResultRestriction)
- void [getBestAndWorstPossibleScore](#) ([Haar::SignatureData](#) \*const querySig, SketchType type, double \*const lowestAndBestScore, double \*const highestAndWorstScore)
- **HaarIface** (const QSet< qlonglong > &images2Scan)
- bool [indexImage](#) (const QString &filename)
- bool [indexImage](#) (const QString &filename, const [DImg](#) &image)
- bool [indexImage](#) (const QString &filename, const QImage &image)
- bool [indexImage](#) (qlonglong imageid, const [DImg](#) &image)
- bool [indexImage](#) (qlonglong imageid, const QImage &image)
- QImage [loadQImage](#) (const QString &filename)
- bool [retrieveSignatureFromDB](#) (qlonglong imageid, [Haar::SignatureData](#) &sig)
- void [setAlbumRootsToSearch](#) (const QList< int > &albumRootIds)
- void [setAlbumRootsToSearch](#) (const QSet< int > &albumRootIds)
- QString [signatureAsText](#) (const QImage &image)

## Static Public Member Functions

- static QSet< qlonglong > [imagesFromAlbumsAndTags](#) (const QList< int > &albums2Scan, const QList< int > &tags2Scan, AlbumTagRelation relation)
- static int [preferredSize](#) ()
- static void [rebuildDuplicatesAlbums](#) (const DuplicatesResultsMap &results, bool isAlbumUpdate)

## 6.748.1 Member Enumeration Documentation

### 6.748.1.1 RefImageSelMethod

```
enum class Digikam::HaarIface::RefImageSelMethod : unsigned int [strong]
```

When adding method here, update also [HaarIface::findDuplicates\(\)](#)

#### Enumerator

OlderOrLarger	Original.
PreferFolder	Prefer select folder to be the reference.
ExcludeFolder	Prefer image not in the selected folder.
NewerCreationDate	Prefer newer creation date image.
NewerModificationDate	Prefer newer modification date image.

## 6.748.2 Member Function Documentation

### 6.748.2.1 bestMatchesForImageWithThreshold() [1/2]

```
QPair< double, QMap< qlonglong, double > > Digikam::HaarIface::bestMatchesForImageWithThreshold (
    const QString & imagePath,
    double requiredPercentage,
    double maximumPercentage,
    const QList< int > & targetAlbums,
    DuplicatesSearchRestrictions searchResultRestriction = DuplicatesSearchRestrictions::None,
    SketchType type = ScannedSketch )
```

Searches the database for the best matches for the specified query image. All matches with a similarity in a given threshold interval are returned. The threshold is in the range requiredPercentage..maximumPercentage.

### 6.748.2.2 bestMatchesForImageWithThreshold() [2/2]

```
QPair< double, QMap< qlonglong, double > > Digikam::HaarIface::bestMatchesForImageWithThreshold (
    qlonglong imageid,
    double requiredPercentage,
    double maximumPercentage,
    const QList< int > & targetAlbums,
    DuplicatesSearchRestrictions searchResultRestriction = DuplicatesSearchRestrictions::None,
    SketchType type = ScannedSketch )
```

Searches the database for the best matches for the specified query image. All matches with a similarity in a given threshold interval are returned. The threshold is in the range requiredPercentage..maximumPercentage.

### 6.748.2.3 findDuplicates()

```
HaarIface::DuplicatesResultsMap Digikam::HaarIface::findDuplicates (
    const QSet< qlonglong > & images2Scan,
    const QSet< qlonglong >::const_iterator & rangeBegin,
    const QSet< qlonglong >::const_iterator & rangeEnd,
    RefImageSelMethod refImageSelectionMethod,
    const QSet< qlonglong > & refs,
    double requiredPercentage,
    double maximumPercentage,
    DuplicatesSearchRestrictions searchResultRestriction = DuplicatesSearchRestrictions::None,
    HaarProgressObserver *const observer = nullptr )
```

Fill a map of duplicates images found over a list of images to scan. For each map item, the result values is list of candidate images which are duplicates of the key image. All images are referenced by id from database. The threshold is in the range 0..1, with 1 meaning identical signature.

**6.748.2.4 fulfillsRestrictions()**

```
bool Digikam::HaarIface::fulfillsRestrictions (
    qlonglong imageId,
    int albumId,
    qlonglong originalImageId,
    int originalAlbumId,
    const QList< int > & targetAlbums,
    DuplicatesSearchRestrictions searchResultRestriction )
```

Checks whether the image with the given *imageId* fulfills all restrictions given in *targetAlbums* and in respect to *searchResultRestriction*.

**6.748.2.5 getBestAndWorstPossibleScore()**

```
void Digikam::HaarIface::getBestAndWorstPossibleScore (
    Haar::SignatureData *const querySig,
    SketchType type,
    double *const lowestAndBestScore,
    double *const highestAndWorstScore )
```

For a given signature, find out the highest and lowest possible score that any other signature could reach, compared to the given signature.

**6.748.2.6 imagesFromAlbumsAndTags()**

```
QSet< qlonglong > Digikam::HaarIface::imagesFromAlbumsAndTags (
    const QList< int > & albums2Scan,
    const QList< int > & tags2Scan,
    AlbumTagRelation relation ) [static]
```

Collects all images from the given album and tag ids according to their relation.

**6.748.2.7 indexImage()**

```
bool Digikam::HaarIface::indexImage (
    const QString & filename )
```

Adds an image to the index in the database.

**6.748.2.8 loadQImage()**

```
QImage Digikam::HaarIface::loadQImage (
    const QString & filename )
```

This method loads a QImage from the given filename.

**Parameters**

<i>filename</i>	the name of the file (path)
-----------------	-----------------------------

**Returns**

A QImage, non-null on success.

**6.748.2.9 rebuildDuplicatesAlbums()**

```
void Digikam::HaarIface::rebuildDuplicatesAlbums (
    const DuplicatesResultsMap & results,
    bool isAlbumUpdate ) [static]
```

This method rebuilds the given SAlbums using the given results.

**Parameters**

<i>results</i>	Map of duplicates images found over a list of images.
<i>isAlbumUpdate</i>	if true update the SAlbums in the database.

**6.748.2.10 retrieveSignatureFromDB()**

```
bool Digikam::HaarIface::retrieveSignatureFromDB (
    qlonglong imageid,
    Haar::SignatureData & sig )
```

Retrieve the Haar signature from database using image id. Return true if item signature exist else false.

**6.748.2.11 setAlbumRootsToSearch()**

```
void Digikam::HaarIface::setAlbumRootsToSearch (
    const QList< int > & albumRootIds )
```

Give a list of albumRoots to which the search shall be limited. Calling with an empty list will disable filtering.

**6.748.2.12 signatureAsText()**

```
QString Digikam::HaarIface::signatureAsText (
    const QImage & image )
```

Calculates the Haar signature, bring it in a form as stored in the DB, and encode it to Ascii data. Can be used for bestMatchesForSignature.

**6.749 Digikam::HaarIface::Private Class Reference****Public Member Functions**

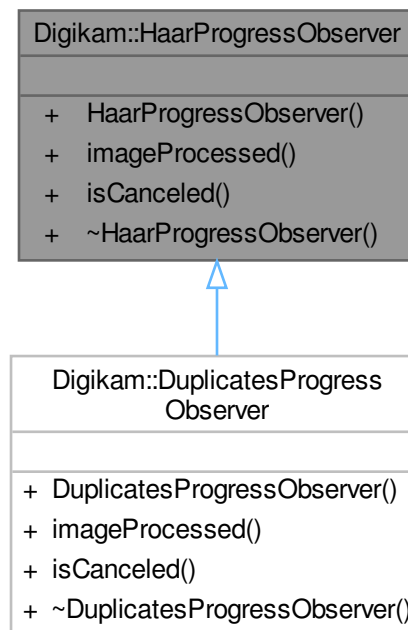
- AlbumCache \* **albumCache** () const
- const QSet< int > & **albumRootsToSearch** () const
- bool **hasSignatureCache** () const
- Haar::ImageData \* **imageData** () const
- void **rebuildSignatureCache** (const QSet< qlonglong > &imageIds={})
- bool **retrieveSignatureFromCache** (qlonglong imageId, Haar::SignatureData &data)
- void **setAlbumRootsToSearch** (const QSet< int > &albumRootIds)
- void **setImageDataFromImage** (const DImg &image)
- void **setImageDataFromImage** (const QImage &image)
- SignatureCache \* **signatureCache** () const

**Public Attributes**

- const QString **signatureQuery** = QString::fromUtf8("SELECT imageid, matrix FROM ImageHaarMatrix;")
- const Haar::WeightBin **weightBin**

**6.750 Digikam::HaarProgressObserver Class Reference**

Inheritance diagram for Digikam::HaarProgressObserver:

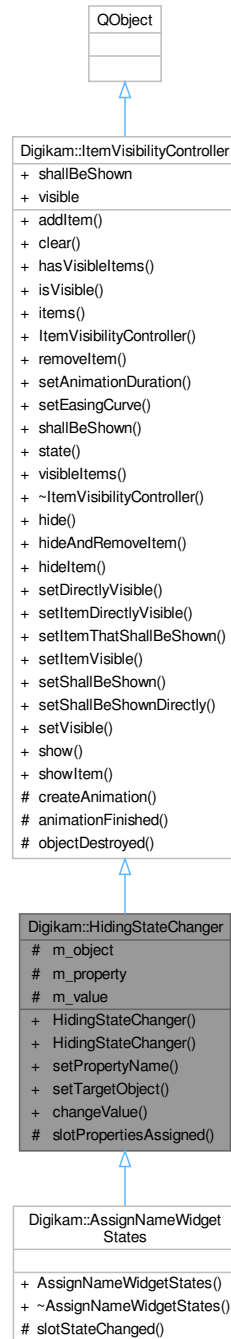
**Public Member Functions**

- virtual void **imageProcessed** (const [ItemInfo](#) &, const QImage &, int)=0
- virtual bool **isCanceled** ()



## 6.751 Digikam::HidingStateChanger Class Reference

Inheritance diagram for Digikam::HidingStateChanger:



### Public Slots

- void **changeValue** (const QVariant &value)

## Public Slots inherited from [Digikam::ItemVisibilityController](#)

- void **hide** ()
- void **hideAndRemoveItem** (QObject \*item)
- void **hideItem** (QObject \*item)
- void **setDirectlyVisible** (bool visible)
- void **setItemDirectlyVisible** (QObject \*item, bool visible)
- void **setItemThatShallBeShown** (QObject \*item)
- void **setItemVisible** (QObject \*item, bool visible)
- void **setShallBeShown** (bool shallBeShown)
- void **setShallBeShownDirectly** (bool shallBeShown)
- void **setVisible** (bool visible)
- void **show** ()
- void **showItem** (QObject \*item)

## Signals

- void **finished** ()
- void **stateChanged** ()

## Signals inherited from [Digikam::ItemVisibilityController](#)

- void **hiddenAndRemoved** (QObject \*item)
- void **propertiesAssigned** (bool visible)
- void **propertiesAssigned** (QObject \*item, bool visible)

## Public Member Functions

- [HidingStateChanger](#) (QObject \*const parent=nullptr)
- [HidingStateChanger](#) (QObject \*const target, const QByteArray &property, QObject \*const parent=nullptr)
- void **setPropertyName** (const QByteArray &propertyName)
- void **setTargetObject** (QObject \*const object)

## Public Member Functions inherited from [Digikam::ItemVisibilityController](#)

- void **addItem** (QObject \*const object)
- void **clear** ()
- bool **hasVisibleItems** ([IncludeFadingOutMode](#) mode=[IncludeFadingOut](#)) const
- bool **isVisible** () const
- QList< QObject \* > **items** () const
- **ItemVisibilityController** (QObject \*const parent=nullptr)
- void **removeItem** (QObject \*const object)
- void **setAnimationDuration** (int msec)
- void **setEasingCurve** (const QEasingCurve &easing)
- bool **shallBeShown** () const
- **State state** () const
- QList< QObject \* > **visibleItems** ([IncludeFadingOutMode](#) mode=[IncludeFadingOut](#)) const

## Protected Slots

- void **slotPropertiesAssigned** (bool)

## Protected Slots inherited from [Digikam::ItemVisibilityController](#)

- void **animationFinished** ()
- void **objectDestroyed** (QObject \*)

## Protected Attributes

- QObject \* **m\_object** = nullptr
- QByteArray **m\_property**
- QVariant **m\_value**

## Additional Inherited Members

## Public Types inherited from [Digikam::ItemVisibilityController](#)

- enum [IncludeFadingOutMode](#) { [IncludeFadingOut](#) , [ExcludeFadingOut](#) }
- enum [State](#) { [Hidden](#) , [FadingIn](#) , [Visible](#) , [FadingOut](#) }

## Protected Member Functions inherited from [Digikam::ItemVisibilityController](#)

- virtual QPropertyAnimation \* [createAnimation](#) (QObject \*item)

## Properties inherited from [Digikam::ItemVisibilityController](#)

- bool **shallBeShown**
- bool **visible**

## 6.751.1 Constructor & Destructor Documentation

### 6.751.1.1 HidingStateChanger() [1/2]

```
Digikam::HidingStateChanger::HidingStateChanger (  
    QObject *const parent = nullptr ) [explicit]
```

This class provides a state change while fading in and out: When `changeValue` is called, first the items are hidden, when this is finished, the property is assigned to the object. Afterwards, the items are shown again. Note that the `targetObject` is not necessarily a controlled item!

### 6.751.1.2 HidingStateChanger() [2/2]

```
Digikam::HidingStateChanger::HidingStateChanger (  
    QObject *const target,  
    const QByteArray & property,  
    QObject *const parent = nullptr )
```

Convenience constructor: Sets target and property name

## 6.751.2 Member Function Documentation

### 6.751.2.1 finished

```
void Digikam::HidingStateChanger::finished ( ) [signal]
```

Emitted when the items were hidden, the target object's property changed, and the items shown again

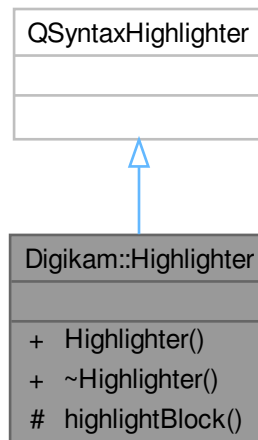
### 6.751.2.2 stateChanged

```
void Digikam::HidingStateChanger::stateChanged ( ) [signal]
```

Emitted when the items were hidden and the target object's property changed

## 6.752 Digikam::Highlighter Class Reference

Inheritance diagram for Digikam::Highlighter:



### Public Member Functions

- **Highlighter** (`QTextDocument *const document`, `Parser *const _parser`)

### Protected Member Functions

- void **highlightBlock** (`const QString &text`) override

## 6.753 Digikam::HistogramBox Class Reference

Inheritance diagram for Digikam::HistogramBox:



### Public Slots

- void **setChannel** (ChannelType channel)
- void **setScale** ([HistogramScale](#) scale)

### Signals

- void **signalChannelChanged** (ChannelType channel)
- void **signalScaleChanged** ([HistogramScale](#) scale)

**Public Member Functions**

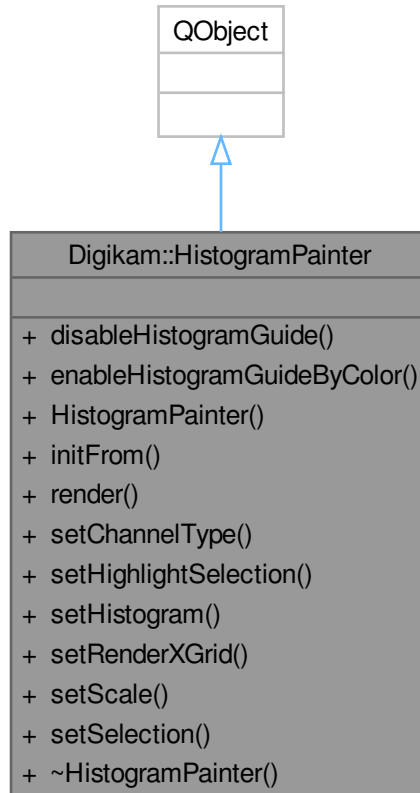
- ChannelType **channel** () const
- [HistogramWidget](#) \* **histogram** () const
- **HistogramBox** (QWidget \*const parent=nullptr, HistogramBoxType type=Digikam::LRGB, bool select←→ Mode=false)
- [HistogramScale](#) **scale** () const
- void **setChannelEnabled** (bool enabled)
- void **setGradientColors** (const QColor &from, const QColor &to)
- void **setGradientVisible** (bool visible)
- void **setHistogramMargin** (int)
- void **setHistogramType** (HistogramBoxType type)
- void **setStatisticsVisible** (bool b)

**Protected Slots**

- void **slotChannelChanged** ()
- void **slotScaleChanged** ()

**6.754 Digikam::HistogramPainter Class Reference**

Inheritance diagram for Digikam::HistogramPainter:



## Public Member Functions

- void `disableHistogramGuide` ()
- void `enableHistogramGuideByColor` (const `DColor` &color)
- `HistogramPainter` (QObject \*const parent)
- void `initFrom` (QWidget \*const widget)
- void `render` (QPixmap &paintDevice)
- void `setChannelType` (ChannelType channelType)
- void `setHighlightSelection` (bool highlightSelection)
- void `setHistogram` (ImageHistogram \*const histogram)
- void `setRenderXGrid` (bool renderXGrid)
- void `setScale` (HistogramScale scale)
- void `setSelection` (double selectionMin, double selectionMax)
- `~HistogramPainter` () override

### 6.754.1 Detailed Description

A class that paints a histogram on a QPixmap.

Warning: before first usage of the render method, you must call `initFrom()` to initialize the painter.

### 6.754.2 Constructor & Destructor Documentation

#### 6.754.2.1 HistogramPainter()

```
Digikam::HistogramPainter::HistogramPainter (
    QObject *const parent ) [explicit]
```

Constructor.

##### Parameters

<i>parent</i>	the parent for Qt's destruction mechanism
---------------	---

#### 6.754.2.2 ~HistogramPainter()

```
Digikam::HistogramPainter::~~HistogramPainter ( ) [override]
```

Destructor.

### 6.754.3 Member Function Documentation

#### 6.754.3.1 disableHistogramGuide()

```
void Digikam::HistogramPainter::disableHistogramGuide ( )
```

Disables the rendering of the color guide.

### 6.754.3.2 enableHistogramGuideByColor()

```
void Digikam::HistogramPainter::enableHistogramGuideByColor (
    const DColor & color )
```

Starts rendering a guide that indicates where in the histogram a specified color can be found.

#### Parameters

<i>color</i>	the color to highlight in the histogram
--------------	---

### 6.754.3.3 initFrom()

```
void Digikam::HistogramPainter::initFrom (
    QWidget *const widget )
```

Stores a widget that is used to initialize the painter used in the next call to render. Therefore you must ensure that this widget will not be destroyed as long as you want to use the render method without a new call to this method!!!

#### Parameters

<i>widget</i>	the widget to initialize painting from
---------------	--

### 6.754.3.4 render()

```
void Digikam::HistogramPainter::render (
    QPixmap & paintDevice )
```

Renders the given histogram on the pixmap. The whole size of the pixmap is used for the histogram.

You must ensure that once before using this method a call to `initFrom` was made and the widget given in that call is still present.

#### Parameters

<i>paintDevice</i>	pixmap to paint the histogram on
--------------------	----------------------------------

### 6.754.3.5 setChannelType()

```
void Digikam::HistogramPainter::setChannelType (
    ChannelType channelType )
```

Set the channel type to render with the next call to render.

#### Parameters

<i>channelType</i>	channel type to render
--------------------	------------------------



### 6.754.3.6 setHighlightSelection()

```
void Digikam::HistogramPainter::setHighlightSelection (
    bool highlightSelection )
```

Decide whether to highlight a specified selection in the histogram or not. The selection must be defined with `setHighlightSelection`.

#### Parameters

<i>highlightSelection</i>	if true, a selection will be highlighted
---------------------------	--

### 6.754.3.7 setHistogram()

```
void Digikam::HistogramPainter::setHistogram (
    ImageHistogram *const histogram )
```

Set the histogram to paint with the next call to render.

#### Parameters

<i>histogram</i>	an existing pointer to a histogram to paint on next call to render. The histogram must still exist at that call.
------------------	--

### 6.754.3.8 setRenderXGrid()

```
void Digikam::HistogramPainter::setRenderXGrid (
    bool renderXGrid )
```

Decide whether to render a separation of the histogram in x direction.

#### Parameters

<i>renderXGrid</i>	if true, a separation at some significant value in x direction is rendered.
--------------------	---

### 6.754.3.9 setScale()

```
void Digikam::HistogramPainter::setScale (
    HistogramScale scale )
```

Set the scale to paint the histogram with.

#### Parameters

<i>scale</i>	scal to paint histogram with
--------------	------------------------------

### 6.754.3.10 setSelection()

```
void Digikam::HistogramPainter::setSelection (
    double selectionMin,
    double selectionMax )
```

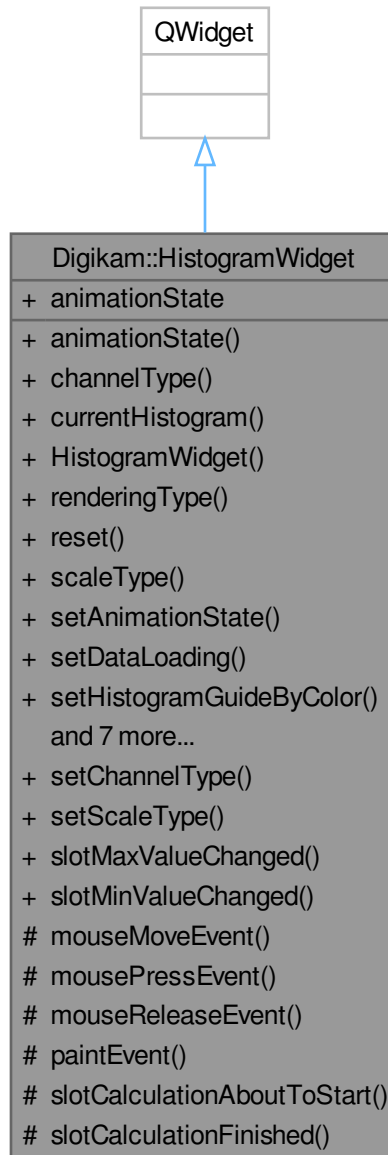
Sets the selection to highlight.

#### Parameters

<i>selectionMin</i>	0 <= value <= 1, percent of the histogram width to start highlighting as percent. Ensure that this value is smaller then <i>selectionMax</i> .
<i>selectionMax</i>	0 <= value <= 1, percent of the histogram width to end highlighting as percent. Ensure that this value is greater then <i>selectionMin</i> .

## 6.755 Digikam::HistogramWidget Class Reference

Inheritance diagram for Digikam::HistogramWidget:



### Public Slots

- void **setChannelType** (ChannelType channel)
- void **setScaleType** (HistogramScale scale)
- void **slotMaxValueChanged** (int max)
- void **slotMinValueChanged** (int min)

## Signals

- void **signalHistogramComputationDone** (bool)
- void **signalHistogramComputationFailed** ()
- void **signalIntervalChanged** (int min, int max)
- void **signalMaximumValueChanged** (int)

## Public Member Functions

- int **animationState** () const
- ChannelType **channelType** () const
- ImageHistogram \* **currentHistogram** () const
- HistogramWidget (int w, int h, QWidget \*const parent=nullptr, bool selectMode=true, bool showProgress=true, bool statisticsVisible=false)
- HistogramRenderingType **renderingType** () const
- void **reset** ()
- HistogramScale **scaleType** () const
- void **setAnimationState** (int animationState)
- void **setDataLoading** ()
- void **setHistogramGuideByColor** (const DColor &color)
- void **setLoadingFailed** ()
- void **setRenderingType** (HistogramRenderingType type)
- void **setStatisticsVisible** (bool b)
- void **stopHistogramComputation** ()
- void **updateData** (const DImg &img, const DImg &sel=DImg(), bool showProgress=true)
- void **updateSelectionData** (const DImg &sel, bool showProgress=true)

## Protected Slots

- void **slotCalculationAboutToStart** ()
- void **slotCalculationFinished** (bool success)

## Protected Member Functions

- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override

## Properties

- int **animationState**

## 6.755.1 Constructor & Destructor Documentation

### 6.755.1.1 HistogramWidget()

```
Digikam::HistogramWidget::HistogramWidget (
    int w,
    int h,
    QWidget *const parent = nullptr,
    bool selectMode = true,
    bool showProgress = true,
    bool statisticsVisible = false )
```

Standard constructor. Needed to use [updateData\(\)](#) methods after to create valid instance.

## 6.755.2 Member Function Documentation

### 6.755.2.1 currentHistogram()

```
ImageHistogram * Digikam::HistogramWidget::currentHistogram ( ) const
```

Currently rendered histogram, depending on current rendering type.

### 6.755.2.2 stopHistogramComputation()

```
void Digikam::HistogramWidget::stopHistogramComputation ( )
```

Stop current histogram computations.

### 6.755.2.3 updateData()

```
void Digikam::HistogramWidget::updateData (
    const DImg & img,
    const DImg & sel = DImg(),
    bool showProgress = true )
```

Update full image histogram data methods.

### 6.755.2.4 updateSelectionData()

```
void Digikam::HistogramWidget::updateSelectionData (
    const DImg & sel,
    bool showProgress = true )
```

Update image selection histogram data methods.

## 6.756 Digikam::HistoryEdgeProperties Class Reference

### Public Member Functions

- [HistoryEdgeProperties](#) & **operator+=** (const [FilterAction](#) &action)

### Public Attributes

- `QList< FilterAction > actions`

### 6.756.1 Detailed Description

Every edge has one associated object of this class.

For two vertices v1, v2 with and edge e, v1 -> v2, describes the actions necessary to create v2 from v2: v1 -> actions[0] -> ... -> actions[n] = v2.

## 6.757 Digikam::HistoryImageld Class Reference

### Public Types

- enum [Type](#) {  
**InvalidType** = 0 , **Original** = 1 << 0 , **Intermediate** = 1 << 1 , **Source** = 1 << 2 ,  
**Current** = 1 << 3 }

### Public Member Functions

- QDateTime **creationDate** () const
- QString **fileName** () const  
*If a file on disk is referenced: Returns the file name (without folder)*
- QString **filePath** () const  
*If a file on disk is referenced: Returns the full file path (folder + filename)*
- qlonglong **fileSize** () const
- bool **hasCreationDate** () const
- bool **hasFileName** () const
- bool **hasFileOnDisk** () const
- bool **hasUniqueHashIdentifier** () const
- bool **hasUuid** () const
- [HistoryImageld](#) ()=default  
*Creates an invalid [HistoryImageld](#).*
- **HistoryImageld** (const QString &uuid, [Type](#) type=[Current](#))  
*Creates an id with the given UUID and type.*
- bool **isCurrentFile** () const
- bool **isIntermediateFile** () const
- bool **isOriginalFile** () const
- bool **isSourceFile** () const
- bool **isValid** () const  
*A valid id needs at least a valid type and a UUID or a filename.*
- bool **operator==** (const [HistoryImageld](#) &other) const
- QString **originalUuid** () const
- QString **path** () const  
*If a file on disk is referenced: Returns the path, without filename, with a trailing slash.*
- void **setCreationDate** (const QDateTime &creationDate)
- void **setFileName** (const QString &fileName)
- void **setPath** (const QString &path)
- void **setPathOnDisk** (const QString &filePath)
- void **setType** ([HistoryImageld::Type](#) type)
- void **setUniqueHash** (const QString &uniqueHash, qlonglong fileSize)
- void **setUuid** (const QString &uuid)
- [Type](#) **type** () const
- QString **uniqueHash** () const
- QString **uuid** () const

## Public Attributes

- QDateTime **m\_creationDate**  
*The creationDate of the original image.*
- QString **m\_fileName**  
*The filename of the referred file.*
- QString **m\_filePath**  
*The path of the referred file (NOTE: without file name!, including trailing slash)*
- qlonglong **m\_fileSize** = 0  
*The file size of the referred file.*
- QString **m\_originalUUID**
- Type **m\_type** = InvalidType  
*Type of this History Image Id.*
- QString **m\_uniqueHash**  
*The uniqueHash of the referred file.*
- QString **m\_uuid**

## 6.757.1 Member Enumeration Documentation

### 6.757.1.1 Type

```
enum Digikam::HistoryImageId::Type
```

#### Enumerator

Original	The original file (typically created by a camera)
Intermediate	A file created during the editing the history, between the original file and the current file.
Source	When a file is created from multiple files, there can be no direct original (panorama) but multiple sources, or one direct original and some other, additional source files. To record source files outside of the direct history, this type is used.
Current	The "current" file. This is a special entry: It refers to the file from which this history was read. It need not be written to the file, because it describes the file itself. There is typically exactly one current entry if the history is associated with an image; there can be no current entry.

## 6.757.2 Constructor & Destructor Documentation

### 6.757.2.1 HistoryImageId()

```
Digikam::HistoryImageId::HistoryImageId ( ) [default]
```

Note: In this class, the Type is used as a simple enum, but it is also prepared for usage as flags.

## 6.757.3 Member Data Documentation

### 6.757.3.1 m\_originalUUID

```
QString Digikam::HistoryImageId::m_originalUUID
```

A unique identifier designating the *original image* from which the referred image was created. Typically, this is a RAW or JPEG created by the camera in the moment of taking the photograph.

### 6.757.3.2 m\_uuid

QString Digikam::HistoryImageId::m\_uuid

A unique identifier for the referred file. This id shall be changed each time the image is edited.

## 6.758 Digikam::HistoryVertexProperties Class Reference

### Public Member Functions

- bool **alwaysMarkedAs** ([HistoryImageId::Type](#)) const
- [ItemInfo](#) **firstItemInfo** () const
- bool **markedAs** ([HistoryImageId::Type](#)) const
- [HistoryVertexProperties](#) & **operator+=** (const [HistoryImageId](#) &info)
- [HistoryVertexProperties](#) & **operator+=** (const [ItemInfo](#) &info)
- [HistoryVertexProperties](#) & **operator+=** (const QString &uuid)
- bool **operator==** (const [HistoryImageId](#) &info) const
- bool **operator==** (const [ItemInfo](#) &info) const
- bool **operator==** (const QString &uuid) const
- bool **operator==** (qulonglong id) const

### Public Attributes

- QList< [ItemInfo](#) > **infos**
- QList< [HistoryImageId](#) > **referredImages**
- QString **uuid**

### 6.758.1 Detailed Description

Every vertex has one associated object of this class

All entries in a vertex refer to *identical* images. There can be multiple referred images in a history entry. Each single [HistoryImageId](#) can resolve into none, one, or multiple [ItemInfos](#). So there is no mapping between the two fields here.

If an image is created from multiple source images (panorama etc.), there will be one vertex per source image!

## 6.759 Digikam::HotPixelContainer Class Reference

### Public Types

- enum **Direction** { **TWODIM\_DIRECTION** = 0 , **VERTICAL\_DIRECTION** = 1 , **HORIZONTAL\_DIRECTION** = 2 }
- enum **InterpolationMethod** { **AVERAGE\_INTERPOLATION** = 0 , **LINEAR\_INTERPOLATION** = 1 , **QUADRATIC\_INTERPOLATION** = 2 , **CUBIC\_INTERPOLATION** = 3 }



### Public Member Functions

- bool **isDefault** () const
- bool **operator==** (const [HotPixelContainer](#) &other) const
- void **writeToFilterAction** ([FilterAction](#) &action, const QString &prefix=QString()) const

### Static Public Member Functions

- static [HotPixelContainer](#) **fromFilterAction** (const [FilterAction](#) &action, const QString &prefix=QString())

### Public Attributes

- [QUrl](#) **blackFrameUrl**
- [InterpolationMethod](#) **filterMethod**
- [QList](#)< [HotPixelProps](#) > **hotPixelsList**

## 6.760 Digikam::HotPixelFixer Class Reference

Inheritance diagram for Digikam::HotPixelFixer:



### Public Member Functions

- [Digikam::FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **HotPixelFixer** ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [HotPixelContainer](#) &settings)
- **HotPixelFixer** ([QObject](#) \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static [QString](#) **DisplayableName** ()
- static [QString](#) **FilterIdentifier** ()
- static [QList](#)< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.760.1 Member Function Documentation

### 6.760.1.1 filterAction()

`Digikam::FilterAction` `Digikam::HotPixelFixer::filterAction ( )` `[override]`, `[virtual]`

Returns the action description corresponding to currently set options.

Implements `Digikam::DImgThreadedFilter`.

### 6.760.1.2 filterIdentifier()

`QString` `Digikam::HotPixelFixer::filterIdentifier ( )` `const` `[inline]`, `[override]`, `[virtual]`

Return the identifier for this filter in the image history.

Implements `Digikam::DImgThreadedFilter`.

### 6.760.1.3 readParameters()

`void` `Digikam::HotPixelFixer::readParameters (`  
`const` `FilterAction` `& action )` `[override]`, `[virtual]`

Implements `Digikam::DImgThreadedFilter`.

## 6.761 Digikam::HotPixelProps Class Reference

### Public Member Functions

- `bool` `fromString` (`const` `QString` `&str`)
- `int` `height` (`)` `const`
- `bool` `operator==` (`const` `HotPixelProps` `&p`) `const`
- `QString` `toString` (`)` `const`
- `int` `width` (`)` `const`
- `int` `x` (`)` `const`
- `int` `y` (`)` `const`

### Static Public Member Functions

- `static` `QList``<` `HotPixelProps` `>` `fromStringList` (`const` `QStringList` `&hplst`)
- `static` `QStringList` `toStringList` (`const` `QList``<` `HotPixelProps` `>` `&lst`)

### Public Attributes

- `int` `luminosity`
- `QRect` `rect`

## 6.761.1 Member Function Documentation

### 6.761.1.1 operator==( )

```
bool Digikam::HotPixelProps::operator==(
    const HotPixelProps & p ) const
```

NOTE:we can say they're same hotpixel spot if they touch (next to) each other horizontally or vertically, not diagonal corners

## 6.762 Digikam::HotPixelSettings Class Reference

Inheritance diagram for Digikam::HotPixelSettings:



### Signals

- void **signalHotPixels** (const QPolygon &pointList)
- void **signalSettingsChanged** ( )

### Public Member Functions

- QString **configGroupName** () const
- [HotPixelContainer](#) **defaultSettings** () const
- **HotPixelSettings** (QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const [HotPixelContainer](#) &settings)
- [HotPixelContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.763 Digikam::HotPixelsWeights Class Reference

### Public Member Functions

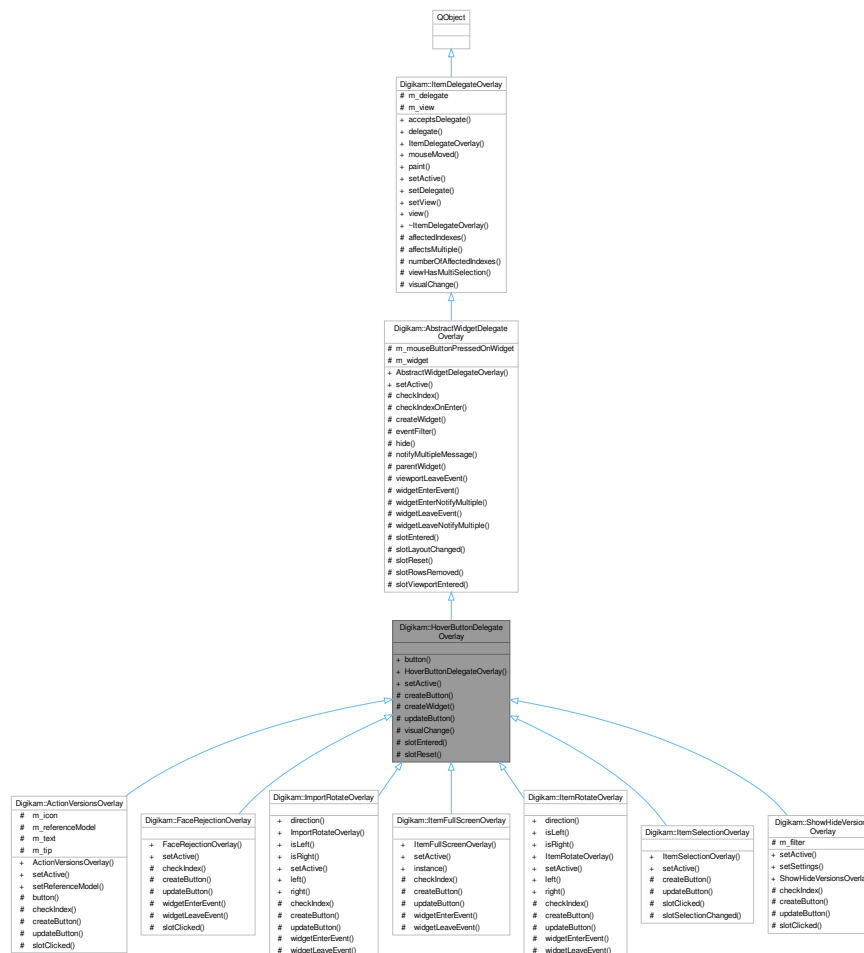
- void **calculateHotPixelsWeights** ()
- unsigned int **height** () const
- **HotPixelsWeights** (const [HotPixelsWeights](#) &w)
- [HotPixelsWeights](#) & **operator=** (const [HotPixelsWeights](#) &w)
- bool **operator==** (const [HotPixelsWeights](#) &ws) const
- double \*\* **operator[]** (int n) const
- unsigned int **polynomeOrder** () const
- const QList< QPoint > **positions** () const
- void **setHeight** (int h)
- void **setPolynomeOrder** (int order)
- void **setTwoDim** (bool td)
- void **setWidth** (int w)
- bool **twoDim** () const
- unsigned int **width** () const

### Protected Member Functions

- int **coefficientNumber** () const
- double \*\*\* **weightMatrices** () const

## 6.764 Digikam::HoverButtonDelegateOverlay Class Reference

Inheritance diagram for Digikam::HoverButtonDelegateOverlay:



### Public Member Functions

- [ItemViewHoverButton](#) \* **button** () const
- **HoverButtonDelegateOverlay** (QObject \*const parent)
- void [setActive](#) (bool active) override

### Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

### Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const



### Protected Slots

- void **slotEntered** (const QModelIndex &index) override
- void **slotReset** () override

### Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void [slotEntered](#) (const QModelIndex &index)
- virtual void **slotLayoutChanged** ()
- virtual void [slotReset](#) ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

### Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

#### Protected Member Functions

- virtual [ItemViewHoverButton](#) \* [createButton](#) ()=0
- QWidget \* [createWidget](#) () override
- virtual void [updateButton](#) (const QModelIndex &index)=0
- void [visualChange](#) () override

### Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual bool [checkIndex](#) (const QModelIndex &index) const
- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- virtual void [widgetEnterEvent](#) ()
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

#### Additional Inherited Members

### Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool `m_mouseButtonPressedOnWidget` = false
- `QWidget * m_widget` = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- `QAbstractItemDelegate * m_delegate` = nullptr
- `QAbstractItemView * m_view` = nullptr

## 6.764.1 Member Function Documentation

### 6.764.1.1 `createButton()`

```
virtual ItemViewHoverButton * Digikam::HoverButtonDelegateOverlay::createButton ( ) [protected],
[pure virtual]
```

Create your widget here. Pass `view()` as parent.

Implemented in [Digikam::FaceRejectionOverlay](#), [Digikam::ItemFullScreenOverlay](#), [Digikam::ItemRotateOverlay](#), [Digikam::ItemSelectionOverlay](#), [Digikam::ShowHideVersionsOverlay](#), [Digikam::ActionVersionsOverlay](#), and [Digikam::ImportRotateOverlay](#).

### 6.764.1.2 `createWidget()`

```
QWidget * Digikam::HoverButtonDelegateOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass `parentWidget()` as parent widget. Ownership of the object is passed. It will be deleted in `setActive(false)`.

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.764.1.3 `setActive()`

```
void Digikam::HoverButtonDelegateOverlay::setActive (
    bool active ) [override], [virtual]
```

Will call [createButton\(\)](#).

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

Reimplemented in [Digikam::ItemFullScreenOverlay](#), [Digikam::ItemRotateOverlay](#), [Digikam::ItemSelectionOverlay](#), [Digikam::ShowHideVersionsOverlay](#), and [Digikam::ImportRotateOverlay](#).

### 6.764.1.4 `updateButton()`

```
virtual void Digikam::HoverButtonDelegateOverlay::updateButton (
    const QModelIndex & index ) [protected], [pure virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implemented in [Digikam::FaceRejectionOverlay](#), [Digikam::ItemFullScreenOverlay](#), [Digikam::ItemRotateOverlay](#), [Digikam::ItemSelectionOverlay](#), [Digikam::ShowHideVersionsOverlay](#), [Digikam::ActionVersionsOverlay](#), and [Digikam::ImportRotateOverlay](#).

### 6.764.1.5 visualChange()

```
void Digikam::HoverButtonDelegateOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

## 6.765 Digikam::HSLContainer Class Reference

### Public Attributes

- double **hue** = 0.0
- double **lightness** = 0.0
- double **saturation** = 0.0
- double **vibrance** = 0.0

## 6.766 Digikam::HSLFilter Class Reference

Inheritance diagram for Digikam::HSLFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **HSLFilter** ([DImg](#) \*const orgImage, [QObject](#) \*const parent=nullptr, const [HSLContainer](#) &settings=[HSLContainer](#)())
- **HSLFilter** ([QObject](#) \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static [QString](#) **DisplayableName** ()
- static [QString](#) **FilterIdentifier** ()
- static [QList](#)< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.766.1 Member Function Documentation

### 6.766.1.1 filterAction()

`FilterAction` Digikam::HSLFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.766.1.2 filterIdentifier()

`QString` Digikam::HSLFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

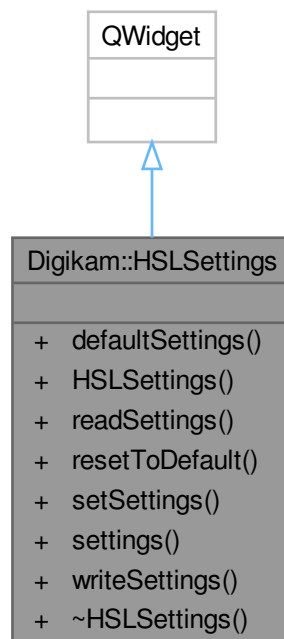
### 6.766.1.3 readParameters()

```
void Digikam::HSLFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.767 Digikam::HSLSettings Class Reference

Inheritance diagram for Digikam::HSLSettings:



## Signals

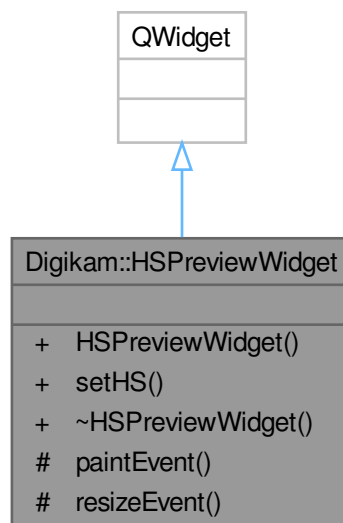
- void **signalSettingsChanged** ()

## Public Member Functions

- [HSLContainer](#) **defaultSettings** () const
- **HSLSettings** (QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const [HSLContainer](#) &settings)
- [HSLContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.768 Digikam::HSPreviewWidget Class Reference

Inheritance diagram for Digikam::HSPreviewWidget:



## Public Member Functions

- **HSPreviewWidget** (QWidget \*const parent=nullptr)
- void **setHS** (double hue, double sat)

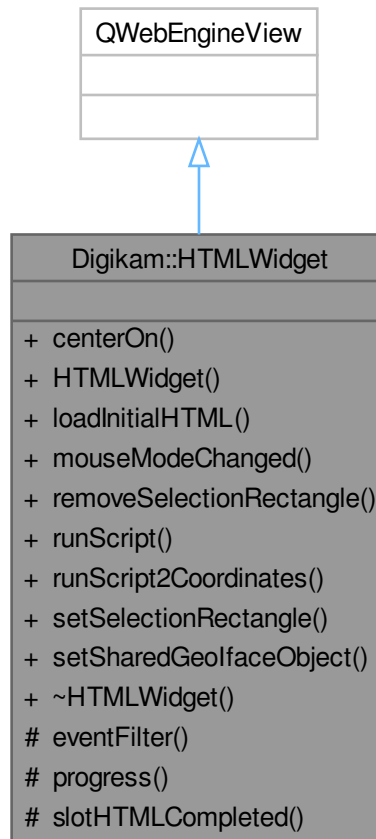
## Protected Member Functions

- void **paintEvent** (QPaintEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override



## 6.769 Digikam::HTMLWidget Class Reference

Inheritance diagram for Digikam::HTMLWidget:



### Signals

- void **selectionHasBeenMade** (const Digikam::GeoCoordinates::Pair &coordinatesRect)
- void **signalHTMLEvents** (const QStringList &events)
- void **signalJavaScriptReady** ()
- void **signalMessageEvent** (const QString &message)

### Public Member Functions

- void **centerOn** (const qreal west, const qreal north, const qreal east, const qreal south, const bool use↔ SaneZoomLevel=true)
- **HTMLWidget** (QWidget \*const parent=nullptr)
- void **loadInitialHTML** (const QString &initialHTML)
- void **mouseModeChanged** (const GeoMouseModes mouseMode)
- void **removeSelectionRectangle** ()
- QVariant **runScript** (const QString &scriptCode, bool async=true)

*Wrapper around executeScript to catch more errors.*

- bool **runScript2Coordinates** (const QString &scriptCode, [GeoCoordinates](#) \*const coordinates)  
*Execute a script which returns coordinates and parse these.*
- void **setSelectionRectangle** (const GeoCoordinates::Pair &searchCoordinates)
- void **setSharedGeofaceObject** ([GeofaceSharedData](#) \*const sharedData)

### Protected Slots

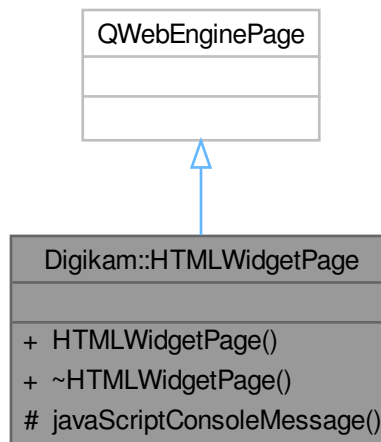
- void **progress** (int progress)
- void **slotHTMLCompleted** (bool ok)

### Protected Member Functions

- bool **eventFilter** (QObject \*, QEvent \*) override

## 6.770 Digikam::HTMLWidgetPage Class Reference

Inheritance diagram for Digikam::HTMLWidgetPage:



### Signals

- void **signalHTMLEvents** (const QStringList &events)
- void **signalMessageEvent** (const QString &message)

### Public Member Functions

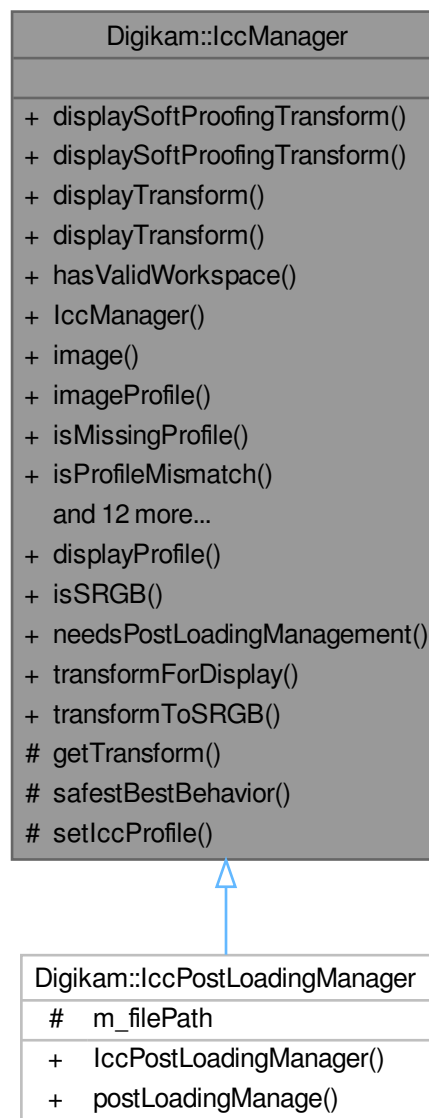
- **HTMLWidgetPage** ([HTMLWidget](#) \*const parent=nullptr)

**Protected Member Functions**

- void **javaScriptConsoleMessage** (JavaScriptConsoleMessageLevel, const QString &, int, const QString &) override

**6.771 Digikam::lccManager Class Reference**

Inheritance diagram for Digikam::lccManager:

**Public Member Functions**

- `lccTransform` **displaySoftProofingTransform** (const `lccProfile` &deviceProfile, const `lccProfile` &displayProfile)

- [IccTransform displaySoftProofingTransform](#) (const [IccProfile](#) &deviceProfile, QWidget \*const displayingWidget=nullptr)
- [IccTransform displayTransform](#) (const [IccProfile](#) &displayProfile)
- [IccTransform displayTransform](#) (QWidget \*const displayingWidget=nullptr)
- bool [hasValidWorkspace](#) () const
- [IccManager](#) (const [DImg](#) &image, const [ICCSettingsContainer](#) &settings=[IccSettings::instance\(\)](#) ->settings())
- [DImg image](#) () const
- [IccProfile imageProfile](#) ([ICCSettingsContainer::Behavior](#) behavior, const [IccProfile](#) &specifiedProfile=[IccProfile](#)())
- bool [isMissingProfile](#) () const
- bool [isProfileMismatch](#) () const
- bool [isUncalibratedColor](#) () const
- [DImgLoaderObserver](#) \* [observer](#) () const
- void [setObserver](#) ([DImgLoaderObserver](#) \*const observer)
- [ICCSettingsContainer settings](#) () const
- void [transform](#) ([ICCSettingsContainer::Behavior](#) behavior, const [IccProfile](#) &specifiedProfile=[IccProfile](#)())
- void [transformDefault](#) ()
- void [transformForDisplay](#) ()
- void [transformForDisplay](#) (const [IccProfile](#) &displayProfile)
- void [transformForDisplay](#) (QWidget \*const widget)
- void [transformForOutput](#) (const [IccProfile](#) &outputProfile)
- void [transformToSRGB](#) ()

### Static Public Member Functions

- static [IccProfile displayProfile](#) (QWidget \*const displayingWidget=nullptr)
- static bool [isSRGB](#) (const [DImg](#) &img)
- static bool [needsPostLoadingManagement](#) (const [DImg](#) &img)
- static void [transformForDisplay](#) (QImage &qimage, const [IccProfile](#) &displayProfile1=displayProfile())
- static void [transformToSRGB](#) (QImage &qimage, const [IccProfile](#) &inputProfile)

### Protected Member Functions

- void [getTransform](#) ([IccTransform](#) &trans, [ICCSettingsContainer::Behavior](#) behavior, const [IccProfile](#) &specifiedProfile)
- [ICCSettingsContainer::Behavior safestBestBehavior](#) () const
- void [setIccProfile](#) (const [IccProfile](#) &profile)

## 6.771.1 Constructor & Destructor Documentation

### 6.771.1.1 IccManager()

```
Digikam::IccManager::IccManager (
    const DImg & image,
    const ICCSettingsContainer & settings = IccSettings::instance\(\) ->settings() )
[explicit]
```

Constructs an [IccManager](#) object. The [DImg](#) will be edited. The filePath is for display only.

## 6.771.2 Member Function Documentation

### 6.771.2.1 displaySoftProofingTransform()

```
IccTransform Digikam::IccManager::displaySoftProofingTransform (
    const IccProfile & deviceProfile,
    QWidget *const displayingWidget = nullptr )
```

Returns a display transform, with soft-proofing enabled for the given device profile.

### 6.771.2.2 imageProfile()

```
IccProfile Digikam::IccManager::imageProfile (
    ICCSettingsContainer::Behavior behavior,
    const IccProfile & specifiedProfile = IccProfile() )
```

Returns the profile that will be used to interpret the image, using the given behavior

### 6.771.2.3 isSRGB()

```
bool Digikam::IccManager::isSRGB (
    const DImg & img ) [static]
```

Returns true if a call to [transformToSRGB\(\)](#) would have an effect.

### 6.771.2.4 needsPostLoadingManagement()

```
bool Digikam::IccManager::needsPostLoadingManagement (
    const DImg & img ) [static]
```

Returns true if the given image is marked as needing user interaction for further color management decision after loading. If this returns true, use [IccPostLoadingManager](#) to do this.

### 6.771.2.5 transform()

```
void Digikam::IccManager::transform (
    ICCSettingsContainer::Behavior behavior,
    const IccProfile & specifiedProfile = IccProfile() )
```

Same as above, but not using default settings but the given settings.

### 6.771.2.6 transformDefault()

```
void Digikam::IccManager::transformDefault ( )
```

Transforms the image for full editing, using default settings. If the default settings require showing a dialog, the image is marked as such but no action is taken. See [IccPostLoadingManager](#).

### 6.771.2.7 transformForDisplay() [1/2]

```
void Digikam::IccManager::transformForDisplay ( )
```

Transforms the image for display on screen. The result is not suitable for editing or storage. You can specify the widget in which the image will be displayed, or specify the display profile yourself. You can retrieve the profile with `displayProfile()` and pass it to `transformForDisplay()` later (in a thread), or you can get a transform from `displayProfile()` and apply it yourself.

### 6.771.2.8 transformForDisplay() [2/2]

```
void Digikam::IccManager::transformForDisplay (
    QImage & qimage,
    const IccProfile & displayProfile = displayProfile() ) [static]
```

Transforms the given QImage from sRGB to given display profile.

### 6.771.2.9 transformForOutput()

```
void Digikam::IccManager::transformForOutput (
    const IccProfile & outputProfile )
```

Transforms the image for output to the specified output profile

### 6.771.2.10 transformToSRGB() [1/2]

```
void Digikam::IccManager::transformToSRGB ( )
```

Transforms the image to sRGB

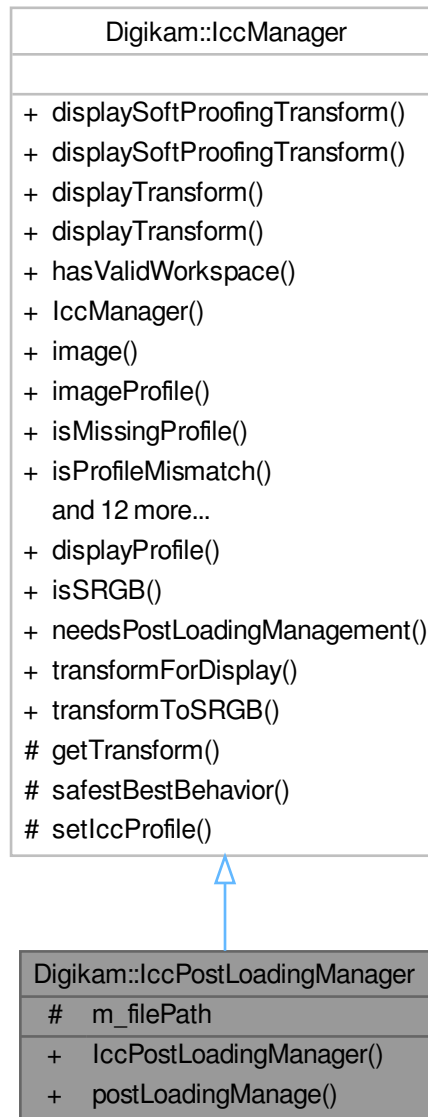
### 6.771.2.11 transformToSRGB() [2/2]

```
void Digikam::IccManager::transformToSRGB (
    QImage & qimage,
    const IccProfile & inputProfile ) [static]
```

Transforms the given QImage from the given inputProfile to sRGB.

## 6.772 Digikam::IccPostLoadingManager Class Reference

Inheritance diagram for Digikam::IccPostLoadingManager:



### Public Member Functions

- `IccPostLoadingManager` (`DImg &image`, `const QString &filePath=QString()`, `const ICCSettingsContainer &settings=IccSettings::instance() ->settings()`)
- `IccTransform postLoadingManage` (`QWidget *const parent=nullptr`)

## Public Member Functions inherited from [Digikam::IccManager](#)

- [IccTransform](#) **displaySoftProofingTransform** (const [IccProfile](#) &deviceProfile, const [IccProfile](#) &display←→Profile)
- [IccTransform](#) **displaySoftProofingTransform** (const [IccProfile](#) &deviceProfile, QWidget \*const displaying←→Widget=nullptr)
- [IccTransform](#) **displayTransform** (const [IccProfile](#) &displayProfile)
- [IccTransform](#) **displayTransform** (QWidget \*const displayingWidget=nullptr)
- bool **hasValidWorkspace** () const
- [IccManager](#) (const [DImg](#) &image, const [IccSettingsContainer](#) &settings=[IccSettings::instance\(\)](#) ->settings())
- [DImg](#) **image** () const
- [IccProfile](#) **imageProfile** ([IccSettingsContainer::Behavior](#) behavior, const [IccProfile](#) &specifiedProfile=[IccProfile](#)())
- bool **isMissingProfile** () const
- bool **isProfileMismatch** () const
- bool **isUncalibratedColor** () const
- [DImgLoaderObserver](#) \* **observer** () const
- void **setObserver** ([DImgLoaderObserver](#) \*const observer)
- [IccSettingsContainer](#) **settings** () const
- void **transform** ([IccSettingsContainer::Behavior](#) behavior, const [IccProfile](#) &specifiedProfile=[IccProfile](#)())
- void **transformDefault** ()
- void **transformForDisplay** ()
- void **transformForDisplay** (const [IccProfile](#) &displayProfile)
- void **transformForDisplay** (QWidget \*const widget)
- void **transformForOutput** (const [IccProfile](#) &outputProfile)
- void **transformToSRGB** ()

## Protected Attributes

- QString **m\_filePath**

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::IccManager](#)

- static [IccProfile](#) **displayProfile** (QWidget \*const displayingWidget=nullptr)
- static bool **isSRGB** (const [DImg](#) &img)
- static bool **needsPostLoadingManagement** (const [DImg](#) &img)
- static void **transformForDisplay** (QImage &qimage, const [IccProfile](#) &displayProfile1=displayProfile())
- static void **transformToSRGB** (QImage &qimage, const [IccProfile](#) &inputProfile)

## Protected Member Functions inherited from [Digikam::IccManager](#)

- void **getTransform** ([IccTransform](#) &trans, [IccSettingsContainer::Behavior](#) behavior, const [IccProfile](#) &specifiedProfile)
- [IccSettingsContainer::Behavior](#) **safestBestBehavior** () const
- void **setIccProfile** (const [IccProfile](#) &profile)



## 6.772.1 Constructor & Destructor Documentation

### 6.772.1.1 IccPostLoadingManager()

```
Digikam::IccPostLoadingManager::IccPostLoadingManager (
    DImg & image,
    const QString & filePath = QString(),
    const ICCSettingsContainer & settings = ICCSettings::instance()->settings() )
[explicit]
```

Constructs an `IccPostLoadingManager` object. The `DImg` will be edited. The `filePath` is for display only.

## 6.772.2 Member Function Documentation

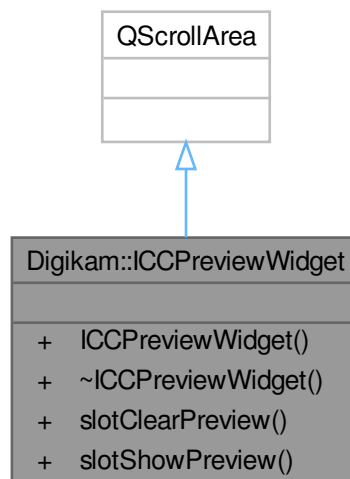
### 6.772.2.1 postLoadingManage()

```
ICCTransform Digikam::IccPostLoadingManager::postLoadingManage (
    QWidget *const parent = nullptr )
```

Carries out color management asking the user for his decision. Afterwards, `needsPostLoadingManagement` will return false.

## 6.773 Digikam::ICCPreviewWidget Class Reference

Inheritance diagram for Digikam::ICCPreviewWidget:



**Public Slots**

- void **slotClearPreview** ()
- void **slotShowPreview** (const QUrl &url)

**Public Member Functions**

- **ICCPreviewWidget** (QWidget \*const parent=nullptr)

**6.774 Digikam::IccProfile Class Reference****Public Types**

- enum **ProfileType** {  
   InvalidType , Input , Output , Display ,  
   Abstract , ColorSpace , DeviceLink , NamedColor }

**Public Member Functions**

- void **close** ()
- QByteArray **data** ()
- QString **description** ()
- QString **filePath** () const
- void \* **handle** () const
- **IccProfile** ()
- **IccProfile** (const **IccProfile** &other)
- **IccProfile** (const QByteArray &data)
- **IccProfile** (const QString &filePath)
- bool **isNull** () const
- bool **isOpen** () const
- bool **isSameProfileAs** (**IccProfile** &other)
- bool **open** ()
- **operator void** \* () const
- bool **operator!=** (const **IccProfile** &other) const
- **IccProfile** & **operator=** (const **IccProfile** &other)
- bool **operator==** (const **IccProfile** &other) const
- **ProfileType** **type** ()
- bool **writeToFile** (const QString &filePath)

**Static Public Member Functions**

- static **IccProfile** **adobeRGB** ()
- static void **considerOriginalAdobeRGB** (const QString &filePath)
- static QList< **IccProfile** > **defaultProfiles** ()
- static QStringList **defaultSearchPaths** ()
- static **IccProfile** **proPhotoRGB** ()
- static QList< **IccProfile** > **scanDirectories** (const QStringList &dirs)
- static **IccProfile** **sRGB** ()
- static **IccProfile** **wideGamutRGB** ()

**6.774.1 Member Enumeration Documentation****6.774.1.1 ProfileType**

enum **Digikam::IccProfile::ProfileType**

## Enumerator

InvalidType	Returned for a null profile or an unknown (non-standard) profile type.
Input	For an input device like a scanner or digital camera.
Output	For an output device like a printer.
Display	For a display device like a monitor.

## 6.774.2 Constructor & Destructor Documentation

### 6.774.2.1 IccProfile() [1/3]

```
Digikam::IccProfile::IccProfile ( )
```

Creates a null profile

### 6.774.2.2 IccProfile() [2/3]

```
Digikam::IccProfile::IccProfile (
    const QByteArray & data ) [explicit]
```

Creates a profile from the given data in memory

### 6.774.2.3 IccProfile() [3/3]

```
Digikam::IccProfile::IccProfile (
    const QString & filePath ) [explicit]
```

Creates a profile from the given file

## 6.774.3 Member Function Documentation

### 6.774.3.1 close()

```
void Digikam::IccProfile::close ( )
```

Close the profile, freeing resources. You can re-open. Called automatically at destruction.

### 6.774.3.2 data()

```
QByteArray Digikam::IccProfile::data ( )
```

Returns the raw profile data. Reads the data from disk if loaded from disk and not yet loaded.

### 6.774.3.3 defaultProfiles()

```
QList< IccProfile > Digikam::IccProfile::defaultProfiles ( ) [static]
```

Returns a list with the profiles above

### 6.774.3.4 defaultSearchPaths()

```
QStringList Digikam::IccProfile::defaultSearchPaths ( ) [static]
```

Returns the default search paths for ICC profiles. This does not include any user-specified settings.

### 6.774.3.5 description()

```
QString Digikam::IccProfile::description ( )
```

Reads the profile description. Opens the profile if necessary.

### 6.774.3.6 filePath()

```
QString Digikam::IccProfile::filePath ( ) const
```

Returns the filename that this profile was read from. returns a null QString() if this profile was loaded from memory.

### 6.774.3.7 handle()

```
void * Digikam::IccProfile::handle ( ) const
```

Access to the LCMS cmsHPROFILE handle.

### 6.774.3.8 isOpen()

```
bool Digikam::IccProfile::isOpen ( ) const
```

Returns if the profile is opened.

### 6.774.3.9 isSameProfileAs()

```
bool Digikam::IccProfile::isSameProfileAs (
    IccProfile & other )
```

This method compares the actual profile data bit by bit.

**6.774.3.10 open()**

```
bool Digikam::IccProfile::open ( )
```

Open this profile. Returns true if the operation succeeded or the profile is already open. Returns false if the profile is null or the operation failed. You need to open each profile after construction.

**6.774.3.11 operator==( )**

```
bool Digikam::IccProfile::operator== (
    const IccProfile & other ) const
```

Returns true if both profiles are null, if both profiles are created from the same file profile, or if the loaded profile data is identical. Note: This will not ensure that the data is loaded. Use `isSameProfile()`.

**6.774.3.12 sRGB()**

```
IccProfile Digikam::IccProfile::sRGB ( ) [static]
```

Returns the profiles available with RawEngine. You still need to call `open()` on them.

**6.774.3.13 type()**

```
IccProfile::ProfileType Digikam::IccProfile::type ( )
```

< 'nkbf', proprietary in Nikon profiles

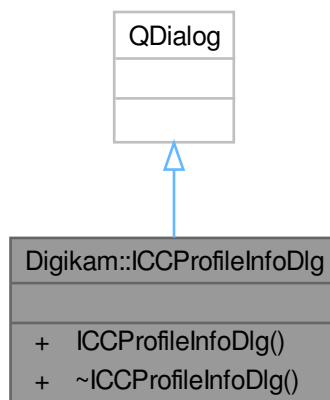
**6.774.3.14 writeToFile()**

```
bool Digikam::IccProfile::writeToFile (
    const QString & filePath )
```

Writes the profile to the given file.

**6.775 Digikam::IccProfileInfoDlg Class Reference**

Inheritance diagram for Digikam::IccProfileInfoDlg:

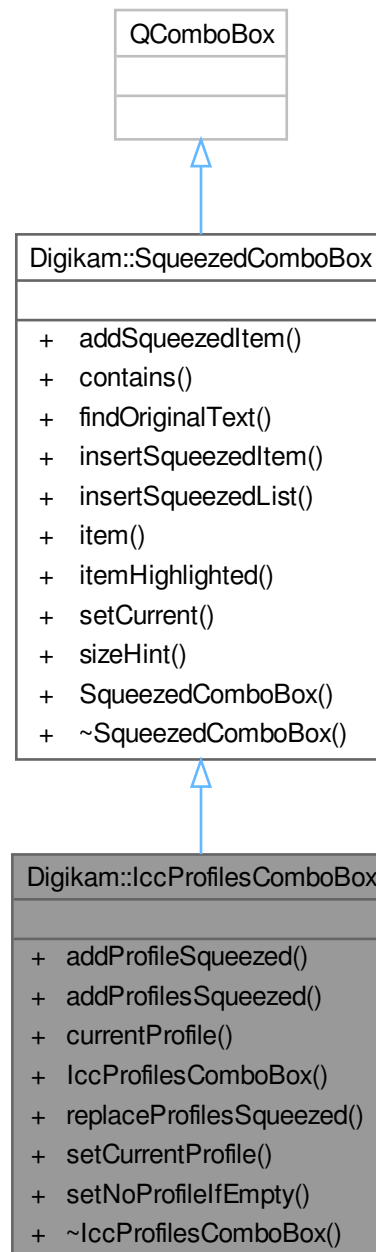


## Public Member Functions

- **ICCPProfileInfoDlg** (QWidget \*const parent, const QString &profilePath, const [IccProfile](#) &profile↔ Data=[IccProfile](#)())

## 6.776 Digikam::IccProfilesComboBox Class Reference

Inheritance diagram for Digikam::IccProfilesComboBox:



## Public Member Functions

- void [addProfileSqueezed](#) (const [IccProfile](#) &profile, const [QString](#) &description=[QString](#)())
- void [addProfilesSqueezed](#) (const [QList](#)< [IccProfile](#) > &profiles)
- [IccProfile](#) [currentProfile](#) () const
- [IccProfilesComboBox](#) ([QWidget](#) \*const parent=nullptr)
- void [replaceProfilesSqueezed](#) (const [QList](#)< [IccProfile](#) > &profiles)
- void [setCurrentProfile](#) (const [IccProfile](#) &profile)
- void [setNoProfileIfEmpty](#) (const [QString](#) &message)

## Public Member Functions inherited from [Digikam::SqueezedComboBox](#)

- void [addSqueezedItem](#) (const [QString](#) &newItem, const [QVariant](#) &userData=[QVariant](#)())
- bool [contains](#) (const [QString](#) &text) const
- int [findOriginalText](#) (const [QString](#) &text, [Qt::CaseSensitivity](#) cs=[Qt::CaseSensitive](#)) const
- void [insertSqueezedItem](#) (const [QString](#) &newItem, int index, const [QVariant](#) &userData=[QVariant](#)())
- void [insertSqueezedList](#) (const [QStringList](#) &newItems, int index)
- [QString](#) [item](#) (int index) const
- [QString](#) [itemHighlighted](#) () const
- void [setCurrent](#) (const [QString](#) &itemText)
- [QSize](#) [sizeHint](#) () const override
- [SqueezedComboBox](#) ([QWidget](#) \*const parent=nullptr, const char \*name=nullptr)
- [~SqueezedComboBox](#) () override

## Additional Inherited Members

## Signals inherited from [Digikam::SqueezedComboBox](#)

- void [signalItemActivated](#) (const [QString](#) &)

## 6.776.1 Constructor & Destructor Documentation

### 6.776.1.1 IccProfilesComboBox()

```
Digikam::IccProfilesComboBox::IccProfilesComboBox (
    QWidget *const parent = nullptr ) [explicit]
```

#### Note

Use the signal [currentIndexChanged\(int\)](#) for change notification

## 6.776.2 Member Function Documentation

### 6.776.2.1 addProfileSqueezed()

```
void Digikam::IccProfilesComboBox::addProfileSqueezed (
    const IccProfile & profile,
    const QString & description = QString() )
```

Add the given profile with the given description, or, if null, a standard description. Does not test for duplicity, does not sort into existing profiles.

### 6.776.2.2 addProfilesSqueezed()

```
void Digikam::IccProfilesComboBox::addProfilesSqueezed (
    const QList< IccProfile > & profiles )
```

Checks the given profiles for validity, creates a suitable description (ICC profile description, file path), removes duplicates by file path, sorts them and adds them in sorted order.

### 6.776.2.3 currentProfile()

```
IccProfile Digikam::IccProfilesComboBox::currentProfile ( ) const
```

Retrieves the current profile, or a null profile if none is selected.

### 6.776.2.4 replaceProfilesSqueezed()

```
void Digikam::IccProfilesComboBox::replaceProfilesSqueezed (
    const QList< IccProfile > & profiles )
```

Clears, does the same as addProfilesSqueezed, and restores the current entry if possible.

### 6.776.2.5 setCurrentProfile()

```
void Digikam::IccProfilesComboBox::setCurrentProfile (
    const IccProfile & profile )
```

Sets the current profile. If profile is not in the list, sets no current item (-1)

### 6.776.2.6 setNoProfileIfEmpty()

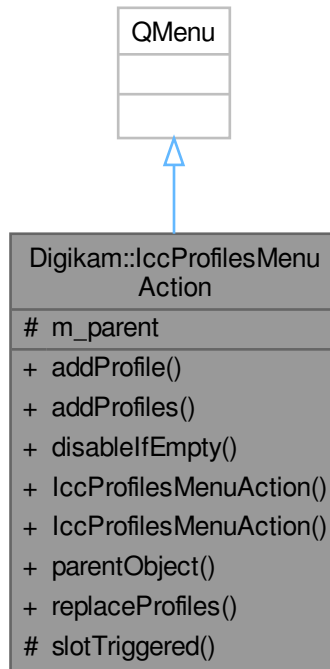
```
void Digikam::IccProfilesComboBox::setNoProfileIfEmpty (
    const QString & message )
```

Sets a message the is displayed in the combo box and disables the combo box, if the combo box is currently empty



## 6.777 Digikam::IccProfilesMenuAction Class Reference

Inheritance diagram for Digikam::IccProfilesMenuAction:



### Signals

- void **triggered** (const [IccProfile](#) &profile)

### Public Member Functions

- void [addProfile](#) (const [IccProfile](#) &profile, const QString &description=QString())
- void [addProfiles](#) (const QList< [IccProfile](#) > &profile)
- void [disableIfEmpty](#) ()
- **IccProfilesMenuAction** (const QIcon &icon, const QString &text, QObject \*const parent)
- **IccProfilesMenuAction** (const QString &text, QObject \*const parent)
- QObject \* [parentObject](#) () const
- void [replaceProfiles](#) (const QList< [IccProfile](#) > &profile)

### Protected Slots

- void **slotTriggered** (QObject \*)

## Protected Attributes

- `QObject * m_parent = nullptr`

## 6.777.1 Member Function Documentation

### 6.777.1.1 addProfile()

```
void Digikam::IccProfilesMenuAction::addProfile (
    const IccProfile & profile,
    const QString & description = QString() )
```

Add the given profile with the given description, or, if null, a standard description. Does not test for duplicity, does not sort into existing profiles.

### 6.777.1.2 addProfiles()

```
void Digikam::IccProfilesMenuAction::addProfiles (
    const QList< IccProfile > & profile )
```

Checks the given profiles for validity, creates a suitable description (ICC profile description, file path), removes duplicates (in newly added list) by file path, sorts them and adds them in sorted order.

### 6.777.1.3 disableIfEmpty()

```
void Digikam::IccProfilesMenuAction::disableIfEmpty ( )
```

Disables if the menu is currently empty.

### 6.777.1.4 parentObject()

```
QObject * Digikam::IccProfilesMenuAction::parentObject ( ) const
```

Return the parent QObject.

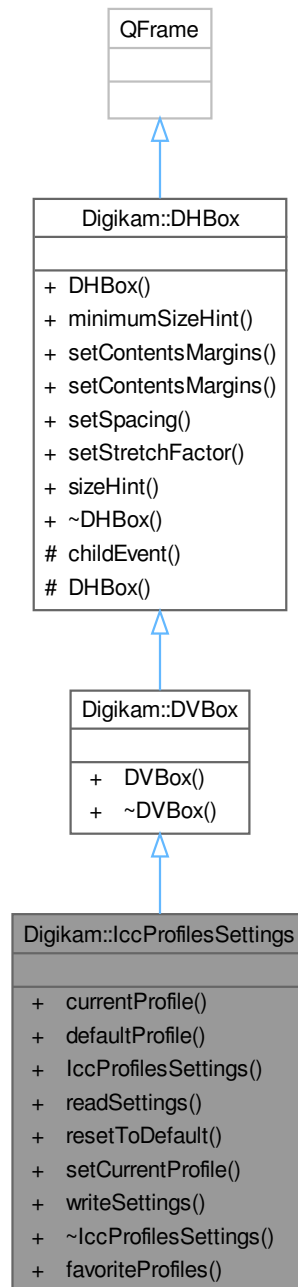
### 6.777.1.5 replaceProfiles()

```
void Digikam::IccProfilesMenuAction::replaceProfiles (
    const QList< IccProfile > & profile )
```

Equivalent to calling `clear()` and `addProfiles()`.

## 6.778 Digikam::IccProfilesSettings Class Reference

Inheritance diagram for Digikam::IccProfilesSettings:



### Signals

- void **signalSettingsChanged** ()

### Public Member Functions

- [IccProfile](#) **currentProfile** () const
- [IccProfile](#) **defaultProfile** () const
- **IccProfilesSettings** (QWidget \*const parent=nullptr)
- void **readSettings** (KConfigGroup &group)
- void **resetToDefault** ()
- void **setCurrentProfile** (const [IccProfile](#) &prof)
- void **writeSettings** (KConfigGroup &group)

### Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

### Static Public Member Functions

- static QStringList **favoriteProfiles** (KConfigGroup &group)

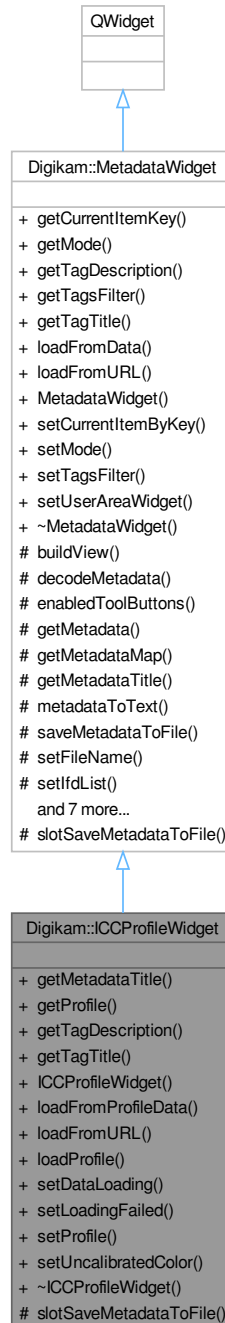
### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.779 Digikam::ICCPProfileWidget Class Reference

Inheritance diagram for Digikam::ICCPProfileWidget:



### Public Member Functions

- QString [getMetadataTitle](#) () const override
- [IccProfile](#) [getProfile](#) () const

- QString [getTagDescription](#) (const QString &key) override
- QString [getTagTitle](#) (const QString &key) override
- **ICCPProfileWidget** (QWidget \*const parent, int w=256, int h=256)
- bool **loadFromProfileData** (const QString &fileName, const QByteArray &data)
- bool [loadFromURL](#) (const QUrl &url) override
- bool **loadProfile** (const QString &fileName, const [IccProfile](#) &data)
- void **setDataLoading** ()
- void **setLoadingFailed** ()
- bool **setProfile** (const [IccProfile](#) &profile)
- void **setUncalibratedColor** ()

### Public Member Functions inherited from [Digikam::MetadataWidget](#)

- QString **getCurrentItemKey** () const
- int **getMode** () const
- QStringList **getTagsFilter** () const
- virtual bool **loadFromData** (const QString &fileName, const [DMetadata](#) &data=[DMetadata](#)())
- **MetadataWidget** (QWidget \*const parent, const QString &name=QString())
- void **setCurrentItemByKey** (const QString &itemKey)
- void **setMode** (int mode)
- void **setTagsFilter** (const QStringList &list)
- void **setUserAreaWidget** (QWidget \*const w)

### Protected Slots

- void **slotSaveMetadataToFile** () override

### Protected Slots inherited from [Digikam::MetadataWidget](#)

- virtual void **slotSaveMetadataToFile** ()=0

### Additional Inherited Members

### Public Types inherited from [Digikam::MetadataWidget](#)

- enum **TagFilters** { **NONE** = 0 , **PHOTO** , **CUSTOM** }

### Signals inherited from [Digikam::MetadataWidget](#)

- void **signalSetupMetadataFilters** ()

## Protected Member Functions inherited from [Digikam::MetadataWidget](#)

- void **enabledToolButtons** (bool)
- [DMetadata](#) \* **getMetadata** () const
- const [DMetadata::MetaDataMap](#) & **getMetadataMap** ()
- QString **metadataToText** () const
- QUrl **saveMetadataToFile** (const QString &caption, const QString &fileFilter)
- void **setFileName** (const QString &fileName)
- void **setIfdList** (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &keysFilter, const QStringList &tagsFilter)
- void **setIfdList** (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &tagsFilter=QStringList())
- bool **setMetadata** (const [DMetadata](#) &data=[DMetadata](#)())
- virtual void **setMetadataEmpty** ()
- void **setMetadataMap** (const [DMetadata::MetaDataMap](#) &data=[DMetadata::MetaDataMap](#)())
- void **setup** ()
- bool **storeMetadataToFile** (const QUrl &url, const QByteArray &metaData)
- [MetadataListView](#) \* **view** () const

### 6.779.1 Member Function Documentation

#### 6.779.1.1 [getMetadataTitle\(\)](#)

```
QString Digikam::ICCPProfileWidget::getMetadataTitle ( ) const [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

#### 6.779.1.2 [getTagDescription\(\)](#)

```
QString Digikam::ICCPProfileWidget::getTagDescription (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

#### 6.779.1.3 [getTagTitle\(\)](#)

```
QString Digikam::ICCPProfileWidget::getTagTitle (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

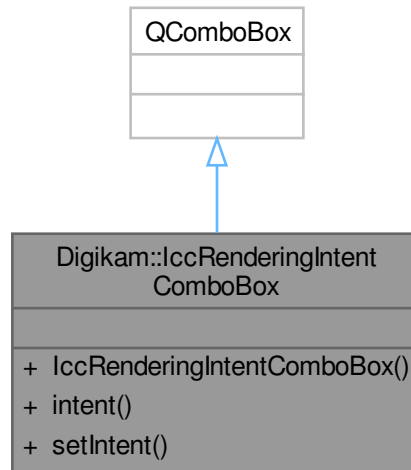
#### 6.779.1.4 [loadFromURL\(\)](#)

```
bool Digikam::ICCPProfileWidget::loadFromURL (
    const QUrl & url ) [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

## 6.780 Digikam::IccRenderingIntentComboBox Class Reference

Inheritance diagram for Digikam::IccRenderingIntentComboBox:



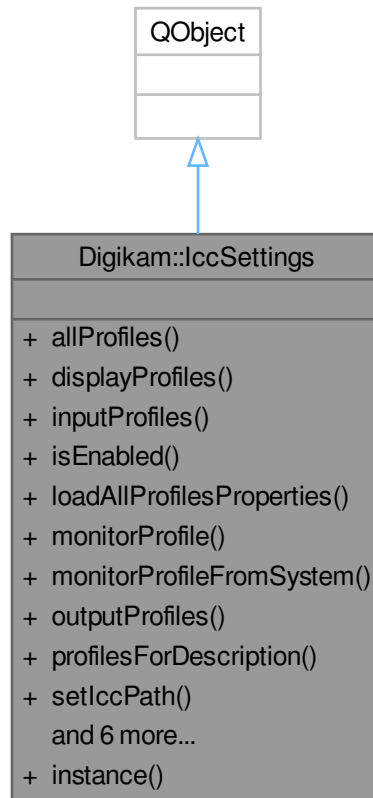
### Public Member Functions

- **IccRenderingIntentComboBox** (QWidget \*const parent=nullptr)
- int **intent** () const
- void **setIntent** (int intent)



## 6.781 Digikam::lccSettings Class Reference

Inheritance diagram for Digikam::lccSettings:



### Classes

- class [Private](#)

### Signals

- void **signalICCSettingsChanged** (const [ICCSettingsContainer](#) &current, const [ICCSettingsContainer](#) &previous)
- void **signalSettingsChanged** ()

### Public Member Functions

- `QList< lccProfile > allProfiles ()`
- `QList< lccProfile > displayProfiles ()`
- `QList< lccProfile > inputProfiles ()`
- `bool isEnabled () const`

- void [loadAllProfilesProperties](#) ()
- [IccProfile](#) [monitorProfile](#) (QWidget \*const widget=nullptr)
- bool [monitorProfileFromSystem](#) () const
- QList< [IccProfile](#) > [outputProfiles](#) ()
- QList< [IccProfile](#) > [profilesForDescription](#) (const QString &description)
- void [setIccPath](#) (const QString &path)
- void [setSettings](#) (const [ICCSettingsContainer](#) &settings)
- [ICCSettingsContainer](#) [settings](#) ()
- void [setUseManagedPreviews](#) (bool [useManagedPreviews](#))
- void [setUseManagedView](#) (bool useManagedView)
- bool [useManagedPreviews](#) () const
- QList< [IccProfile](#) > [workspaceProfiles](#) ()

### Static Public Member Functions

- static [IccSettings](#) \* [instance](#) ()

### Friends

- class [IccSettingsCreator](#)
- class [Private](#)

## 6.781.1 Member Function Documentation

### 6.781.1.1 [displayProfiles\(\)](#)

```
QList< IccProfile > Digikam::IccSettings::displayProfiles ( )
```

Get available profiles suitable as monitor/display profile

### 6.781.1.2 [inputProfiles\(\)](#)

```
QList< IccProfile > Digikam::IccSettings::inputProfiles ( )
```

Get available profiles suitable as input profile

### 6.781.1.3 [instance\(\)](#)

```
IccSettings * Digikam::IccSettings::instance ( ) [static]
```

Global container for ICC settings. All accessor methods are thread-safe.

### 6.781.1.4 [isEnabled\(\)](#)

```
bool Digikam::IccSettings::isEnabled ( ) const
```

Returns if color management is enabled.

### 6.781.1.5 loadAllProfilesProperties()

```
void Digikam::IccSettings::loadAllProfilesProperties ( )
```

[IccProfile](#) caches some of its properties (description, type) when it was read once. Subsequently, to read these values no opening is needed. This ensures that all profiles have these values read. May imply scanning and opening all profiles.

### 6.781.1.6 monitorProfile()

```
IccProfile Digikam::IccSettings::monitorProfile (
    QWidget *const widget = nullptr )
```

Returns the monitor profile (for color managed view). If there are multiple screens, a system-wide settings specifies the monitor profile, and the widget parameter is passed, the returned profile is for the widget's screen. If no settings is specified, the default sRGB profile is returned.

### 6.781.1.7 monitorProfileFromSystem()

```
bool Digikam::IccSettings::monitorProfileFromSystem ( ) const
```

Returns if the monitor profile (as returned by [monitorProfile\(\)](#)) is set system-wide, so that the `monitorProfile` field of the current settings need not be set and will not be used by [monitorProfile\(\)](#).

### 6.781.1.8 outputProfiles()

```
QList< IccProfile > Digikam::IccSettings::outputProfiles ( )
```

Get available profiles suitable as proof/output profiles

### 6.781.1.9 profilesForDescription()

```
QList< IccProfile > Digikam::IccSettings::profilesForDescription (
    const QString & description )
```

Returns a list of profiles with the given description()

### 6.781.1.10 setSettings()

```
void Digikam::IccSettings::setSettings (
    const ICCSettingsContainer & settings )
```

Sets the current ICC settings and writes them to config.

### 6.781.1.11 settings()

```
ICCSettingsContainer Digikam::IccSettings::settings ( )
```

Returns the current ICC settings.

### 6.781.1.12 `setUseManagedView()`

```
void Digikam::IccSettings::setUseManagedView (
    bool useManagedView )
```

Set single parts of the settings

### 6.781.1.13 `useManagedPreviews()`

```
bool Digikam::IccSettings::useManagedPreviews ( ) const
```

Returns if color management for previews is enabled.

### 6.781.1.14 `workspaceProfiles()`

```
QList< IccProfile > Digikam::IccSettings::workspaceProfiles ( )
```

Get available profiles suitable as workspace profile

## 6.782 Digikam::IccSettings::Private Class Reference

### Public Member Functions

- [IccProfile](#) `profileFromWindowSystem` (QWidget \*const widget)
- [IccSettingsContainer](#) `readFromConfig` () const
- QList< [IccProfile](#) > `scanDirectories` (const QStringList &dirs)
- void `scanDirectory` (const QString &path, const QStringList &filter, QList< [IccProfile](#) > \*const profiles)
- void `writeManagedPreviewsToConfig` () const
- void `writeManagedViewToConfig` () const
- void `writeToConfig` () const

### Public Attributes

- const QString `configGroup`
- QMutex `mutex`
- QList< [IccProfile](#) > `profiles`
- QHash< int, [IccProfile](#) > `screenProfiles`
- [IccSettingsContainer](#) `settings`

## 6.783 Digikam::IccSettingsContainer Class Reference

### Public Types

- enum [BehaviorEnum](#) {
   
[InvalidBehavior](#) = 0 , [UseEmbeddedProfile](#) = 1 << 0 , [UseSRGB](#) = 1 << 1 , [UseWorkspace](#) = 1 << 2 ,
   
[UseDefaultInputProfile](#) = 1 << 3 , [UseSpecifiedProfile](#) = 1 << 4 , [AutomaticColors](#) = 1 << 5 , [DoNotInterpret](#) = 1 << 6 ,
   
[KeepProfile](#) = 1 << 10 , [ConvertToWorkspace](#) = 1 << 11 , [LeaveFileUntagged](#) = 1 << 18 , [AskUser](#) = 1
   
<< 20 ,
   
[SafestBestAction](#) = 1 << 21 , [PreserveEmbeddedProfile](#) = UseEmbeddedProfile | KeepProfile ,
   
[EmbeddedToWorkspace](#) = UseEmbeddedProfile | ConvertToWorkspace , [SRGBToWorkspace](#) = UseSRGB | ConvertToWorkspace ,
   
[AutoToWorkspace](#) = AutomaticColors | ConvertToWorkspace , [InputToWorkspace](#) = UseDefaultInputProfile | ConvertToWorkspace ,
   
[SpecifiedToWorkspace](#) = UseSpecifiedProfile | ConvertToWorkspace ,
   
[NoColorManagement](#) = DoNotInterpret | LeaveFileUntagged }

## Public Member Functions

- void **readFromConfig** (KConfigGroup &group)
- void **writeManagedPreviewsToConfig** (KConfigGroup &group) const
- void **writeManagedViewToConfig** (KConfigGroup &group) const
- void **writeToConfig** (KConfigGroup &group) const

## Public Attributes

- QString **defaultInputProfile**
- Behavior **defaultMismatchBehavior** = EmbeddedToWorkspace
- Behavior **defaultMissingProfileBehavior** = SRGBToWorkspace
- QString **defaultProofProfile**
- Behavior **defaultUncalibratedBehavior** = AutoToWorkspace
- bool **doGamutCheck** = false
- bool **enableCM** = true
- QColor **gamutCheckMaskColor** = QColor(126, 255, 255)
- QString **iccFolder**
- Behavior **lastMismatchBehavior** = EmbeddedToWorkspace
- Behavior **lastMissingProfileBehavior** = SRGBToWorkspace
- QString **lastSpecifiedAssignProfile**
- QString **lastSpecifiedInputProfile**
- Behavior **lastUncalibratedBehavior** = AutoToWorkspace
- QString **monitorProfile**
- int **proofingRenderingIntent** = IccTransform::AbsoluteColorimetric  
*Settings specific for soft proofing.*
- int **renderingIntent** = IccTransform::Perceptual
- bool **useBPC** = true
- bool **useManagedPreviews** = true
- bool **useManagedView** = true
- QString **workspaceProfile**

## 6.783.1 Member Enumeration Documentation

### 6.783.1.1 BehaviorEnum

enum Digikam::ICCSettingsContainer::BehaviorEnum

#### Enumerator

InvalidBehavior	Note: Values are stored in config - keep them constant.
UseEmbeddedProfile	Interpretation of the image data.
KeepProfile	Transformation / target profile.
LeaveFileUntagged	Special flags and values.
PreserveEmbeddedProfile	ready combinations for convenience

## 6.784 Digikam::IccTransform Class Reference

### Public Types

- enum **RenderingIntent** { **Perceptual** = 0 , **RelativeColorimetric** = 1 , **Saturation** = 2 , **AbsoluteColorimetric** = 3 }

### Public Member Functions

- bool **apply** (DImg &image, DImgLoaderObserver \*const observer=nullptr)
- bool **apply** (QImage &qimage)
- QColor **checkGamutMaskColor** () const
- void **close** ()
- IccProfile **effectiveInputProfile** () const
- IccProfile **embeddedProfile** () const
- IccTransform (const IccTransform &other)
- IccProfile **inputProfile** () const
- RenderingIntent **intent** () const
- bool **isCheckingGamut** () const
- bool **isUsingBlackPointCompensation** () const
- IccTransform & **operator=** (const IccTransform &other)
- IccProfile **outputProfile** () const
- RenderingIntent **proofIntent** () const
- IccProfile **proofProfile** () const
- void **setCheckGamut** (bool checkGamut)
- void **setCheckGamutMaskColor** (const QColor &color)
- void **setDoNotEmbedOutputProfile** (bool doNotEmbed)
- void **setEmbeddedProfile** (const DImg &image)
- void **setInputProfile** (const IccProfile &profile)
- void **setIntent** (int intent)
- void **setIntent** (RenderingIntent intent)
- void **setOutputProfile** (const IccProfile &profile)
- void **setProofIntent** (int intent)
- void **setProofIntent** (RenderingIntent intent)
- void **setProofProfile** (const IccProfile &profile)
- void **setUseBlackPointCompensation** (bool useBPC)
- bool **willHaveEffect** ()

### Static Public Member Functions

- static void **init** ()

## 6.784.1 Member Function Documentation

### 6.784.1.1 apply() [1/2]

```
bool Digikam::IccTransform::apply (
    DImg & image,
    DImgLoaderObserver *const observer = nullptr )
```

Apply this transform with the set profiles and options to the image. Optionally pass an observer to get progress information.

### 6.784.1.2 apply() [2/2]

```
bool Digikam::IccTransform::apply (
    QImage & qimage )
```

Apply this transform to the QImage. This has only basic functionality.

### 6.784.1.3 close()

```
void Digikam::IccTransform::close ( )
```

Closes the transform, not the profiles. Called at destruction.

### 6.784.1.4 effectiveInputProfile()

```
IccProfile Digikam::IccTransform::effectiveInputProfile ( ) const
```

Returns the embedded profile; if none is set, the input profile; if none is set, sRGB.

### 6.784.1.5 embeddedProfile()

```
IccProfile Digikam::IccTransform::embeddedProfile ( ) const
```

Returns the contained profiles

### 6.784.1.6 init()

```
void Digikam::IccTransform::init ( ) [static]
```

Initialize LittleCMS library

### 6.784.1.7 setDoNotEmbedOutputProfile()

```
void Digikam::IccTransform::setDoNotEmbedOutputProfile (
    bool doNotEmbed )
```

Call this with 'true' if you do not want the output profile to be set as embedded profile after [apply\(\)](#) did a transformation. Default is to set the output profile as embedded profile (false).

### 6.784.1.8 setEmbeddedProfile()

```
void Digikam::IccTransform::setEmbeddedProfile (
    const QImage & image )
```

Sets the input profiles of this transform. You can call both [setEmbeddedProfile](#) and [setInputProfile](#). If the image contains an embedded profile this profile is used and takes precedence over the set input profile, which is used without an embedded profile. If none is set, sRGB is used.

**6.784.1.9 setIntent()**

```
void Digikam::IccTransform::setIntent (
    RenderingIntent intent )
```

Set options

**6.784.1.10 setOutputProfile()**

```
void Digikam::IccTransform::setOutputProfile (
    const IccProfile & profile )
```

Sets the output transform

**6.784.1.11 setProofProfile()**

```
void Digikam::IccTransform::setProofProfile (
    const IccProfile & profile )
```

Makes this transform a proofing transform, if profile is not null

**6.784.1.12 willHaveEffect()**

```
bool Digikam::IccTransform::willHaveEffect ( )
```

Returns if this transformation will have an effect, i.e. if effective input profile and output profile are different.



## 6.785 Digikam::lccTransformFilter Class Reference

Inheritance diagram for Digikam::lccTransformFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **IccTransformFilter** ([DImg](#) \*const orgImage, QObject \*const parent, const [IccTransform](#) &transform)
- **IccTransformFilter** (QObject \*const parent=nullptr)
- bool [parametersSuccessfullyRead](#) () const override
- void [readParameters](#) (const [FilterAction](#) &action) override
- QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int [stop](#), int start=0) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Public Member Functions inherited from [Digikam::DImgLoaderObserver](#)

- virtual bool [continueQuery](#) ()
- virtual float [granularity](#) ()

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static QString [DisplayableName](#) ()
- static QString [FilterIdentifier](#) ()
- static QList< int > [SupportedVersions](#) ()

### Protected Member Functions

- void [filterImage](#) () override
- void [progressInfo](#) (float [progress](#)) override

### Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cleanupFilter](#) ()
- [DImgThreadedFilter](#) ([DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void [initFilter](#) ()
- void [initMaster](#) ()
- void [initSlave](#) ([DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- virtual int [modulateProgress](#) (int [progress](#))
- void [postProgress](#) (int [progress](#))
- virtual void [prepareDestImage](#) ()
- void [run](#) () override
- void [setSlave](#) ([DImgThreadedFilter](#) \*const slave)

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

### Public Slots inherited from [Digikam::DynamicThread](#)

- void [start](#) ()
- void [stop](#) ()
- void [wait](#) ()

### Signals inherited from [Digikam::DImgThreadedFilter](#)

- void [finished](#) (bool success)
- void [progress](#) (int progress)
- void [started](#) ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- [DImg m\\_destImage](#)
- [DImgThreadedFilter \\* m\\_master](#) = nullptr
- [QString m\\_name](#)
- [DImg m\\_orgImage](#)
- int [m\\_progressBegin](#) = 0
- int [m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int [m\\_progressSpan](#) = 0
- [DImgThreadedFilter \\* m\\_slave](#) = nullptr
- int [m\\_version](#) = 1
- bool [m\\_wasCancelled](#) = false

## 6.785.1 Member Function Documentation

### 6.785.1.1 filterAction()

```
FilterAction Digikam::IccTransformFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.785.1.2 filterIdentifier()

```
QString Digikam::IccTransformFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.785.1.3 filterImage()

```
void Digikam::IccTransformFilter::filterImage ( ) [override], [protected], [virtual]
```

Main image filter method. Override in subclass.

Implements [Digikam::DImgThreadedFilter](#).

#### 6.785.1.4 parametersSuccessfullyRead()

```
bool Digikam::IccTransformFilter::parametersSuccessfullyRead ( ) const [override], [virtual]
```

Optional: error handling for readParameters. When readParameters() has been called, this method will return true if the call was successful, and false if not. If returning false, readParametersError() will give an error message. The default implementation always returns success. You only need to reimplement when a filter is likely to fail in a different environment, e.g. depending on availability of installed files. These methods have an undefined return value if readParameters() was not called previously.

Reimplemented from [Digikam::DImgThreadedFilter](#).

#### 6.785.1.5 progressInfo()

```
void Digikam::IccTransformFilter::progressInfo (
    float progress ) [override], [protected], [virtual]
```

Posts progress information about image IO

Reimplemented from [Digikam::DImgLoaderObserver](#).

#### 6.785.1.6 readParameters()

```
void Digikam::IccTransformFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

#### 6.785.1.7 readParametersError()

```
QString Digikam::IccTransformFilter::readParametersError (
    const FilterAction & actionThatFailed ) const [override], [virtual]
```

Reimplemented from [Digikam::DImgThreadedFilter](#).

## 6.786 Digikam::Identity Class Reference

### Public Member Functions

- QString [attribute](#) (const QString &att) const
- QMap< QString, QString > [attributesMap](#) () const
- int [id](#) () const
- [Identity](#) ()
- [Identity](#) (const [Identity](#) &other)
- bool [isNull](#) () const
- [Identity](#) & [operator=](#) (const [Identity](#) &other)
- bool [operator==](#) (const [Identity](#) &other) const
- void [setAttribute](#) (const QString &att, const QString &val)
- void [setAttributesMap](#) (const QMap< QString, QString > &attributes)
- void [setId](#) (int [id](#))

## 6.786.1 Constructor & Destructor Documentation

### 6.786.1.1 Identity()

```
Digikam::Identity::Identity ( )
```

Wraps a face recognition [Identity](#). An identity refers to a natural person. There is an internal id which is used the [FacesEngine](#) storage, and a number of attributes which map the identity to the outside. Prespecified attributes↵: "fullName" The full name as on the ID card, e.g. "Peter Brown" "name" The person's name without further specification, e.g. "Peter" or "Dad" "uuid" A UUID that is assigned to each new identity at creation.

For fullName and name, multiple values are allowed.

Attributes can be used to map an identity to other fields and services where natural persons play a role.

## 6.786.2 Member Function Documentation

### 6.786.2.1 attribute()

```
QString Digikam::Identity::attribute (
    const QString & att ) const
```

Attribute value accessor.

### 6.786.2.2 attributesMap()

```
QMultiMap< QString, QString > Digikam::Identity::attributesMap ( ) const
```

Attributes map accessor.

### 6.786.2.3 id()

```
int Digikam::Identity::id ( ) const
```

Id value accessor.

## 6.787 Digikam::IdentityProvider Class Reference

### Public Member Functions

- [Identity addIdentity](#) (const QMultiMap< QString, QString > &attributes)
- [Identity addIdentityDebug](#) (const QMultiMap< QString, QString > &attributes)
- int [addTraining](#) (const [Identity](#) &identity, const QString &hash, const cv::Mat &feature)
- QList< [Identity](#) > [allIdentities](#) ( ) const
- void [clearAllTraining](#) ( )
- bool [clearTraining](#) (const QString &hash)
- void [deleteIdentities](#) (QList< [Identity](#) > identitiesToBeDeleted)
- void [deleteIdentity](#) (const [Identity](#) &identityToBeDeleted)
- [Identity findIdentity](#) (const QMultiMap< QString, QString > &attributes) const
- [Identity findIdentity](#) (const QString &attribute, const QString &value) const
- [Identity identity](#) (int id) const
- bool [integrityCheck](#) ( )
- bool [isValidId](#) (int label) const
- void [vacuum](#) ( )

## Static Public Member Functions

- static [IdentityProvider](#) \* **instance** ()

## Protected Member Functions

- bool **addIdentityFace** (const [Identity](#) &identity, QString &hash, cv::Mat embedding)
- bool **deleteIdentityFace** (const [Identity](#) &identity, QString &hash)
- cv::Ptr< cv::ml::TrainData > **getTrainingData** () const
- bool **initialize** ()

## Friends

- class **FaceClassifier**
- class **Identity**
- class **IdentityProviderCreator**

## 6.787.1 Member Function Documentation

### 6.787.1.1 addIdentity()

```
Identity Digikam::IdentityProvider::addIdentity (
    const QMap< QString, QString > & attributes )
```

Adds a new identity with the specified attributes. Please note that a UUID is automatically generated.

### 6.787.1.2 addIdentityDebug()

```
Identity Digikam::IdentityProvider::addIdentityDebug (
    const QMap< QString, QString > & attributes )
```

This is the debug version of addIdentity, so the identity is only added to identityCache, but not into the recognition database.

### 6.787.1.3 addTraining()

```
int Digikam::IdentityProvider::addTraining (
    const Identity & identity,
    const QString & hash,
    const cv::Mat & feature )
```

add the face features and hash to the recognition DB returns the ID of the new row

### 6.787.1.4 allIdentities()

```
QList< Identity > Digikam::IdentityProvider::allIdentities ( ) const
```

Returns all identities known to the database

### 6.787.1.5 clearAllTraining()

```
void Digikam::IdentityProvider::clearAllTraining ( )
```

clears all identities and face training from the recognition DB

### 6.787.1.6 clearTraining()

```
bool Digikam::IdentityProvider::clearTraining (
    const QString & hash )
```

Deletes the training image for the given hash, leaving the identity as such in the database.

### 6.787.1.7 deleteIdentities()

```
void Digikam::IdentityProvider::deleteIdentities (
    QList< Identity > identitiesToBeDeleted )
```

Deletes a list of identities from the database.

### 6.787.1.8 deleteIdentity()

```
void Digikam::IdentityProvider::deleteIdentity (
    const Identity & identityToBeDeleted )
```

Deletes an identity from the database.

### 6.787.1.9 findIdentity() [1/2]

```
Identity Digikam::IdentityProvider::findIdentity (
    const QMap< QString, QString > & attributes ) const
```

Finds the identity matching the given attributes. Attributes are first checked with knowledge of their meaning. Secondly, all unknown attributes are used. Returns a null [Identity](#) if no match is possible or the map is empty.

### 6.787.1.10 findIdentity() [2/2]

```
Identity Digikam::IdentityProvider::findIdentity (
    const QString & attribute,
    const QString & value ) const
```

Finds the first identity with matching attribute - value. Returns a null identity if no match is found or attribute is empty.

### 6.787.1.11 getTrainingData()

```
cv::Ptr< cv::ml::TrainData > Digikam::IdentityProvider::getTrainingData ( ) const [protected]
```

Deletes a list of identities from the database.



### 6.787.1.12 integrityCheck()

```
bool Digikam::IdentityProvider::integrityCheck ( )
```

Checks the integrity and returns true if everything is fine.

### 6.787.1.13 isValidId()

```
bool Digikam::IdentityProvider::isValidId (
    int label ) const
```

checks if the id exists in the recognition DB

### 6.787.1.14 vacuum()

```
void Digikam::IdentityProvider::vacuum ( )
```

Shrinks the database.

## 6.788 Digikam::ImageChangeset Class Reference

### Public Member Functions

- [DatabaseFields::Set changes](#) () const
- bool [containsImage](#) (qulonglong id) const
- [QList< qulonglong > ids](#) () const
- [ImageChangeset](#) ()=default
- [ImageChangeset](#) (const [QList< qulonglong > &ids](#), const [DatabaseFields::Set &changes](#))
- [ImageChangeset](#) (qulonglong id, const [DatabaseFields::Set &changes](#))

### 6.788.1 Constructor & Destructor Documentation

#### 6.788.1.1 ImageChangeset()

```
Digikam::ImageChangeset::ImageChangeset ( ) [default]
```

An [ImageChangeset](#) covers adding or changing any properties of an image. It is described by a list of affected image ids, and a set of affected database fields. There is no guarantee that information in the database has actually been changed.

## 6.789 Digikam::ImageCommonContainer Class Reference

### Public Attributes

- int **colorDepth** = 0  
*bits per channel, 8/16*
- QString **colorModel**
- QDateTime **creationDate**
- QDateTime **digitizationDate**
- QDateTime **fileModificationDate**
- QString **fileName**
- qint64 **fileSize** = 0
- QString **format**
- int **height** = 0
- QString **orientation**
- int **rating** = -1
- int **width** = 0

## 6.790 Digikam::ImageCurves Class Reference

### Public Types

- typedef double **CRMatrix**[4][4]
- enum **CurveType** { **CURVE\_SMOOTH** = 0 , **CURVE\_FREE** }

### Public Member Functions

- QByteArray **channelToBinary** (int channel) const
- void **curvesCalculateAllCurves** ()
- void **curvesCalculateCurve** (int channel)
- void **curvesChannelReset** (int channel)
- float **curvesLutFunc** (int n\_channels, int channel, float value)
- void **curvesLutProcess** (uchar \*const srcPR, uchar \*const destPR, int w, int h)
- void **curvesLutSetup** (int nchannels)
- void **curvesReset** ()

*Methods to manipulate the curves data.*

- void **fillFromOtherCurves** (const [ImageCurves](#) \*const otherCurves)
- [CurvesContainer](#) **getContainer** () const
- [CurvesContainer](#) **getContainer** (int channel) const
- QPoint **getCurvePoint** (int channel, int point) const
- QPolygon **getCurvePoints** (int channel) const
- int **getCurvePointX** (int channel, int point) const
- int **getCurvePointY** (int channel, int point) const
- [CurveType](#) **getCurveType** (int channel) const
- int **getCurveValue** (int channel, int bin) const
- QPolygon **getCurveValues** (int channel) const
- **ImageCurves** (bool sixteenBit)
- **ImageCurves** (const [CurvesContainer](#) &container)
- **ImageCurves** (const [ImageCurves](#) &other)
- bool **isCurvePointEnabled** (int channel, int point) const
- bool **isDirty** () const

*Curves properties.*

- bool **isLinear** () const
- bool **isLinear** (int channel) const
- bool **isSixteenBits** () const
- bool **loadCurvesFromGimpCurvesFile** (const QUrl &fileUrl)
- **ImageCurves** & **operator=** (const **ImageCurves** &other)
- bool **saveCurvesToGimpCurvesFile** (const QUrl &fileUrl) const

*Methods to save/load the curves values to/from a Gimp curves text file.*

- bool **setChannelFromBinary** (int channel, const QByteArray &array)
- void **setContainer** (const **CurvesContainer** &container)

*NOTE: bits depth must match.*

- void **setCurvePoint** (int channel, int point, const QPoint &val)
- void **setCurvePoints** (int channel, const QPolygon &vals)
- void **setCurvePointX** (int channel, int point, int x)
- void **setCurvePointY** (int channel, int point, int y)
- void **setCurveType** (**CurveType** type)
- void **setCurveType** (int channel, **CurveType** type)
- void **setCurveValue** (int channel, int bin, int val)

*Methods to set manually the curves values.*

- void **setCurveValues** (int channel, const QPolygon &vals)
- void **unsetCurvePoint** (int channel, int point)

### Static Public Member Functions

- static QPoint **getDisabledValue** ()

### Static Public Attributes

- static const int **MULTIPLIER\_16BIT** = 255
- static const int **NUM\_CHANNELS** = 5
- static const int **NUMBER\_OF\_POINTS** = 17

## 6.790.1 Member Enumeration Documentation

### 6.790.1.1 CurveType

```
enum Digikam::ImageCurves::CurveType
```

Enumerator

CURVE_SMOOTH	Smooth curve type.
CURVE_FREE	Freehand curve type.

## 6.790.2 Member Function Documentation

### 6.790.2.1 channelToBinary()

```
QByteArray Digikam::ImageCurves::channelToBinary (
```

```
int channel ) const
```

Writes the given channel to a raw binary representation. Note that 16bit free curves take a lot of memory (~85kB) while all other forms take less than 400 bytes.

Binary format:

Version 1 :16 Type 0,1,2 : 8 Bytes depth 1,2 : 8 reserved :32 count :32

Type 0 (linear curve): Type 1 (smooth curve): for (0...count) point.x :32 point.y :32 Type 2 (free curve): for (0...count) if (Bytes depth == 1) value : 8 else if (Bytes depth == 2) value :16

In Big Endian byte order. Data then converted to base64.

### 6.790.2.2 fillFromOtherCurves()

```
void Digikam::ImageCurves::fillFromOtherCurves (
    const ImageCurves *const otherCurves )
```

Fills this curves with the data supplied by another curves object. This ensures that 8 and 16 bit curves are properly converted.

Parameters

<i>otherCurves</i>	other curves object to adapt config from
--------------------	--

### 6.790.2.3 getContainer() [1/2]

```
CurvesContainer Digikam::ImageCurves::getContainer ( ) const
```

Returns a container with the settings for all channels of this Curves object

### 6.790.2.4 getContainer() [2/2]

```
CurvesContainer Digikam::ImageCurves::getContainer (
    int channel ) const
```

Returns a container containing the values of this Curves object for the given channel, and linear values for all other channels.

### 6.790.2.5 isLinear()

```
bool Digikam::ImageCurves::isLinear (
    int channel ) const
```

Returns true if the curve is linear for the given channel, or all channels.

### 6.790.2.6 setChannelFromBinary()

```
bool Digikam::ImageCurves::setChannelFromBinary (
    int channel,
    const QByteArray & array )
```

Set the channel from the given raw binary representation. The data is checked for validity, only on valid data true is returned. Note that the bytes depth (isSixteenBits()) of the encoded representation must match the depth of this curves object.

## 6.790.3 Member Data Documentation

### 6.790.3.1 MULTIPLIER\_16BIT

```
const int Digikam::ImageCurves::MULTIPLIER_16BIT = 255 [static]
```

Curve points have to multiplied with this value for 16 bit images.

### 6.790.3.2 NUM\_CHANNELS

```
const int Digikam::ImageCurves::NUM_CHANNELS = 5 [static]
```

Number of channels in a curve.

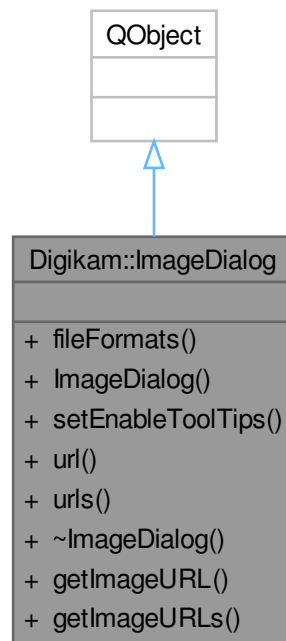
### 6.790.3.3 NUMBER\_OF\_POINTS

```
const int Digikam::ImageCurves::NUMBER_OF_POINTS = 17 [static]
```

The max number of points contained in a curve.

## 6.791 Digikam::ImageDialog Class Reference

Inheritance diagram for Digikam::ImageDialog:



### Classes

- class [Private](#)

### Public Member Functions

- `QStringList fileFormats ()` const
- `ImageDialog (QWidget *const parent, const QUrl &url, bool singleSelect=false, const QString &caption=QString())`
- void `setEnableToolTips (bool val)`
- `QUrl url ()` const
- `QList< QUrl > urls ()` const

### Static Public Member Functions

- static `QUrl getImageURL (QWidget *const parent, const QUrl &url, const QString &caption=QString())`
- static `QList< QUrl > getImageURLs (QWidget *const parent, const QUrl &url, const QString &caption=QString())`

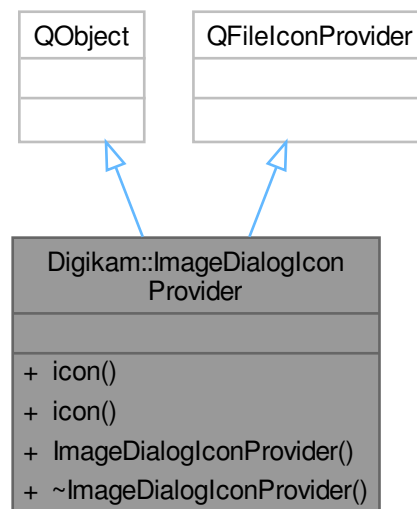
## 6.792 Digikam::ImageDialog::Private Class Reference

### Public Attributes

- [DFileDialog](#) \* **dlg** = nullptr
- QStringList **fileFormats**
- [ImageDialogIconProvider](#) \* **provider** = nullptr
- bool **showToolTips** = true
- [ImageDialogToolTip](#) \* **toolTip** = nullptr
- QModelIndex **toolTipIndex**
- QTimer \* **toolTipTimer** = nullptr
- QUrl **toolTipUrl**
- QAbstractItemView \* **toolTipView** = nullptr
- QList< QUrl > **urls**

## 6.793 Digikam::ImageDialogIconProvider Class Reference

Inheritance diagram for Digikam::ImageDialogIconProvider:



### Signals

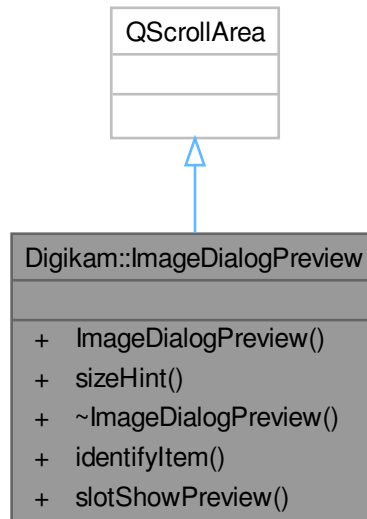
- void **signalThumbnailRefresh** ()

### Public Member Functions

- QIcon **icon** (const QFileInfo &info) const override
- QIcon **icon** (QAbstractFileIconProvider::IconType type) const override

## 6.794 Digikam::ImageDialogPreview Class Reference

Inheritance diagram for Digikam::ImageDialogPreview:



### Public Slots

- void **slotShowPreview** (const QUrl &url)

### Public Member Functions

- **ImageDialogPreview** (QWidget \*const parent=nullptr)
- QSize **sizeHint** () const override

### Static Public Member Functions

- static QString **identifyItem** (const QUrl &url, const QImage &preview=QImage())



## 6.795 Digikam::ImageDialogToolTip Class Reference

Inheritance diagram for Digikam::ImageDialogToolTip:



### Public Member Functions

- void **setData** (QAbstractItemView \*const view, const QModelIndex &index, const QUrl &url)

### Public Member Functions inherited from [Digikam::DItemToolTip](#)

- **DItemToolTip** (QWidget \*const parent=nullptr)

### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DItemToolTip](#)

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- void **resizeEvent** (QResizeEvent \*) override
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()

## 6.796 Digikam::ImageGuideWidget Class Reference

Inheritance diagram for Digikam::ImageGuideWidget:



### Public Types

- enum **ColorPointSrc** { **OriginalImage** = 0 , **PreviewImage** , **TargetPreviewImage** }
- enum **GuideToolMode** { **HVGuideMode** = 0 , **PickColorMode** }

## Public Slots

- void **slotChangeGuideColor** (const QColor &color)
- void **slotChangeGuideSize** (int size)
- void **slotPreviewModeChanged** (int mode)

## Signals

- void **signalResized** ()
- void **spotPositionChangedFromOriginal** (const [Digikam::DColor](#) &color, const QPoint &position)
- void **spotPositionChangedFromTarget** (const [Digikam::DColor](#) &color, const QPoint &position)

## Public Member Functions

- void **exposureSettingsChanged** ()
- QImage **getMask** () const
- [DColor](#) **getSpotColor** (int getColorFrom) const
- QPoint **getSpotPosition** () const
- void **ICCSettingsChanged** ()
- **ImageGuideWidget** (QWidget \*const parent=nullptr, bool spotVisible=true, int guideMode=PickColor↔ Mode, const QColor &guideColor=Qt::red, int guideSize=1, bool blink=false, [Imagelface::PreviewType](#) type=[Imagelface::FullImage](#))
- [Imagelface](#) \* **imagelface** () const
- int **previewMode** () const
- void **resetPoints** ()
- void **resetSpotPosition** ()
- void **setBackgroundColor** (const QColor &)
- void **setEraseMode** (bool erase)
- void **setMaskCursor** ()
- void **setMaskEnabled** (bool enabled)
- void **setMaskPenSize** (int size)
- void **setPaintColor** (const QColor &color)
- void **setPoints** (const QPolygon &p, bool drawLine=false)
- void **setSpotVisible** (bool spotVisible, bool blink=false)
- void **setSpotVisibleNoUpdate** (bool spotVisible)
- void **updatePreview** ()

## Protected Member Functions

- void **drawLineTo** (const QPoint &endPoint)
- void **drawLineTo** (int width, bool erase, const QColor &color, const QPoint &start, const QPoint &end)
- void **drawText** (QPainter \*const p, const QPoint &corner, const QString &text)
- void **enterEvent** (QEnterEvent \*) override
- void **leaveEvent** (QEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override
- void **setSpotPosition** (const QPoint &point)
- void **timerEvent** (QTimerEvent \*) override
- QPoint **translateItemPosition** (const QPoint &point, bool src) const
- QPoint **translatePointPosition** (const QPoint &point) const
- void **updateMaskCursor** ()
- void **updatePixmap** ()
- void **updateSpotPosition** (int x, int y)

## 6.797 Digikam::ImageHistogram Class Reference

Inheritance diagram for Digikam::ImageHistogram:



### Signals

- void [calculationAboutToStart](#) ()
- void **calculationFinished** (bool success)
- void [calculationStarted](#) ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Public Member Functions

- void **calculate** ()
- void **calculateInThread** ()
- double **getCount** (int channel, int start, int end) const
- int **getHistogramSegments** () const
- double **getMaximum** (int channel, int start, int end) const
- int **getMaxSegmentIndex** () const
- double **getMean** (int channel, int start, int end) const
- int **getMedian** (int channel, int start, int end) const
- double **getPixels** () const
- double **getStdDev** (int channel, int start, int end) const
- double **getValue** (int channel, int bin) const
- **ImageHistogram** (const [DImg](#) &img, QObject \*const parent=nullptr)
- bool **isCalculating** () const
- bool **isSixteenBit** () const
- bool **isValid** () const
- void **stopCalculation** ()

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Protected Member Functions

- void **run** () override

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

### Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## 6.797.1 Member Function Documentation

### 6.797.1.1 `calculate()`

```
void Digikam::ImageHistogram::calculate ( )
```

Started computation: synchronous or threaded.

### 6.797.1.2 `calculationAboutToStart`

```
void Digikam::ImageHistogram::calculationAboutToStart ( ) [signal]
```

when calculation in thread is initiated, from other thread

### 6.797.1.3 `calculationStarted`

```
void Digikam::ImageHistogram::calculationStarted ( ) [signal]
```

emitted from calculation thread

### 6.797.1.4 `isSixteenBit()`

```
bool Digikam::ImageHistogram::isSixteenBit ( ) const
```

Methods to access the histogram data.

### 6.797.1.5 `run()`

```
void Digikam::ImageHistogram::run ( ) [override], [protected], [virtual]
```

Implement this pure virtual function in your subclass.

Implements [Digikam::DynamicThread](#).

### 6.797.1.6 stopCalculation()

```
void Digikam::ImageHistogram::stopCalculation ( )
```

Stop threaded computation.

## 6.798 Digikam::ImageHistoryEntry Class Reference

### Public Member Functions

- bool **isNull** () const

### Public Attributes

- QString **history**
- qlonglong **imgeld** = 0
- QString **uuid**

## 6.799 Digikam::Imagelface Class Reference

### Public Types

- enum **PreviewType** { [FullImage](#) , [ImageSelection](#) }

### Public Member Functions

- [DColor](#) **colorInfoFromOriginal** (const QPoint &point) const
- [DColor](#) **colorInfoFromPreview** (const QPoint &point) const
- [DColor](#) **colorInfoFromTargetPreview** (const QPoint &point) const
- void **convertOriginalColorDepth** (int depth)
- QPixmap **convertToPixmap** (const [DImg](#) &img) const
- void **crop** (const QRect &region)
- [Imagelface](#) (const QSize &size=QSize(0, 0))
- [DImg](#) \* **original** () const
- bool **originalHasAlpha** () const
- [IccProfile](#) **originalIccProfile** () const
- [MetaEngineData](#) **originalMetadata** () const
- [PhotoInfoContainer](#) **originalPhotoInfo** () const
- bool **originalSixteenBit** () const
- QSize **originalSize** () const
- void **paint** (QPaintDevice \*const device, const QRect &rect, QPainter \*const painter=nullptr)
- [DImg](#) **preview** () const
- bool **previewHasAlpha** () const
- [DImg](#) \* **previewReference** ()
- bool **previewSixteenBit** () const
- QSize **previewSize** () const
- [PreviewType](#) **previewType** () const
- [DImg](#) **selection** () const
- QRect **selectionRect** () const
- void **setOriginal** (const QString &caller, const [FilterAction](#) &action, const [DImg](#) &img)
- void **setOriginalIccProfile** (const [IccProfile](#) &profile)
- void **setOriginalMetadata** (const [MetaEngineData](#) &meta)
- void **setPreview** (const [DImg](#) &img)
- void **setPreviewIccProfile** (const [IccProfile](#) &profile)
- [DImg](#) **setPreviewSize** (const QSize &size) const
- void **setPreviewType** ([PreviewType](#) type=[FullImage](#))
- void **setSelection** (const QString &caller, const [FilterAction](#) &action, const [DImg](#) &img)



## 6.799.1 Member Enumeration Documentation

### 6.799.1.1 PreviewType

```
enum Digikam::ImageIface::PreviewType
```

Enumerator

FullImage	Preview will be rendered using full image.
ImageSelection	Preview will be rendered using current selection from editor canvas.

## 6.799.2 Constructor & Destructor Documentation

### 6.799.2.1 ImageIface()

```
Digikam::ImageIface::ImageIface (
    const QSize & size = QSize(0, 0) ) [explicit]
```

Standard constructor. Size is the constrain dimension of preview. This can be null size.

## 6.799.3 Member Function Documentation

### 6.799.3.1 colorInfoFromOriginal()

```
DColor Digikam::ImageIface::colorInfoFromOriginal (
    const QPoint & point ) const
```

Get colors from original, (unchanged) preview or target preview (set by setPreviewImage) image.

### 6.799.3.2 convertOriginalColorDepth()

```
void Digikam::ImageIface::convertOriginalColorDepth (
    int depth )
```

Convert depth of original image.

### 6.799.3.3 convertToPixmap()

```
QPixmap Digikam::ImageIface::convertToPixmap (
    const DImg & img ) const
```

Convert a [DImg](#) image to a pixmap for screen using color managed view if necessary.

#### 6.799.3.4 crop()

```
void Digikam::ImageIface::crop (
    const QRect & region )
```

Crop the original image currently hosted by editor to the region.

#### 6.799.3.5 original()

```
DImg * Digikam::ImageIface::original ( ) const
```

Return a pointer to the [DImg](#) object representing the original image. This object may not be modified or stored. Make copies if you need.

#### 6.799.3.6 originalIccProfile()

```
IccProfile Digikam::ImageIface::originalIccProfile ( ) const
```

Original image meta-data management methods.

#### 6.799.3.7 originalSize()

```
QSize Digikam::ImageIface::originalSize ( ) const
```

Methods to get/set original image information.

#### 6.799.3.8 paint()

```
void Digikam::ImageIface::paint (
    QPaintDevice *const device,
    const QRect & rect,
    QPainter *const painter = nullptr )
```

Paint the current target preview image (or the preview image, if `setPreview` has not been called) on the given paint device. at `x|y`, with given maximum width and height taken from rectangle `rect`.

#### 6.799.3.9 preview()

```
DImg Digikam::ImageIface::preview ( ) const
```

Return a [DImg](#) object representing the preview image.

#### 6.799.3.10 previewReference()

```
DImg * Digikam::ImageIface::previewReference ( )
```

Return a pointer to the [DImg](#) object representing the preview image. This function is a backdoor for preview editing.

### 6.799.3.11 previewSize()

```
QSize Digikam::ImageIface::previewSize ( ) const
```

Methods to get/set preview image information.

### 6.799.3.12 selection()

```
DImg Digikam::ImageIface::selection ( ) const
```

Return a [DImg](#) object representing the current original image selection.

### 6.799.3.13 selectionRect()

```
QRect Digikam::ImageIface::selectionRect ( ) const
```

Return current image selection position and size into original image coordinates.

### 6.799.3.14 setOriginal()

```
void Digikam::ImageIface::setOriginal (
    const QString & caller,
    const QAction & action,
    const DImg & img )
```

Replace the data of the original with the given image. The characteristics of the data must match the characteristics of the original image as returned by the original...() methods, The size of image can be changed. Caller is an i18n'ed string that will be shown as the undo/redo action name.

### 6.799.3.15 setOriginalIccProfile()

```
void Digikam::ImageIface::setOriginalIccProfile (
    const IccProfile & profile )
```

Set the color profile of the original image.

### 6.799.3.16 setPreview()

```
void Digikam::ImageIface::setPreview (
    const DImg & img )
```

Replace the stored target preview with the given image. The characteristics of the data must match the characteristics of the current as returned by the preview...() methods. The target preview image is used by the [paint\(\)](#) and [colorInfoFromTargetPreview\(\)](#) methods. The image returned by [getPreview\(\)](#) is unaffected.

**6.799.3.17 setPreviewIccProfile()**

```
void Digikam::ImageIface::setPreviewIccProfile (
    const IccProfile & profile )
```

Set the color profile of the preview image.

**6.799.3.18 setPreviewSize()**

```
DImg Digikam::ImageIface::setPreviewSize (
    const QSize & size ) const
```

Sets preview size and returns new preview as with `getPreview`. The parameters are only hints, `previewSize()` may differ from size.

**6.799.3.19 setPreviewType()**

```
void Digikam::ImageIface::setPreviewType (
    PreviewType type = FullImage )
```

If `useSelection` is true, preview will be rendered using current selection in editor instead the full image. Default preview is `FullImage`.

**6.799.3.20 setSelection()**

```
void Digikam::ImageIface::setSelection (
    const QString & caller,
    const FilterAction & action,
    const DImg & img )
```

Replace the data of the current original image selection with the given data. The characteristics of the data must match the characteristics of the current selection as returned by the `selectionWidth()`, `selectionHeight()`, `originalSixteenBit()` and `originalHasAlpha()` methods. Caller is an i18n'ed string that will be shown as the undo/redo action name.

**6.800 Digikam::ImageLevels Class Reference****Public Member Functions**

- double **getLevelGammaValue** (int channel)
- int **getLevelHighInputValue** (int channel)
- int **getLevelHighOutputValue** (int channel)
- int **getLevelLowInputValue** (int channel)
- int **getLevelLowOutputValue** (int channel)
- **ImageLevels** (bool sixteenBit)
- bool **isDirty** ()
- bool **isSixteenBits** ()
- void **levelsAuto** (const [ImageHistogram](#) \*const hist)
- void **levelsBlackToneAdjustByColors** (int channel, const [DColor](#) &color)

- void **levelsCalculateTransfers** ()
- void **levelsChannelAuto** (const [ImageHistogram](#) \*const hist, int channel)
- void **levelsChannelReset** (int channel)
- void **levelsGrayToneAdjustByColors** (int channel, const [DColor](#) &color)
- int **levelsInputFromColor** (int channel, const [DColor](#) &color)
- float **levelsLutFunc** (int nchannels, int channel, float value)
- void **levelsLutProcess** (uchar \*const srcPR, uchar \*const destPR, uint w, uint h)
- void **levelsLutSetup** (int nchannels)
- void **levelsWhiteToneAdjustByColors** (int channel, const [DColor](#) &color)
- bool **loadLevelsFromGimpLevelsFile** (const [QUrl](#) &fileUrl)
- void **reset** ()
- bool **saveLevelsToGimpLevelsFile** (const [QUrl](#) &fileUrl)
- void **setLevelGammaValue** (int channel, double val)
- void **setLevelHighInputValue** (int channel, int val)
- void **setLevelHighOutputValue** (int channel, int val)
- void **setLevelLowInputValue** (int channel, int val)
- void **setLevelLowOutputValue** (int channel, int val)

## 6.800.1 Member Function Documentation

### 6.800.1.1 levelsChannelReset()

```
void Digikam::ImageLevels::levelsChannelReset (  
    int channel )
```

Methods to manipulate the levels data.

### 6.800.1.2 saveLevelsToGimpLevelsFile()

```
bool Digikam::ImageLevels::saveLevelsToGimpLevelsFile (  
    const QUrl & fileUrl )
```

Methods to save/load the levels values to/from a Gimp levels text file.

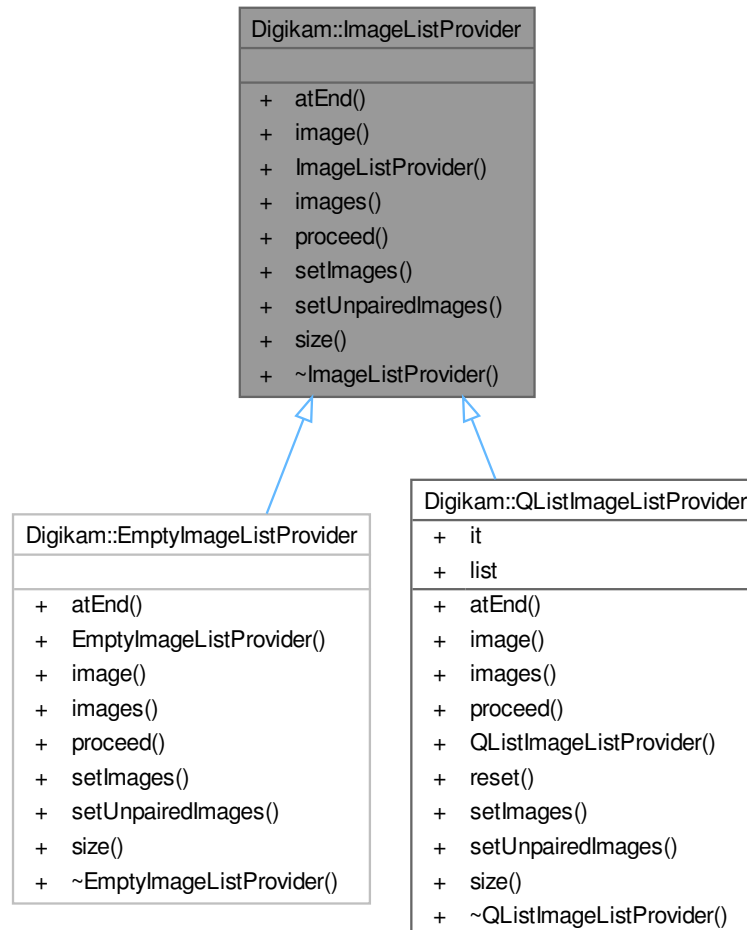
### 6.800.1.3 setLevelGammaValue()

```
void Digikam::ImageLevels::setLevelGammaValue (  
    int channel,  
    double val )
```

Methods to set manually the levels values.

## 6.801 Digikam::ImageListProvider Class Reference

Inheritance diagram for Digikam::ImageListProvider:



### Public Member Functions

- virtual bool **atEnd** () const =0
- virtual QPair< QImage \*, QString > **image** ()=0
- virtual QList< QPair< QImage \*, QString > > **images** ()=0
- virtual void **proceed** (int steps=1)=0
- virtual void **setImages** (const QList< QPair< QImage \*, QString > > &)=0
- virtual void **setUnpairedImages** (const QList< QImage \* > &)=0
- virtual int **size** () const =0

### 6.801.1 Detailed Description

This class provides access to a list of unspecified entities, where for each entry a QImage can be provided. Only forward iteration is required.

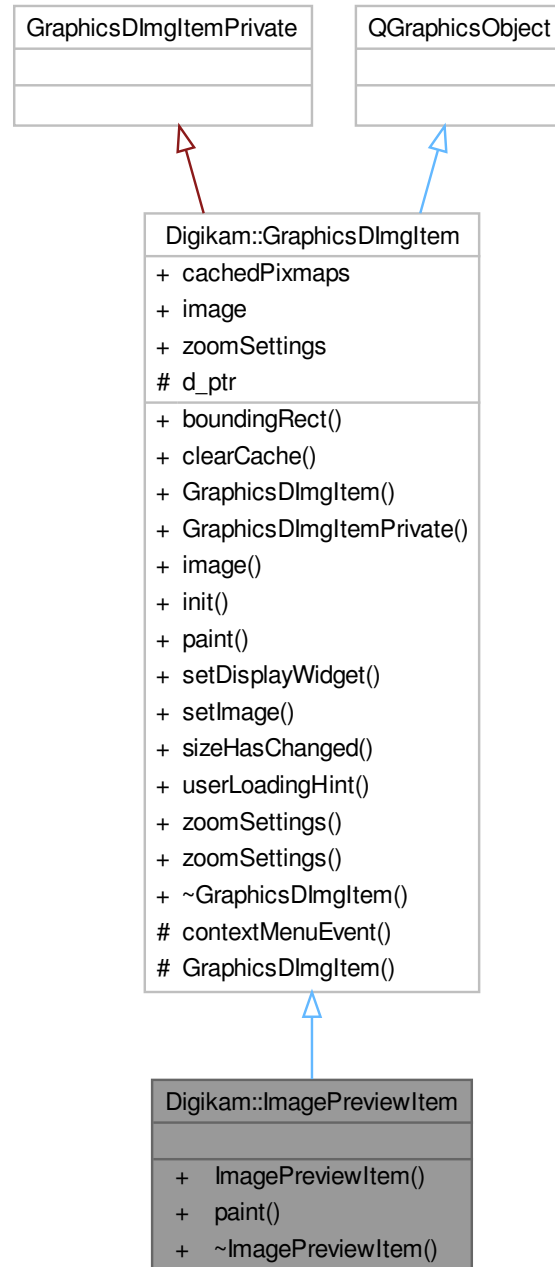
## 6.802 Digikam::ImageMetadataContainer Class Reference

### Public Attributes

- bool **allFieldsNull** = true
- QString **aperture**
- QString **exposureMode**
- QString **exposureProgram**
- QString **exposureTime**
- QString **flashMode**
- QString **focalLength**
- QString **focalLength35**
- QString **lens**
- QString **make**
- QString **meteringMode**
- QString **model**
- QString **sensitivity**
- QString **subjectDistance**
- QString **subjectDistanceCategory**
- QString **whiteBalance**
- QString **whiteBalanceColorTemperature**

## 6.803 Digikam::ImagePreviewItem Class Reference

Inheritance diagram for Digikam::ImagePreviewItem:



### Public Member Functions

- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override



## Public Member Functions inherited from [Digikam::GraphicsDImgItem](#)

- `QRectF boundingRect ()` const override
- `void clearCache ()`
- `GraphicsDImgItem (QGraphicsItem *const parent=nullptr)`
- `GraphicsDImgItemPrivate ()=default`
- `DImg image ()` const
- `void init (GraphicsDImgItem *const q)`
- `void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)` override
- `void setDisplayWidget (QWidget *const widget)`
- `void setImage (const DImg &img)`
- `void sizeHasChanged ()`
- `virtual QString userLoadingHint ()` const
- `ImageZoomSettings * zoomSettings ()`
- `const ImageZoomSettings * zoomSettings ()` const

## Additional Inherited Members

## Signals inherited from [Digikam::GraphicsDImgItem](#)

- `void imageChanged ()`
- `void imageSizeChanged (const QSizeF &size)`
- `void showContextMenu (QGraphicsSceneContextMenuEvent *e)`

## Public Attributes inherited from [Digikam::GraphicsDImgItem](#)

- `CachedPixmap cachedPixmap`
- `DImg image`
- `ImageZoomSettings zoomSettings`

## Protected Member Functions inherited from [Digikam::GraphicsDImgItem](#)

- `void contextMenuEvent (QGraphicsSceneContextMenuEvent *e)` override
- `GraphicsDImgItem (GraphicsDImgItemPrivate &dd, QGraphicsItem *const parent)`

## Protected Attributes inherited from [Digikam::GraphicsDImgItem](#)

- `GraphicsDImgItemPrivate *const d_ptr`

## 6.804 Digikam::ImageQualityCalculator Class Reference

### Classes

- struct [ResultDetection](#)

**Public Member Functions**

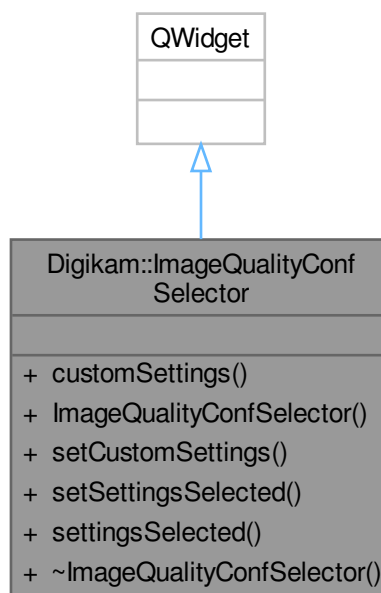
- void **addDetectionResult** (const QString &name, const float score, const float weight) const
- float **calculateQuality** () const

**6.805 Digikam::ImageQualityCalculator::ResultDetection Struct Reference****Public Attributes**

- QString **detetionType**
- float **score**
- float **weight**

**6.806 Digikam::ImageQualityConfSelector Class Reference**

Inheritance diagram for Digikam::ImageQualityConfSelector:

**Public Types**

- enum `SettingsType` { `GlobalSettings` = 0 , `CustomSettings` }

## Signals

- void **signalQualitySetup** ()
- void **signalSettingsChanged** ()

## Public Member Functions

- [ImageQualityContainer](#) **customSettings** () const
- **ImageQualityConfSelector** (QWidget \*const parent=nullptr)
- void **setCustomSettings** (const [ImageQualityContainer](#) &settings)
- void **setSettingsSelected** ([SettingsType](#) type)
- [SettingsType](#) **settingsSelected** () const

## 6.806.1 Member Enumeration Documentation

### 6.806.1.1 SettingsType

```
enum Digikam::ImageQualityConfSelector::SettingsType
```

#### Enumerator

GlobalSettings	Global settings available in setup dialog.
CustomSettings	Settings customized by end-user.

## 6.807 Digikam::ImageQualityContainer Class Reference

### Public Member Functions

- **ImageQualityContainer** (const [ImageQualityContainer](#) &other)
- [ImageQualityContainer](#) & **operator=** (const [ImageQualityContainer](#) &other)
- void **readFromConfig** ()
- void **readFromConfig** (const KConfigGroup &)
- void **writeToConfig** ()
- void **writeToConfig** (KConfigGroup &)

### Public Attributes

- int **acceptedThreshold**  
*Item accepted threshold.*
- int **blurWeight**  
*Item blur level.*
- int **compressionWeight**  
*Item compression level.*
- bool **detectAesthetic**  
*Enable image aesthetic detection.*
- bool **detectBlur**

- Enable image blur detection.*
- bool **detectCompression**

*Enable image compression detection.*
- bool **detectExposure**

*Enable image over and under exposure detection.*
- bool **detectNoise**

*Enable image noise detection.*
- int **exposureWeight**

*Item exposure level.*
- bool **highQAccepted**

*Assign Accepted property to high quality.*
- bool **lowQRejected**

*Assign Rejected property to low quality.*
- bool **mediumQPending**

*Assign Pending property to medium quality.*
- int **noiseWeight**

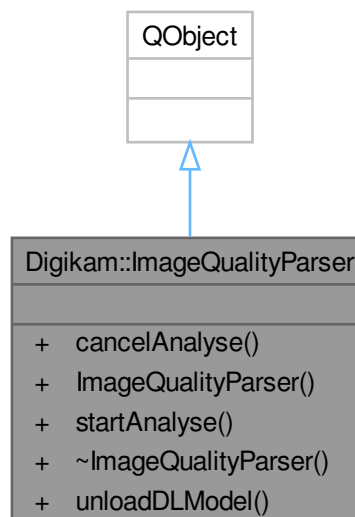
*Item noise level.*
- int **pendingThreshold**

*Item pending threshold.*
- int **rejectedThreshold**

*Item rejection threshold.*

## 6.808 Digikam::ImageQualityParser Class Reference

Inheritance diagram for Digikam::ImageQualityParser:



## Classes

- class [Private](#)

## Public Member Functions

- void **cancelAnalyse** ()
- [ImageQualityParser](#) (const [DImg](#) &image, const [ImageQualityContainer](#) &settings, PickLabel \*const label)
- void **startAnalyse** ()

## Static Public Member Functions

- static void **unloadDLModel** ()

## 6.808.1 Constructor & Destructor Documentation

### 6.808.1.1 ImageQualityParser()

```
Digikam::ImageQualityParser::ImageQualityParser (
    const DImg & image,
    const ImageQualityContainer & settings,
    PickLabel *const label ) [explicit]
```

Standard constructor with picklabel container to fill at end of analyze.

## 6.808.2 Member Function Documentation

### 6.808.2.1 startAnalyse()

```
void Digikam::ImageQualityParser::startAnalyse ( )
```

Perform quality estimation and fill Pick Label value accordingly.

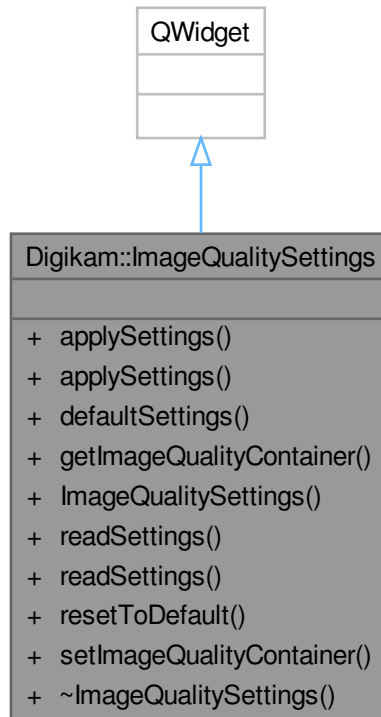
## 6.809 Digikam::ImageQualityParser::Private Class Reference

### Public Attributes

- [ImageQualityCalculator](#) \* **calculator** = nullptr
- [DImg](#) **image**  
*original image*
- [ImageQualityContainer](#) **imq**
- PickLabel \* **label** = nullptr
- volatile bool **running** = true

## 6.810 Digikam::ImageQualitySettings Class Reference

Inheritance diagram for Digikam::ImageQualitySettings:



### Signals

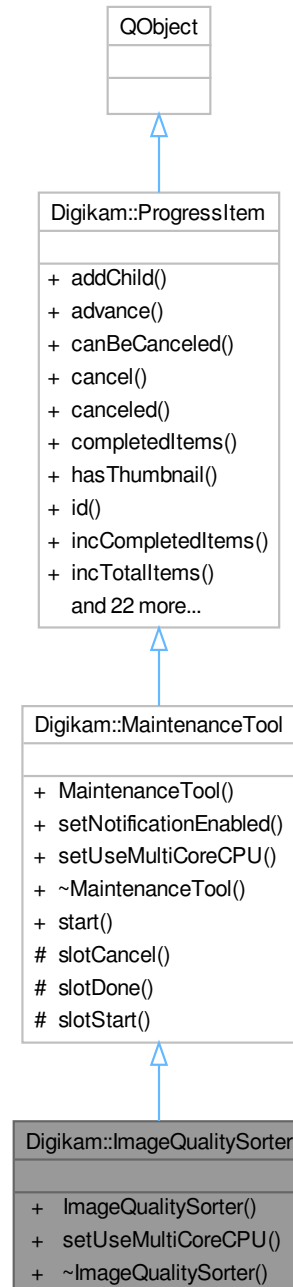
- void **signalSettingsChanged** ()

### Public Member Functions

- void **applySettings** ()
- void **applySettings** (KConfigGroup &)
- [ImageQualityContainer](#) **defaultSettings** () const
- [ImageQualityContainer](#) **getImageQualityContainer** () const
- **ImageQualitySettings** (QWidget \*const parent=nullptr)
- void **readSettings** ()
- void **readSettings** (const KConfigGroup &)
- void **resetToDefault** ()
- void **setImageQualityContainer** (const [ImageQualityContainer](#) &imq)

## 6.811 Digikam::ImageQualitySorter Class Reference

Inheritance diagram for Digikam::ImageQualitySorter:



### Public Types

- enum `QualityScanMode` { `AllItems = 0`, `NonAssignedItems` }

### Public Member Functions

- [ImageQualitySorter](#) ([QualityScanMode](#) mode, const AlbumList &list=AlbumList(), const [ImageQualityContainer](#) &quality=[ImageQualityContainer](#)(), [ProgressItem](#) \*const parent=nullptr)
- void [setUseMultiCoreCPU](#) (bool b) override

### Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- [MaintenanceTool](#) (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)

### Public Member Functions inherited from [Digikam::ProgressItem](#)

- void [addChild](#) ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)
  - Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void [cancel](#) ()
- bool [canceled](#) () const
- unsigned int [completedItems](#) () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool [incCompletedItems](#) (unsigned int v=1)
- void [incTotalItems](#) (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool [canBeCanceled](#), bool hasThumb)
- void [removeChild](#) ([ProgressItem](#) \*const kiddo)
- void [reset](#) ()
  - Reset the progress value of this item to 0 and the status string to the empty string.*
- void [setComplete](#) ()
  - Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool [setCompletedItems](#) (unsigned int v)
- void [setLabel](#) (const QString &v)
- void [setProgress](#) (unsigned int v)
  - Set the progress (percentage of completion) value of this item.*
- void [setShowAtStart](#) (bool showAtStart)
  - Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void [setStatus](#) (const QString &v)
  - Set the string to be used for showing this item's current status.*
- void [setThumbnail](#) (const QIcon &icon)
  - Sets whether this item has a thumbnail.*
- void [setTotalItems](#) (unsigned int v)
- void [setUsesBusyIndicator](#) (bool useBusyIndicator)
  - Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool [showAtStart](#) () const
- const QString & [status](#) () const
- bool [totalCompleted](#) () const
- unsigned int [totalItems](#) () const
- void [updateProgress](#) ()
  - Recalculate progress according to total/completed items and update.*
- bool [usesBusyIndicator](#) () const



## Additional Inherited Members

### Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

### Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

### Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

### Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.811.1 Member Enumeration Documentation

### 6.811.1.1 QualityScanMode

```
enum Digikam::ImageQualitySorter::QualityScanMode
```

## Enumerator

AllItems	Clean all Pick Labels assignments and re-scan all items.
NonAssignedItems	Scan only items with no Pick Labels assigned.

## 6.811.2 Constructor & Destructor Documentation

### 6.811.2.1 ImageQualitySorter()

```
Digikam::ImageQualitySorter::ImageQualitySorter (
    QualityScanMode mode,
    const AlbumList & list = AlbumList(),
    const ImageQualityContainer & quality = ImageQualityContainer(),
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor using AlbumList as argument. If list is empty, whole Albums collection is processed.

## 6.811.3 Member Function Documentation

### 6.811.3.1 setUseMultiCoreCPU()

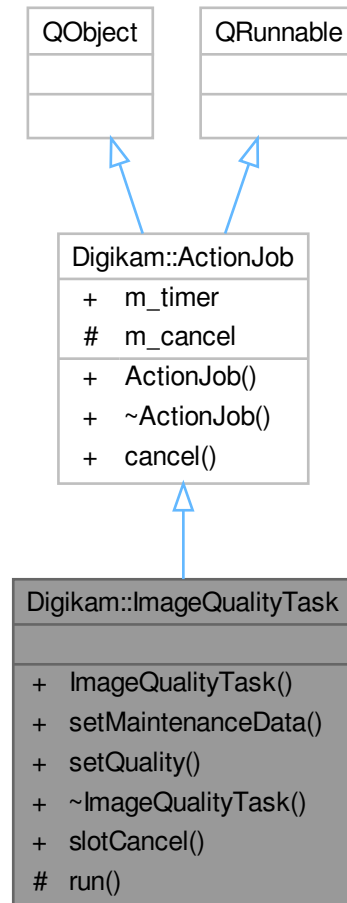
```
void Digikam::ImageQualitySorter::setUseMultiCoreCPU (
    bool ) [override], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented from [Digikam::MaintenanceTool](#).

## 6.812 Digikam::ImageQualityTask Class Reference

Inheritance diagram for Digikam::ImageQualityTask:



### Public Slots

- void `slotCancel ()`

### Public Slots inherited from [Digikam::ActionJob](#)

- void `cancel ()`

### Signals

- void `signalFinished` (const [ItemInfo](#) &, const QImage &, int)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Member Functions

- void **setMaintenanceData** ([MaintenanceData](#) \*const data=nullptr)
- void **setQuality** (const [ImageQualityContainer](#) &quality)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Protected Member Functions

- void **run** () override

### Additional Inherited Members

### Public Attributes inherited from [Digikam::ActionJob](#)

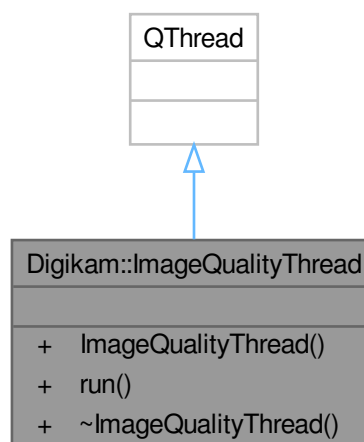
- QElapsedTimer [m\\_timer](#)

### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.813 Digikam::ImageQualityThread Class Reference

Inheritance diagram for Digikam::ImageQualityThread:

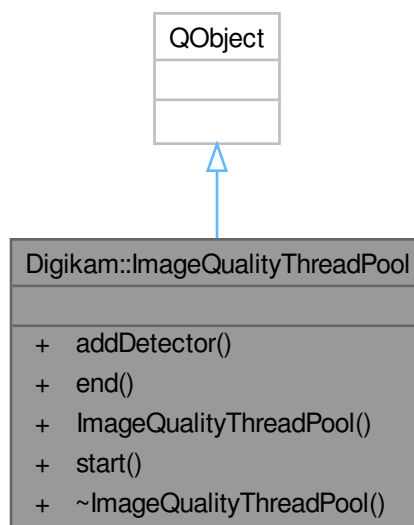


### Public Member Functions

- **ImageQualityThread** (QObject \*const parent, [AbstractDetector](#) \*const detector, const cv::Mat &image, [ImageQualityCalculator](#) \*const calculator, float weight\_quality)
- void **run** () override

## 6.814 Digikam::ImageQualityThreadPool Class Reference

Inheritance diagram for Digikam::ImageQualityThreadPool:

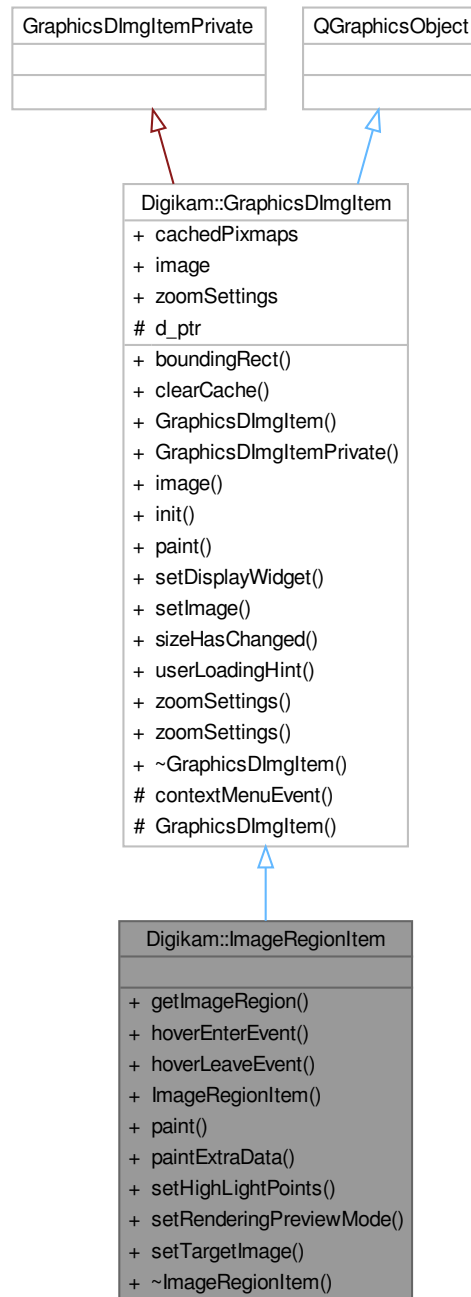


### Public Member Functions

- void **addDetector** (const cv::Mat &image, float weight\_quality, [AbstractDetector](#) \*const detector)
- void **end** ()
- **ImageQualityThreadPool** (QObject \*const parent, [ImageQualityCalculator](#) \*const calculator)
- void **start** ()

## 6.815 Digikam::ImageRegionItem Class Reference

Inheritance diagram for Digikam::ImageRegionItem:



### Public Member Functions

- `QRect` **getImageRegion** () const
- void **hoverEnterEvent** (`QGraphicsSceneHoverEvent *`) override

- void **hoverLeaveEvent** (QGraphicsSceneHoverEvent \*) override
- **ImageRegionItem** (ImageRegionWidget \*const view, bool paintExtras=true)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override
- void **paintExtraData** (QPainter \*const painter)
- void **setHighLightPoints** (const QPolygon &pointsList)
- void **setRenderingPreviewMode** (int mode)
- void **setTargetImage** (const DImg &img)

### Public Member Functions inherited from Digikam::GraphicsDImgItem

- QRectF **boundingRect** () const override
- void **clearCache** ()
- **GraphicsDImgItem** (QGraphicsItem \*const parent=nullptr)
- **GraphicsDImgItemPrivate** ()=default
- **DImg image** () const
- void **init** (GraphicsDImgItem \*const q)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override
- void **setDisplayWidget** (QWidget \*const widget)
- void **setImage** (const DImg &img)
- void **sizeHasChanged** ()
- virtual QString **userLoadingHint** () const
- **ImageZoomSettings \* zoomSettings** ()
- const **ImageZoomSettings \* zoomSettings** () const

### Additional Inherited Members

### Signals inherited from Digikam::GraphicsDImgItem

- void **imageChanged** ()
- void **imageSizeChanged** (const QSizeF &size)
- void **showContextMenu** (QGraphicsSceneContextMenuEvent \*e)

### Public Attributes inherited from Digikam::GraphicsDImgItem

- **CachedPixmaps cachedPixmaps**
- **DImg image**
- **ImageZoomSettings zoomSettings**

### Protected Member Functions inherited from Digikam::GraphicsDImgItem

- void **contextMenuEvent** (QGraphicsSceneContextMenuEvent \*e) override
- **GraphicsDImgItem** (GraphicsDImgItemPrivate &dd, QGraphicsItem \*const parent)

### Protected Attributes inherited from Digikam::GraphicsDImgItem

- GraphicsDImgItemPrivate \*const **d\_ptr**

## 6.816 Digikam::ImageRegionWidget Class Reference

Inheritance diagram for Digikam::ImageRegionWidget:



### Public Slots

- void **slotOriginalImageRegionChanged** (bool targetDone=true)
- void **slotPreviewModeChanged** (int mode)



## Signals

- void **signalCapturedPointFromOriginal** (const [Digikam::DColor](#) &, const QPoint &)
- void **signalOriginalClipFocusChanged** ()

## Signals inherited from [Digikam::GraphicsDImgView](#)

- void **activated** ()
- void **contentsMoved** (bool panningFinished)
- void **contentsMoving** (int, int)
- void **leftButtonClicked** ()
- void **leftButtonDoubleClicked** ()
- void **resized** ()
- void **rightButtonClicked** ()
- void **toNextImage** ()
- void **toPreviousImage** ()
- void **viewportRectChanged** (const QRectF &viewportRect)

## Public Member Functions

- bool **capturePointMode** () const
- void **exposureSettingsChanged** ()
- [DImg](#) **getOriginalImage** () const
- QRect **getOriginalImageRegionToRender** () const
- [DImg](#) **getOriginalRegionImage** (bool useDownscaledImage=false) const
- void **ICCSettingsChanged** ()
- [ImageRegionWidget](#) (QWidget \*const parent=nullptr, bool paintExtras=true)
- void **setCapturePointMode** (bool b)
- void **setHighLightPoints** (const QPolygon &pointsList)
- void **setPreviewImage** (const [DImg](#) &img)
- void **updateImage** (const [DImg](#) &img)

## Public Member Functions inherited from [Digikam::GraphicsDImgView](#)

- int **contentsX** () const
- int **contentsY** () const
- void **drawText** (QPainter \*p, const QRectF &rect, const QString &text)
- void **fitToWindow** ()
- [GraphicsDImgView](#) (QWidget \*const parent=nullptr)
- [GraphicsDImgItem](#) \* **item** () const
- [SinglePhotoPreviewLayout](#) \* **layout** () const
- [DImgPreviewItem](#) \* **previewItem** () const
- void **scrollPointOnPoint** (const QPointF &scenePos, const QPoint &viewportPos)
- void **setContentsPos** (int x, int y)
- void **setItem** ([GraphicsDImgItem](#) \*const item)
- void **toggleFullScreen** (bool set)
- QRect **visibleArea** () const

## Protected Member Functions

- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override

## Protected Member Functions inherited from [Digikam::GraphicsDImgView](#)

- virtual bool **acceptsMouseClicked** (QMouseEvent \*e)
- void **continuePanning** (const QPoint &pos)
- void **drawForeground** (QPainter \*painter, const QRectF &rect) override
- void **finishPanning** ()
- void **installPanIcon** ()
- void **mouseDoubleClickEvent** (QMouseEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override
- void **scrollContentsBy** (int dx, int dy) override
- void **setScaleFitToWindow** (bool value)
- void **setShowText** (bool value)
- void **startPanning** (const QPoint &pos)
- void **wheelEvent** (QWheelEvent \*) override

### Additional Inherited Members

## Protected Slots inherited from [Digikam::GraphicsDImgView](#)

- void **slotContentsMoved** ()
- void **slotCornerButtonPressed** ()
- void **slotPanIconHidden** ()
- virtual void **slotPanIconSelectionMoved** (const QRect &, bool)

## 6.816.1 Member Function Documentation

### 6.816.1.1 `getOriginalImageRegionToRender()`

```
QRect Digikam::ImageRegionWidget::getOriginalImageRegionToRender ( ) const
```

To get target image region area to render.

### 6.816.1.2 `getOriginalRegionImage()`

```
DImg Digikam::ImageRegionWidget::getOriginalRegionImage (
    bool useDownscaledImage = false ) const
```

To get target image region image to use for render operations If the bool parameter is true a downscaled version of the image region at screen resolution will be sent. Should be use to increase preview speed for the effects whose behaviour is a function of each pixel.

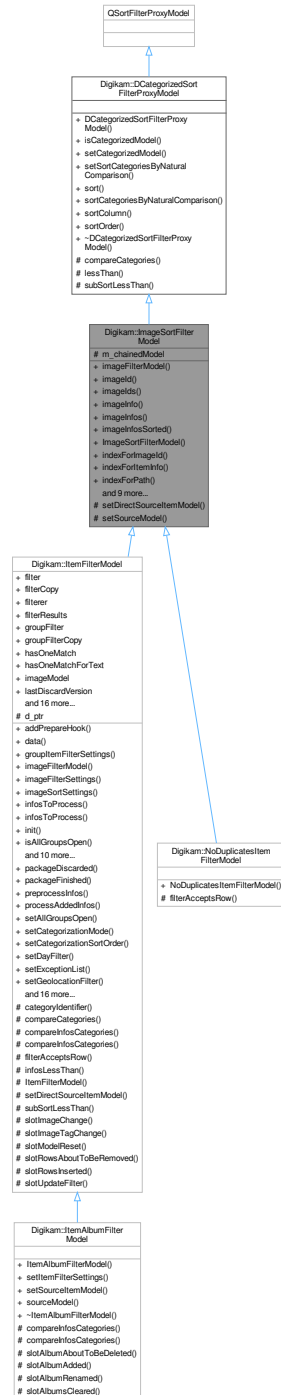
## 6.817 Digikam::ImageRelation Class Reference

### Public Attributes

- qlonglong **objectId** = 0
- qlonglong **subjectId** = 0
- DatabaseRelation::Type **type** = DatabaseRelation::UndefinedType

## 6.818 Digikam::ImageSortFilterModel Class Reference

Inheritance diagram for Digikam::ImageSortFilterModel:



### Public Member Functions

- virtual [ItemFilterModel](#) \* [imageFilterModel](#) () const
- qlonglong [imageId](#) (const QModelIndex &index) const

- `QList< qulonglong > imagelds` (const `QList< QModelIndex > &indexes`) const
- `ItemInfo imagelInfo` (const `QModelIndex &index`) const
- `QList< ItemInfo > imagelInfos` (const `QList< QModelIndex > &indexes`) const
- `QList< ItemInfo > imagelInfosSorted` () const
- `ImageSortFilterModel` (`QObject *const parent=nullptr`)
- `QModelIndex indexForImageld` (`qulonglong id`) const
- `QModelIndex indexForItemInfo` (const `ItemInfo &info`) const
- `QModelIndex indexForPath` (const `QString &filePath`) const
- `QModelIndex mapFromDirectSourceToSourceItemModel` (const `QModelIndex &sourceModel_index`) const
- `QModelIndex mapFromSourceItemModel` (const `QModelIndex &imagemodel_index`) const
- `QList< QModelIndex > mapListFromSource` (const `QList< QModelIndex > &sourceIndexes`) const
- `QList< QModelIndex > mapListToSource` (const `QList< QModelIndex > &indexes`) const
- `QModelIndex mapToSourceItemModel` (const `QModelIndex &index`) const
- void `setSourceFilterModel` (`ImageSortFilterModel *const model`)
- void `setSourceItemModel` (`ItemModel *const model`)
- `ImageSortFilterModel * sourceFilterModel` () const
- `ItemModel * sourceItemModel` () const

### Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- `DCategorizedSortFilterProxyModel` (`QObject *const parent=nullptr`)
- bool `isCategorizedModel` () const
- void `setCategorizedModel` (bool `categorizedModel`)
- void `setSortCategoriesByNaturalComparison` (bool `sortCategoriesByNaturalComparison`)
- void `sort` (int `column`, `Qt::SortOrder order=Qt::AscendingOrder`) override
- bool `sortCategoriesByNaturalComparison` () const
- int `sortColumn` () const
- `Qt::SortOrder sortOrder` () const

### Protected Member Functions

- virtual void `setDirectSourceItemModel` (`ItemModel *const model`)
- void `setSourceModel` (`QAbstractItemModel *const model`) override

*NOTE: made protected.*

### Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- virtual int `compareCategories` (const `QModelIndex &left`, const `QModelIndex &right`) const
- bool `lessThan` (const `QModelIndex &left`, const `QModelIndex &right`) const override
- virtual bool `subSortLessThan` (const `QModelIndex &left`, const `QModelIndex &right`) const

### Protected Attributes

- `ImageSortFilterModel * m_chainedModel` = nullptr

### Additional Inherited Members

### Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum `AdditionalRoles` { `CategoryDisplayRole` = 0x17CE990A , `CategorySortRole` = 0x27857E60 }

## 6.818.1 Member Function Documentation

### 6.818.1.1 imageFilterModel()

```
ItemFilterModel * Digikam::ImageSortFilterModel::imageFilterModel ( ) const [virtual]
```

Returns this, any chained [ItemFilterModel](#), or 0.

Reimplemented in [Digikam::ItemFilterModel](#).

### 6.818.1.2 imageInfosSorted()

```
QList< ItemInfo > Digikam::ImageSortFilterModel::imageInfosSorted ( ) const
```

Returns a list of all image infos, sorted according to this model. If you do not need a sorted list, use [ItemModel](#)'s [imageInfos\(\)](#) method.

### 6.818.1.3 mapListToSource()

```
QList< QModelIndex > Digikam::ImageSortFilterModel::mapListToSource (
    const QList< QModelIndex > & indexes ) const
```

Convenience methods mapped to [ItemModel](#). Mentioned indexes returned come from the source image model.

### 6.818.1.4 setDirectSourceItemModel()

```
void Digikam::ImageSortFilterModel::setDirectSourceItemModel (
    ItemModel *const model ) [protected], [virtual]
```

Reimplement if needed. Called only when model shall be set as (direct) sourceModel.

Reimplemented in [Digikam::ItemFilterModel](#).

## 6.819 Digikam::ImageTagChangeset Class Reference

### Public Types

- enum [Operation](#) {  
    **Unknown** , **Added** , **Moved** , **Removed** ,  
    **RemovedAll** , **PropertiesChanged** }

## Public Member Functions

- bool **containsImage** (qulonglong id) const
- bool **containsTag** (int id) const
- QList< qulonglong > **ids** () const
- **ImageTagChangeset** (const QList< qulonglong > &ids, const QList< int > &tags, [Operation](#) operation)
- **ImageTagChangeset** (qulonglong id, const QList< int > &tags, [Operation](#) operation)
- **ImageTagChangeset** (qulonglong id, int tag, [Operation](#) operation)
- [Operation](#) **operation** () const
- **ImageTagChangeset** & **operator<<** (const **ImageTagChangeset** &other)
- bool **propertiesWereChanged** () const
- QList< int > **tags** () const
- bool **tagsWereAdded** () const
- bool **tagsWereRemoved** () const

## 6.819.1 Member Enumeration Documentation

### 6.819.1.1 Operation

```
enum Digikam::ImageTagChangeset::Operation
```

An [ImageTagChangeset](#) covers adding and removing the association of a tag with an image. It is described by a list of affected image ids, a list of affected tags, and an operation. There is no guarantee that information in the database has actually been changed.

## 6.819.2 Member Function Documentation

### 6.819.2.1 operator<<()

```
ImageTagChangeset & Digikam::ImageTagChangeset::operator<< (
    const ImageTagChangeset & other )
```

Combines two [ImageTagChangesets](#). The operations shall not differ between the two sets; the operation is set to Unknown if it differs. This is especially not suitable for RemovedAll changesets.

## 6.820 Digikam::ImageTagProperty Class Reference

### Public Member Functions

- bool **isNull** () const

### Public Attributes

- qulonglong **imageId** = -1
- QString **property**
- int **tagId** = -1
- QString **value**

## 6.821 Digikam::ImageTagPropertyName Class Reference

### Static Public Member Functions

- static QLatin1String **autodetectedFace** ()
- static QLatin1String **autodetectedPerson** ()
- static QLatin1String **faceToTrain** ()
- static QLatin1String **ignoredFace** ()
- static QLatin1String **tagRegion** ()

## 6.822 Digikam::ImageWindow Class Reference

Inheritance diagram for Digikam::ImageWindow:



### Classes

- class [Private](#)



## Public Slots

- void **loadItemInfos** (const [ItemInfoList](#) &imageInfoList, const [ItemInfo](#) &imageInfoCurrent, const QString &caption)
- void **openImage** (const [ItemInfo](#) &info)
- void **slotAssignColorLabel** (int colorId)
- void **slotAssignPickLabel** (int pickId)
- void **slotAssignRating** (int rating)
- void **slotSetup** () override
- void **slotSetupChanged** ()
- void **slotSetupICC** () override

## Public Slots inherited from [Digikam::EditorWindow](#)

- void **slotSetup** () override=0
- virtual void **slotSetupICC** ()=0

## Signals

- void **loadCurrentLater** ()
- void **signalSavingDialogProgress** (float value)
- void **signalURLChanged** (const QUrl &url)

## Signals inherited from [Digikam::EditorWindow](#)

- void **signalNoCurrentItem** ()
- void **signalPreviewModeChanged** (int)
- void **signalSelectionChanged** (const QRect &)
- void **signalToolApplied** ()

## Public Member Functions

- [DInfoInterface](#) \* **infoInterface** ([DPluginAction](#) \*const ac) override
- bool **queryClose** () override
- void **toggleTag** (int tagID)
- [VersionManager](#) \* **versionManager** () const override

## Public Member Functions inherited from [Digikam::EditorWindow](#)

- bool **actionEnabledState** () const
- [EditorWindow](#) (const QString &name, QWidget \*const parent=nullptr)
- void **loadTool** ([EditorTool](#) \*const tool)
- void **registerExtraPluginsActions** (QString &dom) override

## Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- `QList< QAction * > allActions () const`
- `void cleanupActions ()`
- `QString configGroupName () const`
- `void createFullScreenAction (const QString &name)`
- `void createHelpActions (const QString &handbookSection, bool coreOptions=true)`
- `void createSettingsActions ()`
- `void createSidebarActions ()`
- `DXmlGuiWindow (QWidget *const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())`
- `bool fullScreenIsActive () const`
- `void readFullScreenSettings (const KConfigGroup &group)`
- `void registerPluginsActions ()`
- `void setConfigGroupName (const QString &name)`
- `void setFullScreenOptions (int options)`
- `void unminimizeAndActivateWindow ()`

## Static Public Member Functions

- `static ImageWindow * imageWindow ()`
- `static bool imageWindowCreated ()`

## Static Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- `static QAction * buildStdAction (StdActionType type, const QObject *const recvr, const char *const slot, QObject *const parent)`
- `static QString configFullScreenHideSideBarsEntry ()`
- `static QString configFullScreenHideStatusBarEntry ()`
- `static QString configFullScreenHideThumbBarEntry ()`
- `static QString configFullScreenHideToolBarsEntry ()`
- `static void restoreWindowSize (QWindow *const win, const KConfigGroup &group)`
- `static void saveWindowSize (QWindow *const win, KConfigGroup &group)`
- `static void setGoodDefaultWindowSize (QWindow *const win)`
- `static void setupIconTheme ()`

## Protected Member Functions

- `void closeEvent (QCloseEvent *e) override`
- `void dragMoveEvent (QDragMoveEvent *e) override`
- `void dropEvent (QDropEvent *e) override`
- `void showEvent (QShowEvent *e) override`

## Protected Member Functions inherited from [Digikam::EditorWindow](#)

- void **addServicesMenuForUrl** (const QUrl &url)
- void **applyColorManagementSettings** ()
- void **applyIOSettings** ()
- void **applyStandardSettings** ()
- bool **checkOverwrite** (const QUrl &url)
- bool **checkPermissions** (const QUrl &url)
- void **colorManage** ()
- [EditorStackView](#) \* **editorStackView** () const
- void **execSavingProgressDialog** ()
- [ExposureSettingsContainer](#) \* **exposureSettings** () const
- virtual void **finishSaving** (bool success)
- virtual void **moveFile** ()
- bool **moveLocalFile** (const QString &src, const QString &dest)
- void **movingSaveFileFinished** (bool successful)
- void **openWith** (const QUrl &url, QAction \*action)
- bool **promptForOverWrite** ()
- bool **promptUserDelete** (const QUrl &url)
- bool **promptUserSave** (const QUrl &url, SaveAskMode mode=AskIfNeeded, bool allowCancel=true)
- virtual void **readSettings** ()
- void **readStandardSettings** ()
- void **resetOrigin** ()
- void **resetOriginSwitchFile** ()
- [VersionFileOperation](#) **saveAsVersionFileOperation** (const QUrl &url, const QUrl &saveLocation, const QString &format)
- [VersionFileOperation](#) **saveInFormatVersionFileOperation** (const QUrl &url, const QString &format)
- virtual void **saveSettings** ()
- void **saveStandardSettings** ()
- [VersionFileOperation](#) **saveVersionFileOperation** (const QUrl &url, bool fork)
- void **setupContextMenu** ()
- void **setupSelectToolsAction** ()
- void **setupStandardActions** ()
- void **setupStandardConnections** ()
- void **setupStatusBar** ()
- [SidebarSplitter](#) \* **sidebarSplitter** () const
- void **startingSave** (const QUrl &url)
- bool **startingSaveAs** (const QUrl &url)
- bool **startingSaveCurrentVersion** (const QUrl &url)
- bool **startingSaveNewVersion** (const QUrl &url)
- bool **startingSaveNewVersionAs** (const QUrl &url)
- bool **startingSaveNewVersionInFormat** (const QUrl &url, const QString &format)
- virtual void **toggleActions** (bool val)
- void **toggleNonDestructiveActions** ()
- void **toggleStandardActions** (bool val)
- void **toggleToolActions** ([EditorTool](#) \*tool=nullptr)
- void **toggleZoomActions** (bool val)
- bool **waitForSavingToComplete** ()

## Protected Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- void **closeEvent** (QCloseEvent \*e) override
- void **editKeyboardShortcuts** (KActionCollection \*const extraac=nullptr, const QString &actitle=QString())
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override
- void **keyPressEvent** (QKeyEvent \*e) override
- QAction \* **showMenuBarAction** () const
- QAction \* **showStatusBarAction** () const

## Additional Inherited Members

### Public Types inherited from [Digikam::EditorWindow](#)

- enum **TransformType** { **RotateLeft** , **RotateRight** , **FlipHorizontal** , **FlipVertical** }

### Static Public Attributes inherited from [Digikam::EditorWindow](#)

- static const QString **CONFIG\_GROUP\_NAME**

### Protected Types inherited from [Digikam::EditorWindow](#)

- enum **SaveAskMode** {  
**AskIfNeeded** , **OverwriteWithoutAsking** , **AlwaysSaveAs** , **SaveVersionWithoutAsking** = **Overwrite**↔  
**WithoutAsking** ,  
**AlwaysNewVersion** = **AlwaysSaveAs** }

### Protected Slots inherited from [Digikam::EditorWindow](#)

- virtual bool **saveOrSaveAs** ()
- void **slotAboutToShowRedoMenu** ()
- void **slotAboutToShowUndoMenu** ()
- virtual void **slotAddedDroppedItems** (QDropEvent \*e)=0
- virtual void **slotBackward** ()=0
- virtual void **slotChanged** ()=0
- void **slotComponentsInfo** () override
- virtual void **slotContextMenu** ()=0
- virtual void **slotDeleteCurrentItem** ()=0
- virtual void **slotDiscardChanges** ()
- virtual void **slotFileOriginChanged** (const QString &filePath)
- virtual void **slotFileWithDefaultApplication** ()=0
- virtual void **slotFirst** ()=0
- virtual void **slotForward** ()=0
- virtual void **slotLast** ()=0
- virtual void **slotLoadingFinished** (const QString &filename, bool success)
- void **slotLoadingProgress** (const QString &filePath, float progress)
- virtual void **slotLoadingStarted** (const QString &filename)
- void **slotNameLabelCancelButtonPressed** ()
- virtual void **slotOpenOriginal** ()
- virtual void **slotOpenWith** (QAction \*action=nullptr)=0
- virtual void **slotPrepareToLoad** ()
- virtual void **slotRevert** ()=0
- void **slotSavingProgress** (const QString &filePath, float progress)
- virtual void **slotSavingStarted** (const QString &filename)
- void **slotSelected** (bool)
- virtual void **slotUpdateItemInfo** ()=0

### Protected Slots inherited from [Digikam::DXmlGuiWindow](#)

- bool **slotClose** ()

## Protected Attributes inherited from [Digikam::EditorWindow](#)

- bool `m_actionEnabledState` = false
- QAction \* `m_applyToolAction` = nullptr
- QAction \* `m_backwardAction` = nullptr
- QColor `m_bgColor`
- [Canvas](#) \* `m_canvas` = nullptr
- QAction \* `m_closeToolAction` = nullptr
- QMenu \* `m_contextMenu` = nullptr
- QAction \* `m_discardChangesAction` = nullptr
- bool `m_editingOriginalImage` = true
- QAction \* `m_exportAction` = nullptr
- QAction \* `m_fileDeleteAction` = nullptr
- QAction \* `m_firstAction` = nullptr
- QString `m_formatForRAWVersioning`
- QString `m_formatForSubversions`
- QAction \* `m_forwardAction` = nullptr
- [IOFileSettings](#) \* `m_IOFileSettings` = nullptr
- QAction \* `m_lastAction` = nullptr
- [StatusProgressBar](#) \* `m_nameLabel` = nullptr
- bool `m_nonDestructive` = true
- QAction \* `m_openVersionAction` = nullptr
- KToolBarPopupAction \* `m_redoAction` = nullptr
- [DAdjustableLabel](#) \* `m_resLabel` = nullptr
- QAction \* `m_revertAction` = nullptr
- QAction \* `m_saveAction` = nullptr
- QAction \* `m_saveAsAction` = nullptr
- QAction \* `m_saveCurrentVersionAction` = nullptr
- KToolBarPopupAction \* `m_saveNewVersionAction` = nullptr
- QAction \* `m_saveNewVersionAsAction` = nullptr
- QMenu \* `m_saveNewVersionInFormatAction` = nullptr
- [SavingContext](#) `m_savingContext`
- QPointer< QProgressDialog > `m_savingProgressDialog` = nullptr
- QAction \* `m_serviceAction` = nullptr
- QMenu \* `m_servicesMenu` = nullptr
- bool `m_setExifOrientationTag` = true
- QAction \* `m_showBarAction` = nullptr
- [SidebarSplitter](#) \* `m_splitter` = nullptr
- [EditorStackView](#) \* `m_stackView` = nullptr
- QVector< TransformType > `m_transformQue`  
*NOTE: using QVector to store transforms.*
- KToolBarPopupAction \* `m_undoAction` = nullptr

## Protected Attributes inherited from [Digikam::DXmlGuiWindow](#)

- [DLogoAction](#) \* `m_animLogo` = nullptr

### 6.822.1 Member Function Documentation

#### 6.822.1.1 infoface()

```
DInfoInterface * Digikam::ImageWindow::infoIface (
    DPluginAction *const ac ) [override], [virtual]
```

Return the interface instance to access to items information.

Implements [Digikam::DXmlGuiWindow](#).

### 6.822.1.2 versionManager()

`VersionManager * Digikam::ImageWindow::versionManager ( ) const [override], [virtual]`

Reimplemented from [Digikam::EditorWindow](#).

## 6.823 Digikam::ImageWindow::Private Class Reference

### Public Member Functions

- QModelIndex **currentIndex** () const
- bool **currentIsValid** () const
- QModelIndex **currentSourceIndex** () const
- QUrl **currentUrl** () const
- void **ensureModelContains** (const [ItemInfo](#) &info)
- QModelIndex **firstIndex** () const
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- QModelIndex **lastIndex** () const
- QModelIndex **nextIndex** () const
- QModelIndex **previousIndex** () const
- void **setThumbBarToCurrent** ()

### Public Attributes

- const QString **configShowThumbbarEntry** = QLatin1String("Show Thumbbar")
- [ItemInfo](#) **currentItemInfo**
- [ItemDragDropHandler](#) \* **dragDropHandler** = nullptr
- QList< [FaceTagsIface](#) > **facesList**
- QAction \* **fileDeletePermanentlyAction** = nullptr
- Delete actions.*
- QAction \* **fileDeletePermanentlyDirectlyAction** = nullptr
- QAction \* **fileTrashDirectlyAction** = nullptr
- [ItemFilterModel](#) \* **imageFilterModel** = nullptr
- [ItemListModel](#) \* **imageInfoModel** = nullptr
- [ItemPropertiesSideBarDB](#) \* **rightSideBar** = nullptr
- [ItemThumbnailBar](#) \* **thumbBar** = nullptr
- [ThumbBarDock](#) \* **thumbBarDock** = nullptr
- QAction \* **toMainWindowAction** = nullptr
- [DatabaseVersionManager](#) **versionManager**
- QMainWindow \* **viewContainer** = nullptr

## 6.824 Digikam::ImageZoomSettings Class Reference

### Public Types

- enum **FitToSizeMode** { **AlwaysFit** , **OnlyScaleDown** }

## Public Member Functions

- void [fitToSize](#) (const QSizeF &frameSize, FitToSizeMode=AlwaysFit)
- double [fitToSizeZoomFactor](#) (const QSizeF &frameSize, FitToSizeMode=AlwaysFit) const
- QSizeF [imageSize](#) () const
- **ImageZoomSettings** (const QSize &[imageSize](#), const QSize &originalSize=QSize())
- bool [isFitToSize](#) (const QSizeF &frameSize) const
- QPointF [mapImageToZoom](#) (const QPointF &imagePoint) const
- QRectF [mapImageToZoom](#) (const QRectF &imagePoint) const
- QPointF [mapZoomToImage](#) (const QPointF &zoomedPoint) const
- QRectF [mapZoomToImage](#) (const QRectF &imageRect) const
- QSizeF [originalImageSize](#) () const
- double [realZoomFactor](#) () const
- void [setDisplayWidget](#) (QWidget \*const widget)
- void [setImageSize](#) (const QSize &size, const QSize &originalSize=QSize())
- void [setZoomFactor](#) (double zoom)
- double [snappedZoomFactor](#) (double newZoom, const QSizeF &frameSize) const
- double [snappedZoomStep](#) (double nextZoom, const QSizeF &frameSize) const
- QRectF [sourceRect](#) (const QRectF &imageRect) const
- QSizeF [zoomedSize](#) () const
- double [zoomFactor](#) () const

## Static Public Member Functions

- static bool [getImageSmoothScale](#) ()
- static void [setImageSmoothScale](#) (bool enable)

## Protected Attributes

- QWidget \* [m\\_displayWidget](#) = nullptr
- QSizeF [m\\_size](#)
- double [m\\_zoom](#) = 1.0
- double [m\\_zoomConst](#) = 1.0

## 6.824.1 Member Function Documentation

### 6.824.1.1 fitToSize()

```
void Digikam::ImageZoomSettings::fitToSize (
    const QSizeF & frameSize,
    FitToSizeMode mode = AlwaysFit )
```

Sets the current zoom factor to the factor needed to fit the current (original) image size into the given view size. Aspect ratio will be respected, that means the frameSize may not be completely filled in one dimension, and [zoomedSize\(\)](#) can differ from frameSize in one dimension.

### 6.824.1.2 fitToSizeZoomFactor()

```
double Digikam::ImageZoomSettings::fitToSizeZoomFactor (
    const QSizeF & frameSize,
    FitToSizeMode mode = AlwaysFit ) const
```

Returns the zoom factor that would be used by [fitToSize\(\)](#) called with the given frameSize.

### 6.824.1.3 imageSize()

```
QSizeF Digikam::ImageZoomSettings::imageSize ( ) const
```

Returns the (available) image size

### 6.824.1.4 mapImageToZoom() [1/2]

```
QPointF Digikam::ImageZoomSettings::mapImageToZoom (
    const QPointF & imagePoint ) const
```

For a given point (in (0,0), [imageSize\(\)](#)) returns the corresponding point in (0,0),[zoomedSize\(\)](#).

### 6.824.1.5 mapImageToZoom() [2/2]

```
QRectF Digikam::ImageZoomSettings::mapImageToZoom (
    const QRectF & imagePoint ) const
```

For a given rect contained in ((0,0), [imageSize\(\)](#)) returns the corresponding rectangle in (0,0),[zoomedSize\(\)](#).

### 6.824.1.6 mapZoomToImage()

```
QPointF Digikam::ImageZoomSettings::mapZoomToImage (
    const QPointF & zoomedPoint ) const
```

For a given point (in (0,0), [zoomedSize\(\)](#)) returns the corresponding point in (0,0),[imageSize\(\)](#).

### 6.824.1.7 originalImageSize()

```
QSizeF Digikam::ImageZoomSettings::originalImageSize ( ) const
```

Return the original image size. Can be identical to size().

### 6.824.1.8 realZoomFactor()

```
double Digikam::ImageZoomSettings::realZoomFactor ( ) const
```

Return the real zoom factor dependent on device pixel ratio

### 6.824.1.9 setDisplayWidget()

```
void Digikam::ImageZoomSettings::setDisplayWidget (
    QWidget *const widget )
```

Set the graphics view widget to track the device pixel ratio.



#### 6.824.1.10 setImageSize()

```
void Digikam::ImageZoomSettings::setImageSize (
    const QSize & size,
    const QSize & originalSize = QSize() )
```

Sets the size of the (available) image data. Optionally, you can specify an original size, if the available image data is a scaled-down version. In this case, zoom factors will refer to the original size. The zoom factor is unchanged, you need to call `fitToSize` again.

#### 6.824.1.11 setImageSmoothScale()

```
void Digikam::ImageZoomSettings::setImageSmoothScale (
    bool enable ) [static]
```

Static functions to define the smooth scaling of the image.

#### 6.824.1.12 setZoomFactor()

```
void Digikam::ImageZoomSettings::setZoomFactor (
    double zoom )
```

Sets the current zoom factor, relative to (original) size.

#### 6.824.1.13 snappedZoomFactor()

```
double Digikam::ImageZoomSettings::snappedZoomFactor (
    double newZoom,
    const QSizeF & frameSize ) const
```

When setting a new zoom factor (absolute value), the new value may be very close to a special value. Returns this special value if this is the case, returns `newZoom` if not applicable.

#### 6.824.1.14 snappedZoomStep()

```
double Digikam::ImageZoomSettings::snappedZoomStep (
    double nextZoom,
    const QSizeF & frameSize ) const
```

When changing the zoom from current zoom to given `nextZoom`, sometimes a special value may be crossed, and this could then be used instead of `nextZoom`. Returns this special zoom, or `nextZoom` if not applicable.

#### 6.824.1.15 sourceRect()

```
QRectF Digikam::ImageZoomSettings::sourceRect (
    const QRectF & imageRect ) const
```

For a given rectangle contained in `((0,0), zoomedSize())` returns the corresponding rectangle in `(0,0), imageSize()`.

**6.824.1.16 zoomedSize()**

```
QSizeF Digikam::ImageZoomSettings::zoomedSize ( ) const
```

Return the size of the image when the current zoom factor is applied. This is the size the image should be displayed at.

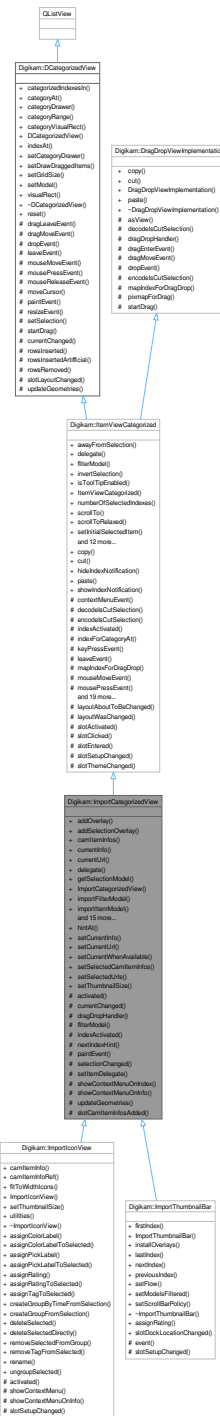
**6.824.1.17 zoomFactor()**

```
double Digikam::ImageZoomSettings::zoomFactor ( ) const
```

Return the currently set zoom factor

## 6.825 Digikam::ImportCategorizedView Class Reference

Inheritance diagram for Digikam::ImportCategorizedView:



### Public Slots

- void [hintAt](#) (const [CamItemInfo](#) &info)
- void [setCurrentInfo](#) (const [CamItemInfo](#) &info)

- void [setCurrentUrl](#) (const QUrl &url)
- void [setCurrentWhenAvailable](#) (qulonglong camlItemId)
- void [setSelectedCamlItemInfos](#) (const QList< [CamlItemInfo](#) > &infos)
- void [setSelectedUrls](#) (const QList< QUrl > &urlList)
- void [setThumbnailSize](#) (int size)

### Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void [copy](#) () override
- void [cut](#) () override
- void [hideIndexNotification](#) ()
- void [paste](#) () override
- void [showIndexNotification](#) (const QModelIndex &index, const QString &message)

### Public Slots inherited from [Digikam::DCategorizedView](#)

- void [reset](#) () override

### Signals

- void [camlItemInfoActivated](#) (const [CamlItemInfo](#) &info)
- void [currentChanged](#) (const [CamlItemInfo](#) &info)
- void [deselected](#) (const QList< [CamlItemInfo](#) > &nowDeselectedInfos)
- void [modelChanged](#) ()
- void [selected](#) (const QList< [CamlItemInfo](#) > &newSelectedInfos)

### Signals inherited from [Digikam::ItemViewCategorized](#)

- void [clicked](#) (const QMouseEvent \*e, const QModelIndex &index)
- void [entered](#) (const QMouseEvent \*e, const QModelIndex &index)
- void [keyPressed](#) (QKeyEvent \*e)
- void [selectionChanged](#) ()
- void [selectionCleared](#) ()
- void [viewportClicked](#) (const QMouseEvent \*e)
- void [zoomInStep](#) ()
- void [zoomOutStep](#) ()

### Public Member Functions

- void [addOverlay](#) ([ItemDelegateOverlay](#) \*overlay, [ImportDelegate](#) \*delegate=nullptr)
- void [addSelectionOverlay](#) ([ImportDelegate](#) \*delegate=nullptr)
- QList< [CamlItemInfo](#) > [camlItemInfos](#) () const
- [CamlItemInfo](#) [currentInfo](#) () const
- QUrl [currentUrl](#) () const
- [ImportDelegate](#) \* [delegate](#) () const
- QItemSelectionModel \* [getSelectionModel](#) () const
- [ImportCategorizedView](#) (QWidget \*const parent=nullptr)
- [ImportFilterModel](#) \* [importFilterModel](#) () const
- [ImportItemModel](#) \* [importItemModel](#) () const
- [ImportSortFilterModel](#) \* [importSortFilterModel](#) () const

- [ImportThumbnailModel](#) \* [importThumbnailModel](#) () const
- [CamItemInfo](#) [nextInfo](#) (const [CamItemInfo](#) &info)
- [CamItemInfo](#) [nextInOrder](#) (const [CamItemInfo](#) &startingPoint, int nth)
- [CamItemInfo](#) [previousInfo](#) (const [CamItemInfo](#) &info)
- void [removeOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- QList< [CamItemInfo](#) > [selectedCamItemInfos](#) () const
- QList< [CamItemInfo](#) > [selectedCamItemInfosCurrentFirst](#) () const
- QList< QUrl > [selectedUrls](#) () const
- void [setModels](#) ([ImportItemModel](#) \*model, [ImportSortFilterModel](#) \*filterModel)
- virtual void [setThumbnailSize](#) (const [ThumbnailSize](#) &size)
- [ThumbnailSize](#) [thumbnailSize](#) () const
- void [toIndex](#) (const QUrl &url)
- QList< QUrl > [urls](#) () const

## Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void [awayFromSelection](#) ()
- [DItemDelegate](#) \* [delegate](#) () const
- void [invertSelection](#) ()
- bool [isToolTipEnabled](#) () const
- [ItemViewCategorized](#) (QWidget \*const parent=nullptr)
- int [numberOfSelectedIndexes](#) () const
- void [scrollTo](#) (const QModelIndex &index, ScrollHint hint=EnsureVisible) override
- void [scrollToRelaxed](#) (const QModelIndex &index, ScrollHint hint=EnsureVisible)
- void [setInitialSelectedItem](#) (bool enabled)
- void [setScrollCurrentToCenter](#) (bool enabled)
- void [setScrollStepGranularity](#) (int factor)
- void [setSelectedIndexes](#) (const QList< QModelIndex > &indexes)
- void [setSpacing](#) (int spacing)
- void [setToolTipEnabled](#) (bool enabled)
- void [setUsePointingHandCursor](#) (bool useCursor)
- void [toFirstIndex](#) ()
- void [toIndex](#) (const QModelIndex &index)
- void [toLastIndex](#) ()
- void [toNextIndex](#) ()
- void [toPreviousIndex](#) ()

## Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual QModelIndexList [categorizedIndexesIn](#) (const QRect &rect) const
- virtual QModelIndex [categoryAt](#) (const QPoint &point) const
- [DCategoryDrawer](#) \* [categoryDrawer](#) () const
- virtual QItemSelectionRange [categoryRange](#) (const QModelIndex &index) const
- virtual QRect [categoryVisualRect](#) (const QModelIndex &index) const
- [DCategorizedView](#) (QWidget \*const parent=nullptr)
- QModelIndex [indexAt](#) (const QPoint &point) const override
- void [setCategoryDrawer](#) ([DCategoryDrawer](#) \*categoryDrawer)
- void [setDrawDraggedItems](#) (bool drawDraggedItems)
- void [setGridSize](#) (const QSize &size)
- void [setModel](#) (QAbstractItemModel \*model) override
- QRect [visualRect](#) (const QModelIndex &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

## Protected Slots

- void **slotCamItemInfosAdded** ()

## Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void **layoutAboutToBeChanged** ()
- void **layoutWasChanged** ()
- void **slotActivated** (const QModelIndex &index)
- void **slotClicked** (const QModelIndex &index)
- void **slotEntered** (const QModelIndex &index)
- virtual void **slotSetupChanged** ()
- virtual void **slotThemeChanged** ()

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## Protected Member Functions

- virtual void **activated** (const [CamItemInfo](#) &info, Qt::KeyboardModifiers modifiers)  
*Reimplement these in a subclass.*
- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- [QSortFilterProxyModel](#) \* **filterModel** () const override  
*reimplemented from parent class*
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- [QModelIndex](#) **nextIndexHint** (const QModelIndex &indexToAnchor, const [QItemSelectionRange](#) &removed) const override
- void **paintEvent** ([QPaintEvent](#) \*e) override
- void **selectionChanged** (const [QItemSelection](#) &, const [QItemSelection](#) &) override
- void **setItemDelegate** ([ImportDelegate](#) \*delegate)
- void **showContextMenuOnIndex** ([QContextMenuEvent](#) \*event, const QModelIndex &index) override  
*Reimplement these in a subclass.*
- virtual void **showContextMenuOnInfo** ([QContextMenuEvent](#) \*event, const [CamItemInfo](#) &info)
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** (DItemDelegate \*delegate)
- void **setToolTip** (ItemViewToolTip \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

## Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

## 6.825.1 Member Function Documentation

### 6.825.1.1 activated()

```
void Digikam::ImportCategorizedView::activated (
    const CamItemInfo & info,
    Qt::KeyboardModifiers modifiers ) [protected], [virtual]
```

Reimplemented in [Digikam::ImportIconView](#).

### 6.825.1.2 addOverlay()

```
void Digikam::ImportCategorizedView::addOverlay (
    ItemDelegateOverlay * overlay,
    ImportDelegate * delegate = nullptr )
```

Add and remove an overlay. It will as well be removed automatically when destroyed. Unless you pass a different delegate, the current delegate will be used.

### 6.825.1.3 camItemInfoActivated

```
void Digikam::ImportCategorizedView::camItemInfoActivated (
    const CamItemInfo & info ) [signal]
```

Emitted when the given [CamItemInfo](#) is activated. Info is never null.

### 6.825.1.4 deselected

```
void Digikam::ImportCategorizedView::deselected (
    const QList< CamItemInfo > & nowDeselectedInfos ) [signal]
```

Emitted when items are deselected. There may be other selected infos left. This signal is not emitted when the model is reset; then only selectionCleared is emitted.

### 6.825.1.5 dragDropHandler()

```
AbstractItemDragDropHandler * Digikam::ImportCategorizedView::dragDropHandler ( ) const [override],
[protected], [virtual]
```

You need to implement these three methods Returns the drag drop handler.

Implements [Digikam::DragDropViewImplementation](#).

### 6.825.1.6 filterModel()

```
QSortFilterProxyModel * Digikam::ImportCategorizedView::filterModel ( ) const [override],
[protected], [virtual]
```

Implements [Digikam::ItemViewCategorized](#).



### 6.825.1.7 hintAt

```
void Digikam::ImportCategorizedView::hintAt (
    const CamItemInfo & info ) [slot]
```

Does something to gain attention for info, but not changing current selection

### 6.825.1.8 importFilterModel()

```
ImportFilterModel * Digikam::ImportCategorizedView::importFilterModel ( ) const
```

Returns any [ImportFilterModel](#) in chain. May not be sourceModel()

### 6.825.1.9 importThumbnailModel()

```
ImportThumbnailModel * Digikam::ImportCategorizedView::importThumbnailModel ( ) const
```

Returns 0 if the [ImportItemModel](#) is not an [ImportThumbnailModel](#)

### 6.825.1.10 indexActivated()

```
void Digikam::ImportCategorizedView::indexActivated (
    const QModelIndex & index,
    Qt::KeyboardModifiers modifiers ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

### 6.825.1.11 modelChanged

```
void Digikam::ImportCategorizedView::modelChanged ( ) [signal]
```

Emitted when a new model is set

### 6.825.1.12 nextIndexHint()

```
QModelIndex Digikam::ImportCategorizedView::nextIndexHint (
    const QModelIndex & indexToAnchor,
    const QItemSelectionRange & removed ) const [override], [protected], [virtual]
```

Assuming the given indexes would be removed (hypothetically!), return the index to be selected instead, starting from anchor. The default implementation returns the next remaining sibling.

Reimplemented from [Digikam::ItemViewCategorized](#).

#### 6.825.1.13 nextInOrder()

```
CamItemInfo Digikam::ImportCategorizedView::nextInOrder (
    const CamItemInfo & startingPoint,
    int nth )
```

Returns the n-th info after the given one. Specifically, return the previous info for  $nth = -1$  and the next info for  $n = 1$ . Returns a null info if either startingPoint or the nth info are not contained in the model

#### 6.825.1.14 selected

```
void Digikam::ImportCategorizedView::selected (
    const QList< CamItemInfo > & newSelectedInfos ) [signal]
```

Emitted when new items are selected. The parameter includes only the newly selected infos, there may be other already selected infos.

#### 6.825.1.15 setCurrentInfo

```
void Digikam::ImportCategorizedView::setCurrentInfo (
    const CamItemInfo & info ) [slot]
```

Set as current item the item identified by the [CamItemInfo](#)

#### 6.825.1.16 setCurrentUrl

```
void Digikam::ImportCategorizedView::setCurrentUrl (
    const QUrl & url ) [slot]
```

Set as current item the item identified by its file url

#### 6.825.1.17 setCurrentWhenAvailable

```
void Digikam::ImportCategorizedView::setCurrentWhenAvailable (
    qulonglong camItemId ) [slot]
```

Scroll the view to the given item when it becomes available

#### 6.825.1.18 setSelectedCamItemInfos

```
void Digikam::ImportCategorizedView::setSelectedCamItemInfos (
    const QList< CamItemInfo > & infos ) [slot]
```

Set selected items

### 6.825.1.19 setSelectedUrls

```
void Digikam::ImportCategorizedView::setSelectedUrls (
    const QList< QUrl > & urlList ) [slot]
```

Set selected items identified by their file urls

### 6.825.1.20 showContextMenuOnIndex()

```
void Digikam::ImportCategorizedView::showContextMenuOnIndex (
    QContextMenuEvent * event,
    const QModelIndex & index ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

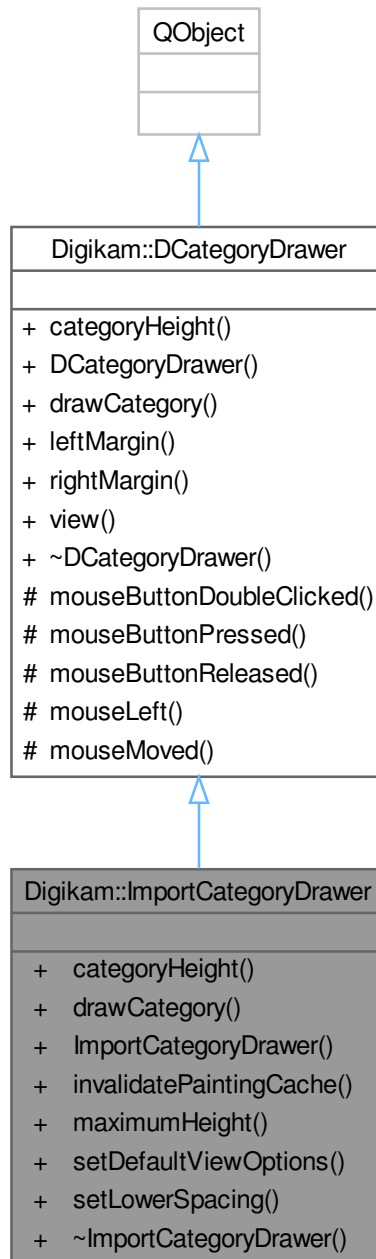
### 6.825.1.21 toIndex()

```
void Digikam::ImportCategorizedView::toIndex (
    const QUrl & url )
```

Selects the index as current and scrolls to it

## 6.826 Digikam::ImportCategoryDrawer Class Reference

Inheritance diagram for Digikam::ImportCategoryDrawer:



### Public Member Functions

- int `categoryHeight` (const `QModelIndex` &index, const `QStyleOption` &option) const override
- void `drawCategory` (const `QModelIndex` &index, int sortRole, const `QStyleOption` &option, `QPainter` \*painter) const override

- **ImportCategoryDrawer** ([ImportCategorizedView](#) \*const parent)
- void **invalidatePaintingCache** ()
- virtual int **maximumHeight** () const
- void **setDefaultViewOptions** (const [QStyleOptionViewItem](#) &option)
- void **setLowerSpacing** (int spacing)

## Public Member Functions inherited from [Digikam::DCategoryDrawer](#)

- [DCategoryDrawer](#) ([DCategoryView](#) \*const view)
- virtual int **leftMargin** () const
- virtual int **rightMargin** () const
- [DCategoryView](#) \* view () const

## Additional Inherited Members

## Signals inherited from [Digikam::DCategoryDrawer](#)

- void [actionRequested](#) (int action, const [QModelIndex](#) &index)
- void [collapseOrExpandClicked](#) (const [QModelIndex](#) &index)

## Protected Member Functions inherited from [Digikam::DCategoryDrawer](#)

- virtual void [mouseButtonDoubleClicked](#) (const [QModelIndex](#) &index, const [QRect](#) &blockRect, [QMouseEvent](#) \*event)
- virtual void [mouseButtonPressed](#) (const [QModelIndex](#) &index, const [QRect](#) &blockRect, [QMouseEvent](#) \*event)
- virtual void [mouseButtonReleased](#) (const [QModelIndex](#) &index, const [QRect](#) &blockRect, [QMouseEvent](#) \*event)
- virtual void [mouseLeft](#) (const [QModelIndex](#) &index, const [QRect](#) &blockRect)
- virtual void [mouseMoved](#) (const [QModelIndex](#) &index, const [QRect](#) &blockRect, [QMouseEvent](#) \*event)

## 6.826.1 Member Function Documentation

### 6.826.1.1 categoryHeight()

```
int Digikam::ImportCategoryDrawer::categoryHeight (
    const QModelIndex & index,
    const QStyleOption & option ) const [override], [virtual]
```

#### Returns

The category height for the category represented by index `index` with style options `option`.

Reimplemented from [Digikam::DCategoryDrawer](#).

### 6.826.1.2 drawCategory()

```
void Digikam::ImportCategoryDrawer::drawCategory (
    const QModelIndex & index,
    int sortRole,
    const QStyleOption & option,
    QPainter * painter ) const [override], [virtual]
```

This method purpose is to draw a category represented by the given

## Parameters

<i>index</i>	with the given
<i>sortRole</i>	sorting role
<i>option</i>	painter style options
<i>painter</i>	painter instance

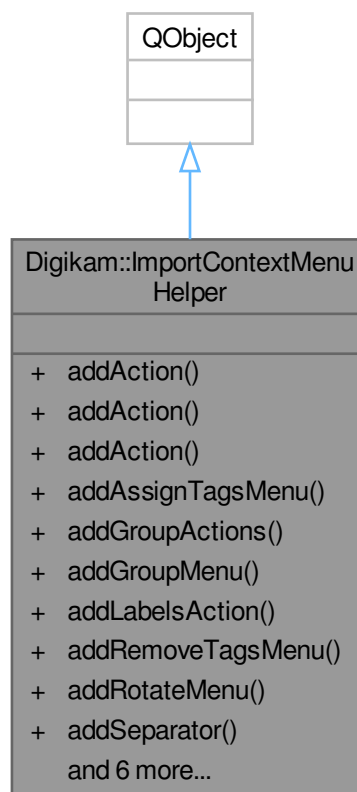
## Note

This method will be called one time per category, always with the first element in that category

Reimplemented from [Digikam::DCategoryDrawer](#).

## 6.827 Digikam::ImportContextMenuHelper Class Reference

Inheritance diagram for Digikam::ImportContextMenuHelper:



## Public Types

- typedef const QList< qlonglong > **itemIds**

## Signals

- void **signalAddNewTagFromABCMenu** (const QString &)
- void **signalAssignColorLabel** (int)
- void **signalAssignPickLabel** (int)
- void **signalAssignRating** (int)

## Public Member Functions

- void [addAction](#) (const QString &name, bool addDisabled=false)
- void [addAction](#) (QAction \*action, bool addDisabled=false)
- void [addAction](#) (QAction \*action, QObject \*recv, const char \*slot, bool addDisabled=false)
- void [addAssignTagsMenu](#) (itemIds &ids)
- void **addGroupActions** (itemIds &ids)
- void [addGroupMenu](#) (itemIds &ids)
- void [addLabelsAction](#) ()
- void [addRemoveTagsMenu](#) (itemIds &ids)
- void [addRotateMenu](#) (itemIds &ids)
- void [addSeparator](#) ()
- void [addServicesMenu](#) (const QList< QUrl > &selectedItems)
- void [addSubMenu](#) (QMenu \*subMenu)
- QAction \* [exec](#) (const QPoint &pos, QAction \*at=nullptr)
- [ImportContextMenuHelper](#) (QMenu \*const parent, KActionCollection \*const actionCollection=nullptr)
- void [setImportFilterModel](#) ([ImportFilterModel](#) \*model)

## 6.827.1 Constructor & Destructor Documentation

### 6.827.1.1 ImportContextMenuHelper()

```
Digikam::ImportContextMenuHelper::ImportContextMenuHelper (
    QMenu *const parent,
    KActionCollection *const actionCollection = nullptr ) [explicit]
```

Constructs the helper class.

#### Parameters

<i>parent</i>	the menu the helper class is linked to
<i>actionCollection</i>	the actionCollection that should be used. If not set, the standard action from <a href="#">DigikamApp</a> is used

## 6.827.2 Member Function Documentation

### 6.827.2.1 addAction() [1/3]

```
void Digikam::ImportContextMenuHelper::addAction (
    const QString & name,
    bool addDisabled = false )
```

Add an action from the actionCollection.

This method adds actions from the actionCollection. The actionCollection can be set in the constructor of the [ImportContextMenuHelper](#) class.

#### Parameters

<i>name</i>	the name of the action in the actionCollection
<i>addDisabled</i>	if set, disabled actions are added to the menu

### 6.827.2.2 addAction() [2/3]

```
void Digikam::ImportContextMenuHelper::addAction (
    QAction * action,
    bool addDisabled = false )
```

Add a temporary action.

Sometimes it is necessary to define actions that only exist in the current context menu content. Use this method to add such an action.

#### Parameters

<i>action</i>	the action to add
<i>addDisabled</i>	if set, disabled actions are added to the menu

### 6.827.2.3 addAction() [3/3]

```
void Digikam::ImportContextMenuHelper::addAction (
    QAction * action,
    QObject * recv,
    const char * slot,
    bool addDisabled = false )
```

Add a temporary action and assign it to a custom slot.

Use this method if you want to add a temporary action and immediately connect it to the receiving slot.

#### Parameters

<i>action</i>	the action to add
<i>recv</i>	the receiver of the triggered action
<i>slot</i>	the slot to connect the triggered action to
<i>addDisabled</i>	if set, disabled actions are added to the menu

### 6.827.2.4 addAssignTagsMenu()

```
void Digikam::ImportContextMenuHelper::addAssignTagsMenu (
    itemIds & ids )
```



Add actions to add, remove or edit a tag. The tag modification helper is used to execute the action. You must set the parent tag to use on modification helper. Add "Assign Tags" menu.

This menu will provide a list of all tags available so that they can be assigned to the current selected items.

To make this menu work, you need to run `exec()` from this class, otherwise the signals are not emitted and you will not be able to react on triggered actions from this menu. Make sure to connect the signals to the appropriate slots in the context menu handling method.

#### Parameters

<code>ids</code>	the selected items
------------------	--------------------

#### See also

[exec\(\)](#)  
[signalAssignTag\(\)](#)

#### 6.827.2.5 addGroupMenu()

```
void Digikam::ImportContextMenuHelper::addGroupMenu (
    itemIds & ids )
```

Add a "Group" menu. This menu will provide actions open, close, add to, remove from, or split a group.

`addGroupActions` will add the actions as a flat list, not in a submenu. Note: Call `setItemFilterModel` before to have Open/Close group actions.

#### 6.827.2.6 addLabelsAction()

```
void Digikam::ImportContextMenuHelper::addLabelsAction ( )
```

Add "Pick/Color/Rating Labels" action.

This action will provide methods to assign pick/color/rating labels to the currently selected items.

To make this menu work, you need to run `exec()` from this class, otherwise the signals are not emitted and you will not be able to react on triggered actions from this menu. Make sure to connect the signals to the appropriate slots in the context menu handling method.

#### See also

[exec\(\)](#)  
[signalAssignPickLabel\(\)](#)  
[signalAssignColorLabel\(\)](#)  
[signalAssignRating\(\)](#)

#### 6.827.2.7 addRemoveTagsMenu()

```
void Digikam::ImportContextMenuHelper::addRemoveTagsMenu (
    itemIds & ids )
```

Add "Remove Tags" menu.

This menu will provide a list of all tags assigned to the current items. Actions triggered in here will remove the selected tag from the items.

To make this menu work, you need to run `exec()` from this class, otherwise the signals are not emitted and you will not be able to react on triggered actions from this menu. Make sure to connect the signals to the appropriate slots in the context menu handling method.

**Parameters**

<i>ids</i>	the selected items
------------	--------------------

**See also**

[exec\(\)](#)  
[signalRemoveTag\(\)](#)

**6.827.2.8 addRotateMenu()**

```
void Digikam::ImportContextMenuHelper::addRotateMenu (
    itemIds & ids )
```

Add a menu to rotate item.

**Parameters**

<i>ids</i>	the selected items
------------	--------------------

**6.827.2.9 addSeparator()**

```
void Digikam::ImportContextMenuHelper::addSeparator ( )
```

Add a separator to the context menu

**6.827.2.10 addServicesMenu()**

```
void Digikam::ImportContextMenuHelper::addServicesMenu (
    const QList< QUrl > & selectedItems )
```

Add the services menu to the menu.

The services menu is used to open the selected items in a different application. It will query the item for registered services and provide them in a submenu. The menu will be titled "Open With...".

**Parameters**

<i>selectedItems</i>	the list of selected items
----------------------	----------------------------

**6.827.2.11 addSubMenu()**

```
void Digikam::ImportContextMenuHelper::addSubMenu (
    QMenu * subMenu )
```

Add a submenu to the parent context menu.

**Parameters**

<i>subMenu</i>	the submenu to be added
----------------	-------------------------

**6.827.2.12 exec()**

```
QAction * Digikam::ImportContextMenuHelper::exec (
    const QPoint & pos,
    QAction * at = nullptr )
```

Execute the registered parent menu and evaluate the triggered actions.

Always use this method instead the one from the parent menu. It will ensure that the signals are emitted and special cases are handled.

**Parameters**

<i>pos</i>	position of the triggered action in the registered menu
<i>at</i>	the action that should be at the position pos

**Returns**

the triggered action

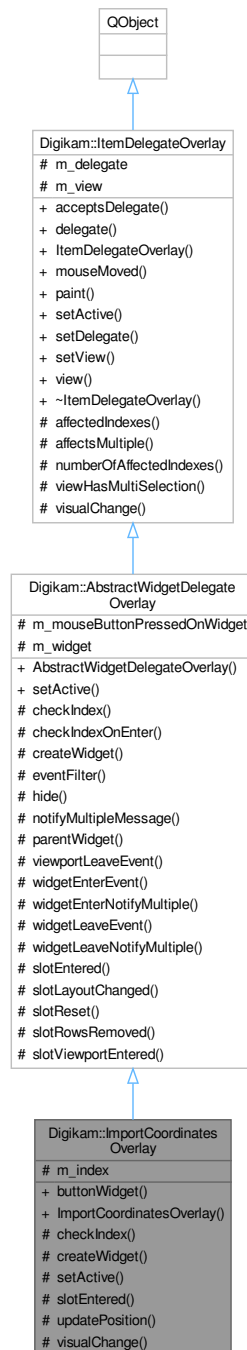
**6.827.2.13 setImportFilterModel()**

```
void Digikam::ImportContextMenuHelper::setImportFilterModel (
    ImportFilterModel * model )
```

Set a filter model. Some of the group actions will operate directly on the model.

## 6.828 Digikam::ImportCoordinatesOverlay Class Reference

Inheritance diagram for Digikam::ImportCoordinatesOverlay:



### Public Member Functions

- [ImportOverlayWidget](#) \* **buttonWidget** () const
- **ImportCoordinatesOverlay** (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- QWidget \* **createWidget** () override
- void **setActive** (bool active) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **visualChange** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **checkIndexOnEnter** (const QModelIndex &index) const
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual void **hide** ()
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* **parentWidget** () const
- virtual void **viewportLeaveEvent** (QObject \*obj, QEvent \*event)
- virtual void **widgetEnterEvent** ()
- void **widgetEnterNotifyMultiple** (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool **affectsMultiple** (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool **viewHasMultiSelection** () const

## Protected Attributes

- QPersistentModelIndex **m\_index**

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **m\_mouseButtonPressedOnWidget** = false
- QWidget \* **m\_widget** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* **m\_delegate** = nullptr
- QAbstractItemView \* **m\_view** = nullptr

## Additional Inherited Members

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

### 6.828.1 Member Function Documentation

#### 6.828.1.1 checkIndex()

```
bool Digikam::ImportCoordinatesOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

#### 6.828.1.2 createWidget()

```
QWidget * Digikam::ImportCoordinatesOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.828.1.3 setActive()

```
void Digikam::ImportCoordinatesOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.828.1.4 slotEntered()

```
void Digikam::ImportCoordinatesOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.828.1.5 visualChange()

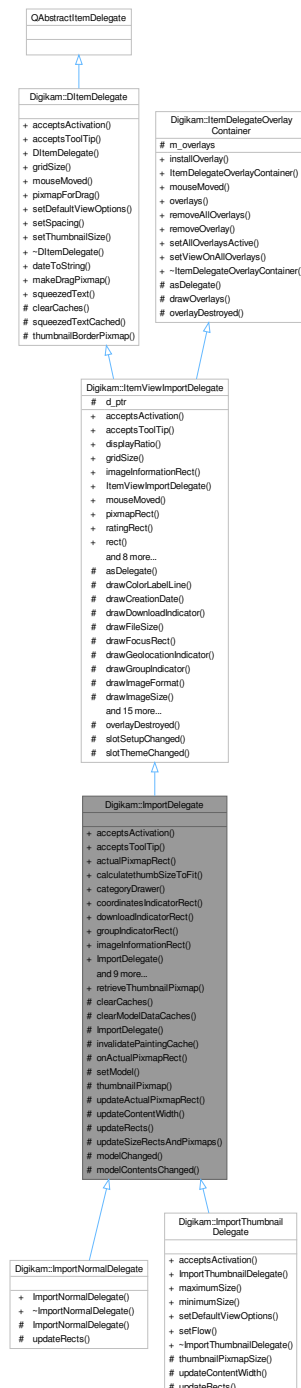
```
void Digikam::ImportCoordinatesOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

## 6.829 Digikam::ImportDelegate Class Reference

Inheritance diagram for Digikam::ImportDelegate:



### Classes

- class [ImportDelegatePrivate](#)



**Public Member Functions**

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect **actualPixmapRect** (const QModelIndex &index) const
- int **calculatethumbSizeToFit** (int ws)
- [ImportCategoryDrawer](#) \* **categoryDrawer** () const
- QRect **coordinatesIndicatorRect** () const
- QRect **downloadIndicatorRect** () const
- QRect **groupIndicatorRect** () const
- QRect [imageInformationRect](#) () const override
- **ImportDelegate** (QWidget \*const parent)
- QRect **lockIndicatorRect** () const
- void **paint** (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setSpacing](#) (int spacing) override
- void **setView** ([ImportCategorizedView](#) \*view)
- QRect **tagsRect** () const

**Public Member Functions inherited from [Digikam::ItemViewImportDelegate](#)**

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double **displayRatio** () const
- QSize [gridSize](#) () const override
- **ItemViewImportDelegate** (QWidget \*const parent)
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect [ratingRect](#) () const
- QRect **rect** () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setRatingEdited](#) (const QModelIndex &index)
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from [DItemDelegate](#)*
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int **spacing** () const
- [ThumbnailSize](#) **thumbnailSize** () const

**Public Member Functions inherited from [Digikam::DItemDelegate](#)**

- **DItemDelegate** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Static Public Member Functions

- static QPixmap **retrieveThumbnailPixmap** (const QModelIndex &index, int thumbnailSize)

## Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

## Protected Slots

- void **modelChanged** ()
- void **modelContentsChanged** ()

## Protected Slots inherited from [Digikam::ItemViewImportDelegate](#)

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

## Protected Member Functions

- void **clearCaches** () override
- virtual void **clearModelDataCaches** ()
- **ImportDelegate** ([ImportDelegate::ImportDelegatePrivate](#) &dd, QWidget \*const parent)
- void **invalidatePaintingCache** () override  
*reimplement these in subclasses*
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- virtual void **updateContentWidth** ()
- virtual void **updateRects** ()=0
- void **updateSizeRectsAndPixmaps** () override

## Protected Member Functions inherited from [Digikam::ItemViewImportDelegate](#)

- `QAbstractItemDelegate * asDelegate ()` override  
*Returns the delegate, typically, the derived class.*
- void `drawColorLabelLine` (QPainter \*p, const QRect &pixRect, int colorId) const
- void `drawCreationDate` (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void `drawDownloadIndicator` (QPainter \*p, const QRect &r, int itemType) const
- void `drawFileSize` (QPainter \*p, const QRect &r, qlonglong bytes) const
- void `drawFocusRect` (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void `drawGeolocationIndicator` (QPainter \*p, const QRect &r) const
- void `drawGroupIndicator` (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
- void `drawImageFormat` (QPainter \*p, const QRect &dimsRect, const QString &mime) const
- void `drawImageSize` (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void `drawLockIndicator` (QPainter \*p, const QRect &r, int lockStatus) const
- void `drawMouseOverRect` (QPainter \*p, const QStyleOptionViewItem &option) const
- void `drawName` (QPainter \*p, const QRect &nameRect, const QString &name) const
- void `drawPickLabelIcon` (QPainter \*p, const QRect &r, int pickLabel) const
- void `drawRating` (QPainter \*p, const QModelIndex &index, const QRect &[ratingRect](#), int rating, bool is← Selected) const
- void `drawTags` (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const
- QRect `drawThumbnail` (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail) const  
*Use the tool methods for painting in subclasses.*
- `ItemViewImportDelegate` (`ItemViewImportDelegatePrivate` &dd, QWidget \*const parent)
- void `prepareBackground` ()
- void `prepareFonts` ()
- void `prepareMetrics` (int maxWidth)
- void `prepareRatingPixmap` (bool composeOverBackground=true)
- QPixmap `ratingPixmap` (int rating, bool selected) const  
*Returns the relevant pixmap from the cached rating pixmaps.*

## Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString `squeezedTextCached` (QPainter \*const p, int width, const QString &text) const
- QPixmap `thumbnailBorderPixmap` (const QSize &pixSize, bool isGrouped=false) const

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void `drawOverlays` (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void `overlayDestroyed` (QObject \*o)  
*Declare as slot in the derived class calling this method.*

## Additional Inherited Members

## Signals inherited from [Digikam::ItemViewImportDelegate](#)

- void `hideNotification` ()
- void `requestNotification` (const QModelIndex &index, const QString &message)

## Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Protected Attributes inherited from [Digikam::ItemViewImportDelegate](#)

- [ItemViewImportDelegatePrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > **m\_overlays**

## 6.829.1 Member Function Documentation

### 6.829.1.1 acceptsActivation()

```
bool Digikam::ImportDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect = nullptr ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

### 6.829.1.2 acceptsToolTip()

```
bool Digikam::ImportDelegate::acceptsToolTip (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * tooltipRect = nullptr ) const [override], [virtual]
```

These methods take four parameters: The position on viewport, the rect on viewport, the index, and optionally a parameter into which, if the return value is true, a rectangle can be written for which the return value will be true as well.

Implements [Digikam::DItemDelegate](#).

### 6.829.1.3 clearCaches()

```
void Digikam::ImportDelegate::clearCaches ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::DItemDelegate](#).

#### 6.829.1.4 clearModelDataCaches()

```
void Digikam::ImportDelegate::clearModelDataCaches ( ) [protected], [virtual]
```

Reimplement to clear caches based on model indexes (hash on row number etc.) Change signals are listened to this is called whenever such properties become invalid.

#### 6.829.1.5 imageInformationRect()

```
QRect Digikam::ImportDelegate::imageInformationRect ( ) const [override], [virtual]
```

Returns the area where the image information is drawn, or null if empty / not supported. The image information is textual or graphical information, but not the pixmap. The [ratingRect\(\)](#) will e.g. typically be contained in this area.

Reimplemented from [Digikam::ItemViewImportDelegate](#).

#### 6.829.1.6 invalidatePaintingCache()

```
void Digikam::ImportDelegate::invalidatePaintingCache ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewImportDelegate](#).

#### 6.829.1.7 pixmapForDrag()

```
QPixmap Digikam::ImportDelegate::pixmapForDrag (
    const QStyleOptionViewItem & option,
    const QList< QModelIndex > & indexes ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

#### 6.829.1.8 pixmapRect()

```
QRect Digikam::ImportDelegate::pixmapRect ( ) const [override], [virtual]
```

Returns the area where the pixmap is drawn, or null if not supported

Reimplemented from [Digikam::ItemViewImportDelegate](#).

#### 6.829.1.9 retrieveThumbnailPixmap()

```
QPixmap Digikam::ImportDelegate::retrieveThumbnailPixmap (
    const QModelIndex & index,
    int thumbnailSize ) [static]
```

Retrieve the thumbnail pixmap in given size for the [ImportItemModel::ThumbnailRole](#) for the given index from the given index, which must adhere to [ImportThumbnailModel](#) semantics.

#### 6.829.1.10 `setDefaultViewOptions()`

```
void Digikam::ImportDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. `option.rect` shall be the viewport rectangle. Call on resize, font change.

Implements [Digikam::DItemDelegate](#).

Reimplemented in [Digikam::ImportThumbnailDelegate](#).

#### 6.829.1.11 `setSpacing()`

```
void Digikam::ImportDelegate::setSpacing (
    int spacing ) [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

#### 6.829.1.12 `updateContentWidth()`

```
void Digikam::ImportDelegate::updateContentWidth ( ) [protected], [virtual]
```

Reimplement this to set `contentWidth`. This is the maximum width of all content rectangles, typically excluding margins on both sides.

Reimplemented in [Digikam::ImportThumbnailDelegate](#).

#### 6.829.1.13 `updateRects()`

```
virtual void Digikam::ImportDelegate::updateRects ( ) [protected], [pure virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The `paint()` method operates depending on these rects.

Implemented in [Digikam::ImportThumbnailDelegate](#), and [Digikam::ImportNormalDelegate](#).

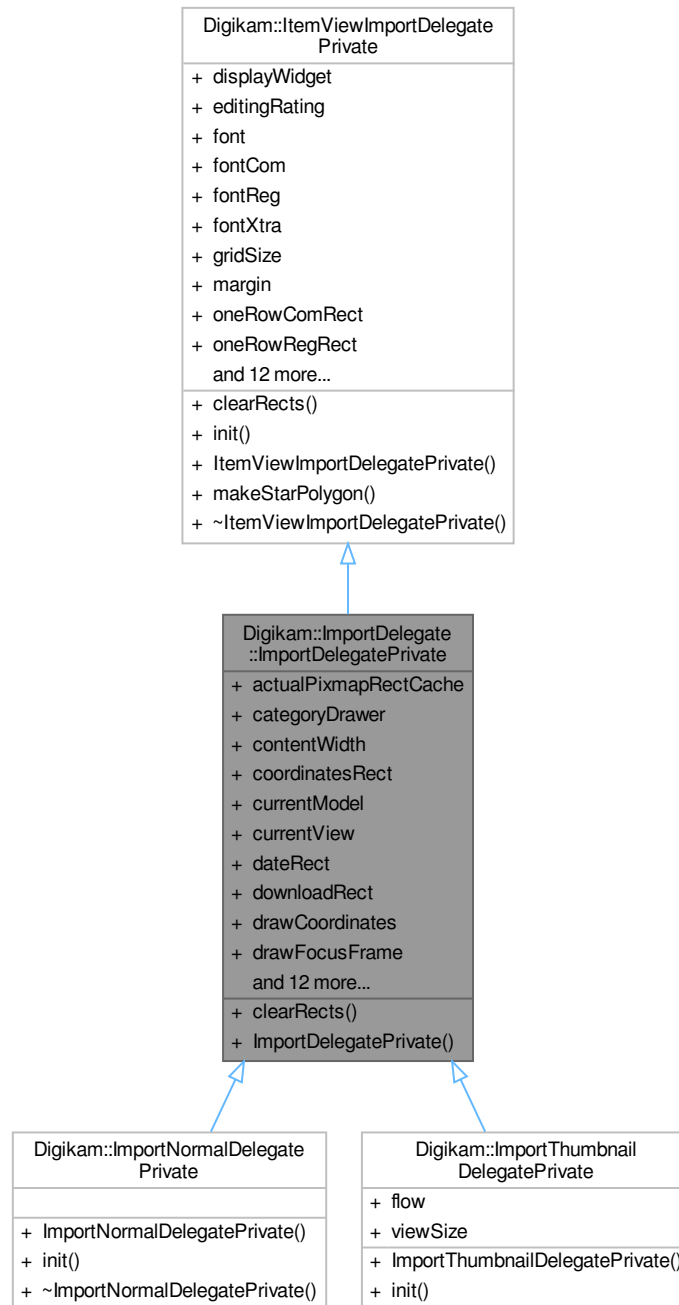
#### 6.829.1.14 `updateSizeRectsAndPixmaps()`

```
void Digikam::ImportDelegate::updateSizeRectsAndPixmaps ( ) [override], [protected], [virtual]
```

Implements [Digikam::ItemViewImportDelegate](#).

## 6.830 Digikam::ImportDelegate::ImportDelegatePrivate Class Reference

Inheritance diagram for Digikam::ImportDelegate::ImportDelegatePrivate:



### Public Member Functions

- void [clearRects](#) () override

*Resets cached rects. Remember to reimplement in subclass for added rects.*

## Public Member Functions inherited from [Digikam::ItemViewImportDelegatePrivate](#)

- void **init** ([ItemViewImportDelegate](#) \*const \_q, QWidget \*const \_widget)
- void **makeStarPolygon** ()

## Public Attributes

- QCache< int, QRect > **actualPixmapRectCache**
- [ImportCategoryDrawer](#) \* **categoryDrawer** = nullptr
- int **contentWidth** = 0
- QRect **coordinatesRect**
- QAbstractItemModel \* **currentModel** = nullptr
- [ImportCategorizedView](#) \* **currentView** = nullptr
- QRect **dateRect**
- QRect **downloadRect**
- bool **drawCoordinates** = false
- bool **drawFocusFrame** = true
- bool **drawImageFormat** = false
- bool **drawMouseOverFrame** = true
- QRect **groupRect**
- QRect **imageInformationRect**
- QRect **lockRect**
- QRect **nameRect**
- QRect **pickLabelRect**
- QRect **pixmapRect**
- bool **ratingOverThumbnail** = false
- QRect **resolutionRect**
- QRect **sizeRect**
- QRect **tagRect**

## Public Attributes inherited from [Digikam::ItemViewImportDelegatePrivate](#)

- QWidget \* **displayWidget** = nullptr
- QPersistentModelIndex **editingRating**
- QFont **font**
- QFont **fontCom**
- QFont **fontReg**
- QFont **fontXtra**
- QSize **gridSize**
- int **margin** = 5
- QRect **oneRowComRect**
- QRect **oneRowRegRect**
- QRect **oneRowXtraRect**
- [ItemViewImportDelegate](#) \* **q** = nullptr
- int **radius** = 3
- *constant values for drawing*
- QVector< QPixmap > **ratingPixmaps** = QVector< QPixmap >(10)
- QRect **ratingRect**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- QPolygon **starPolygon**
- QSize **starPolygonSize**
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)



## 6.830.1 Member Function Documentation

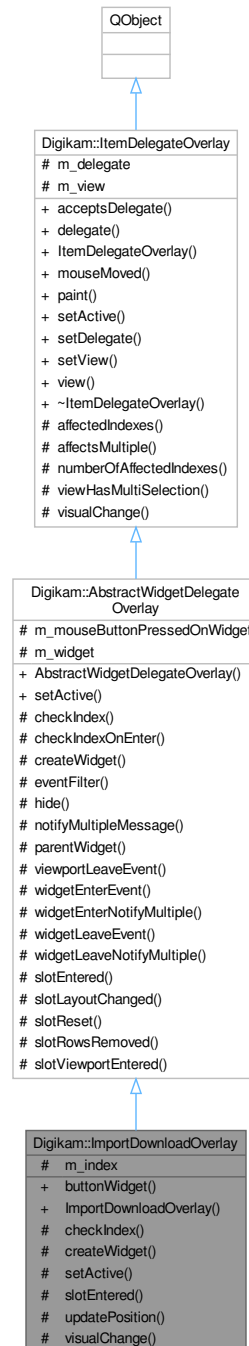
### 6.830.1.1 clearRects()

```
void Digikam::ImportDelegate::ImportDelegatePrivate::clearRects ( ) [override], [virtual]
```

Reimplemented from [Digikam::ItemViewImportDelegatePrivate](#).

## 6.831 Digikam::ImportDownloadOverlay Class Reference

Inheritance diagram for Digikam::ImportDownloadOverlay:



### Public Member Functions

- [ImportOverlayWidget](#) \* **buttonWidget** () const
- **ImportDownloadOverlay** (QObject \*const parent)

**Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

**Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)**

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

**Protected Member Functions**

- bool **checkIndex** (const QModelIndex &index) const override
- QWidget \* **createWidget** () override
- void **setActive** (bool active) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **visualChange** () override

**Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- bool **checkIndexOnEnter** (const QModelIndex &index) const
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual void **hide** ()
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* **parentWidget** () const
- virtual void **viewportLeaveEvent** (QObject \*obj, QEvent \*event)
- virtual void **widgetEnterEvent** ()
- void **widgetEnterNotifyMultiple** (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

**Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)**

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool **affectsMultiple** (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool **viewHasMultiSelection** () const

**Protected Attributes**

- QPersistentModelIndex **m\_index**

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **m\_mouseButtonPressedOnWidget** = false
- QWidget \* **m\_widget** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* **m\_delegate** = nullptr
- QAbstractItemView \* **m\_view** = nullptr

## Additional Inherited Members

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

### 6.831.1 Member Function Documentation

#### 6.831.1.1 checkIndex()

```
bool Digikam::ImportDownloadOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

#### 6.831.1.2 createWidget()

```
QWidget * Digikam::ImportDownloadOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.831.1.3 setActive()

```
void Digikam::ImportDownloadOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.831.1.4 slotEntered()

```
void Digikam::ImportDownloadOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.831.1.5 visualChange()

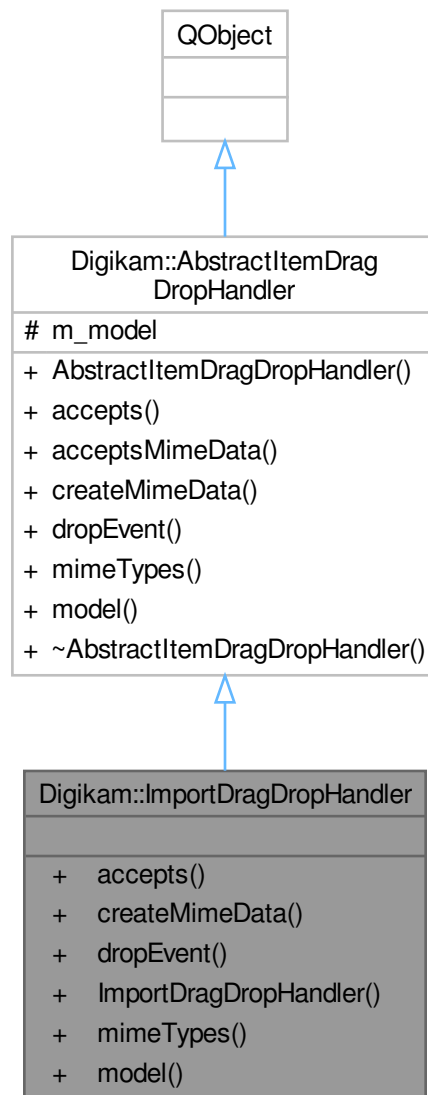
```
void Digikam::ImportDownloadOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

## 6.832 Digikam::ImportDragDropHandler Class Reference

Inheritance diagram for Digikam::ImportDragDropHandler:



### Public Member Functions

- Qt::DropAction `accepts` (const QDropEvent \*e, const QModelIndex &dropIndex) override
- QMimeData \* `createMimeData` (const QList< QModelIndex > &) override
- bool `dropEvent` (QAbstractItemView \*view, const QDropEvent \*e, const QModelIndex &droppedOn) override
- **ImportDragDropHandler** (`ImportItemModel` \*const model)
- QStringList `mimeTypes` () const override
- `ImportItemModel` \* **model** () const

## Public Member Functions inherited from [Digikam::AbstractItemDragDropHandler](#)

- **AbstractItemDragDropHandler** (QAbstractItemModel \*const model)
- virtual bool [acceptsMimeData](#) (const QMimeData \*data)
- QAbstractItemModel \* **model** () const

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::AbstractItemDragDropHandler](#)

- QAbstractItemModel \* **m\_model** = nullptr

## 6.832.1 Member Function Documentation

### 6.832.1.1 [accepts\(\)](#)

```
Qt::DropAction Digikam::ImportDragDropHandler::accepts (
    const QDropEvent * e,
    const QModelIndex & dropIndex ) [override], [virtual]
```

Returns if the given mime data is accepted for drop on dropIndex. Returns the proposed action, or Qt::IgnoreAction if not accepted.

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.832.1.2 [createMimeData\(\)](#)

```
QMimeData * Digikam::ImportDragDropHandler::createMimeData (
    const QList< QModelIndex > & ) [override], [virtual]
```

Create a mime data object for starting a drag from the given Albums

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.832.1.3 [dropEvent\(\)](#)

```
bool Digikam::ImportDragDropHandler::dropEvent (
    QAbstractItemView * view,
    const QDropEvent * e,
    const QModelIndex & droppedOn ) [override], [virtual]
```

Gives the view and the occurring drop event. The index is the index where the drop was dropped on. It may be invalid (dropped on decoration, viewport) Returns true if the event is to be accepted.

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.832.1.4 mimeTypees()

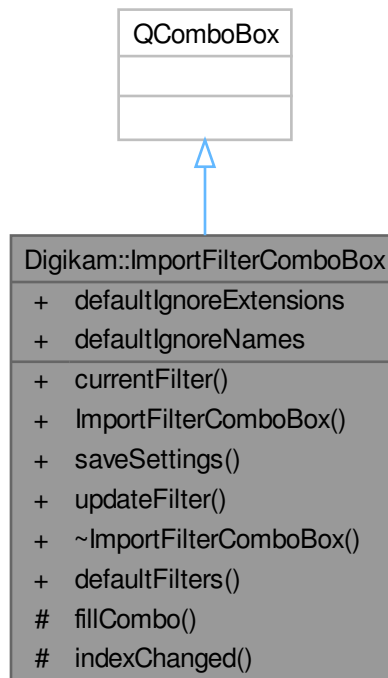
```
QStringList Digikam::ImportDragDropHandler::mimeTypees ( ) const [override], [virtual]
```

Returns the supported mime types. Called by the default implementation of model's [mimeTypees\(\)](#).

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

## 6.833 Digikam::ImportFilterComboBox Class Reference

Inheritance diagram for Digikam::ImportFilterComboBox:



### Signals

- void **signalFilterChanged** ([Filter](#) \*)

### Public Member Functions

- [Filter](#) \* **currentFilter** ( ) const
- **ImportFilterComboBox** (QWidget \*const parent)
- void **saveSettings** ( )
- void **updateFilter** ( )



### Static Public Member Functions

- static void **defaultFilters** (FilterList \*const filters)

### Static Public Attributes

- static const QString **defaultIgnoreExtensions**
- static const QString **defaultIgnoreNames**

### Protected Slots

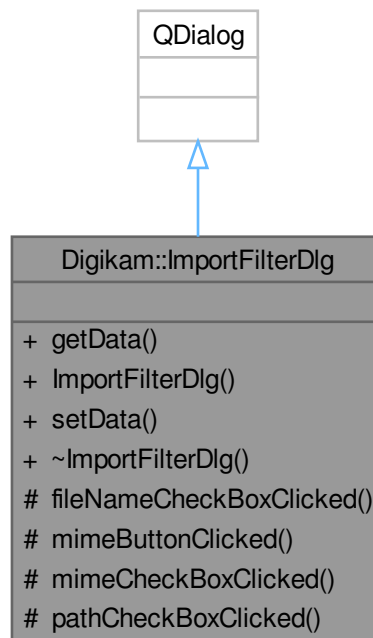
- void **indexChanged** (int index)

### Protected Member Functions

- void **fillCombo** ()

## 6.834 Digikam::ImportFilterDlg Class Reference

Inheritance diagram for Digikam::ImportFilterDlg:



**Public Member Functions**

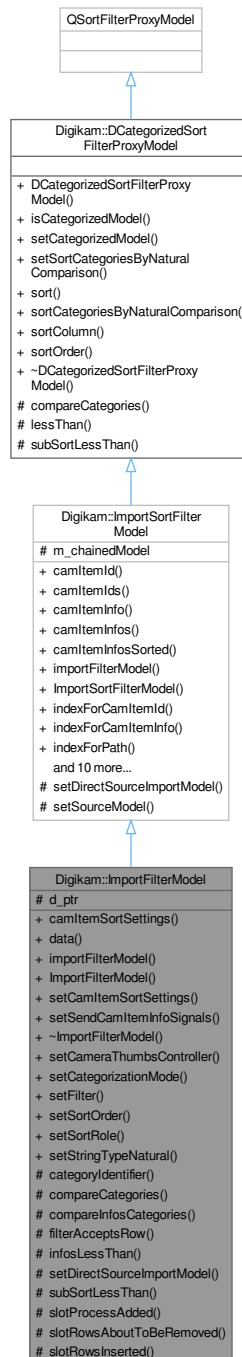
- void **getData** ([Filter](#) \*const filter)
- **ImportFilterDlg** (QWidget \*const parent=nullptr)
- void **setData** (const [Filter](#) &filter)

**Protected Slots**

- void **fileNameCheckBoxClicked** ()
- void **mimeButtonClicked** ()
- void **mimeCheckBoxClicked** ()
- void **pathCheckBoxClicked** ()

## 6.835 Digikam::ImportFilterModel Class Reference

Inheritance diagram for Digikam::ImportFilterModel:



### Public Types

- enum `ImportFilterModelRoles` {
  - `CategorizationModeRole` = `ImportItemModel::FilterModelRoles + 1` , `SortOrderRole` = `ImportItemModel::FilterModelRoles + 2` , `CategoryFormatRole` = `ImportItemModel::FilterModelRoles + 3` , `CategoryDateRole`

```
= ImportItemModel::FilterModelRoles + 4 ,
ImportFilterModelPointerRole = ImportItemModel::FilterModelRoles + 50 }
```

## Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## Public Slots

- void **setCameraThumbsController** ([CameraThumbsCtrl](#) \*const thumbsCtrl)
- void **setCategorizationMode** ([CamItemSortSettings::CategorizationMode](#) mode)
- void **setFilter** ([Filter](#) \*)
- void **setSortOrder** ([CamItemSortSettings::SortOrder](#) order)
- void **setSortRole** ([CamItemSortSettings::SortRole](#) role)
- void **setStringTypeNatural** (bool natural)

## Signals

- void **camItemInfosAboutToBeRemoved** (const QList< [CamItemInfo](#) > &infos)
- void **camItemInfosAdded** (const QList< [CamItemInfo](#) > &infos)

## Public Member Functions

- [CamItemSortSettings](#) **camItemSortSettings** () const
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- [ImportFilterModel](#) \* **importFilterModel** () const override  
*Returns this, any chained [ImportFilterModel](#), or 0.*
- [ImportFilterModel](#) (QObject \*const parent=nullptr)
- void **setCamItemSortSettings** (const [CamItemSortSettings](#) &sorter)
- void **setSendCamItemInfoSignals** (bool sendSignals)  
*Enables sending [camItemInfosAdded](#) and [camItemInfosAboutToBeRemoved](#).*

## Public Member Functions inherited from [Digikam::ImportSortFilterModel](#)

- qulonglong **camItemId** (const QModelIndex &index) const
- QList< qulonglong > **camItemIds** (const QList< QModelIndex > &indexes) const
- [CamItemInfo](#) **camItemInfo** (const QModelIndex &index) const
- QList< [CamItemInfo](#) > **camItemInfos** (const QList< QModelIndex > &indexes) const
- QList< [CamItemInfo](#) > **camItemInfosSorted** () const
- [ImportSortFilterModel](#) (QObject \*const parent=nullptr)
- QModelIndex **indexForCamItemId** (qulonglong id) const
- QModelIndex **indexForCamItemInfo** (const [CamItemInfo](#) &info) const
- QModelIndex **indexForPath** (const QString &filePath) const
- QModelIndex **mapFromDirectSourceToSourceImportModel** (const QModelIndex &sourceModelIndex) const
- QModelIndex **mapFromSourceImportModel** (const QModelIndex &importModelIndex) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceImportModel** (const QModelIndex &proxyIndex) const
- void **setSourceFilterModel** ([ImportSortFilterModel](#) \*const sourceModel)
- void **setSourceImportModel** ([ImportItemModel](#) \*const sourceModel)
- [ImportSortFilterModel](#) \* **sourceFilterModel** () const
- [ImportItemModel](#) \* **sourceImportModel** () const

**Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)**

- **DCategorizedSortFilterProxyModel** (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void **setCategorizedModel** (bool categorizedModel)
- void **setSortCategoriesByNaturalComparison** (bool [sortCategoriesByNaturalComparison](#))
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool **sortCategoriesByNaturalComparison** () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

**Protected Slots**

- void **slotProcessAdded** (const QList< [CamItemInfo](#) > &)
- void **slotRowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end)
- void **slotRowsInserted** (const QModelIndex &parent, int start, int end)

**Protected Member Functions**

- virtual QString **categoryIdentifier** (const [CamItemInfo](#) &info) const
- int **compareCategories** (const QModelIndex &left, const QModelIndex &right) const override
- virtual int **compareInfosCategories** (const [CamItemInfo](#) &left, const [CamItemInfo](#) &right) const
- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override
- virtual bool **infosLessThan** (const [CamItemInfo](#) &left, const [CamItemInfo](#) &right) const
- void **setDirectSourceImportModel** ([ImportItemModel](#) \*const sourceModel) override  
*Reimplement if needed. Called only when model shall be set as (direct) sourceModel.*
- bool **subSortLessThan** (const QModelIndex &left, const QModelIndex &right) const override

**Protected Member Functions inherited from [Digikam::ImportSortFilterModel](#)**

- void **setSourceModel** (QAbstractItemModel \*sourceModel) override

**Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)**

- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override

**Protected Attributes**

- [ImportFilterModelPrivate](#) \*const **d\_ptr**

**Protected Attributes inherited from [Digikam::ImportSortFilterModel](#)**

- [ImportSortFilterModel](#) \* **m\_chainedModel** = nullptr

**6.835.1 Member Enumeration Documentation****6.835.1.1 ImportFilterModelRoles**

```
enum Digikam::ImportFilterModel::ImportFilterModelRoles
```

## Enumerator

CategorizationModeRole	Returns the current categorization mode.
SortOrderRole	Returns the current sort order.
CategoryFormatRole	Returns the format of the index which is used for category.
CategoryDateRole	Returns the date of the index which is used for category.
ImportFilterModelPointerRole	Returns true if the given camera item is a group leader, and the group is opened.

## 6.835.2 Member Function Documentation

### 6.835.2.1 camItemInfosAdded

```
void Digikam::ImportFilterModel::camItemInfosAdded (
    const QList< CamItemInfo > & infos ) [signal]
```

Changes the current image filter settings and refilters. Changes the current image sort settings and resorts. These signals need to be explicitly enabled with `setSendItemInfoSignals()`.

### 6.835.2.2 categoryIdentifier()

```
QString Digikam::ImportFilterModel::categoryIdentifier (
    const CamItemInfo & info ) const [protected], [virtual]
```

Returns a unique identifier for the category if info. The string need not be for user display.

### 6.835.2.3 compareCategories()

```
int Digikam::ImportFilterModel::compareCategories (
    const QModelIndex & left,
    const QModelIndex & right ) const [override], [protected], [virtual]
```

This method compares the category of the `left` index with the category of the `right` index.

Internally and if not reimplemented, this method will ask for `left` and `right` models for role `Category↔SortRole`. In order to correctly sort categories, the `data()` method of the model should return a `qulonglong` (or numeric) value, or a `QString` object. `QString` objects will be sorted with `QString::localeAwareCompare` if `sortCategoriesByNaturalComparison()` is true.

#### Note

Please have present that: `QString(QChar(QChar::ObjectReplacementCharacter)) > QString(QChar(QChar↔::ReplacementCharacter)) > [ all possible strings ] > QString();`

This means that `QString()` will be sorted the first one, while `QString(QChar(QChar::ObjectReplacementCharacter))` and `QString(QChar(QChar::ReplacementCharacter))` will be sorted in last position.

### Warning

Please note that `data()` method of the model should return always information of the same type. If you return a `QString` for an index, you should return always `QStrings` for all indexes for role `CategorySortRole` in order to correctly sort categories. You can't mix by returning a `QString` for one index, and a `qlonglong` for other.

### Note

If you need a more complex layout, you will have to reimplement this method.

### Returns

A negative value if the category of `left` should be placed before the category of `right`. 0 if `left` and `right` are on the same category, and a positive value if the category of `left` should be placed after the category of `right`.

Reimplemented from [Digikam::DCategorizedSortFilterProxyModel](#).

#### 6.835.2.4 `compareInfosCategories()`

```
int Digikam::ImportFilterModel::compareInfosCategories (
    const CamItemInfo & left,
    const CamItemInfo & right ) const [protected], [virtual]
```

Reimplement to customize category sorting, Return negative if category of `left` < category `right`, Return 0 if `left` and `right` are in the same category, else return positive.

#### 6.835.2.5 `importFilterModel()`

```
ImportFilterModel * Digikam::ImportFilterModel::importFilterModel ( ) const [override], [virtual]
```

Reimplemented from [Digikam::ImportSortFilterModel](#).

#### 6.835.2.6 `infosLessThan()`

```
bool Digikam::ImportFilterModel::infosLessThan (
    const CamItemInfo & left,
    const CamItemInfo & right ) const [protected], [virtual]
```

Reimplement to customize sorting. Do not take categories into account here.

#### 6.835.2.7 `setDirectSourceImportModel()`

```
void Digikam::ImportFilterModel::setDirectSourceImportModel (
    ImportItemModel *const sourceModel ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ImportSortFilterModel](#).

### 6.835.2.8 subSortLessThan()

```
bool Digikam::ImportFilterModel::subSortLessThan (  
    const QModelIndex & left,  
    const QModelIndex & right ) const [override], [protected], [virtual]
```

This method has a similar purpose as [lessThan\(\)](#) has on `QSortFilterProxyModel`. It is used for sorting items that are in the same category.

#### Returns

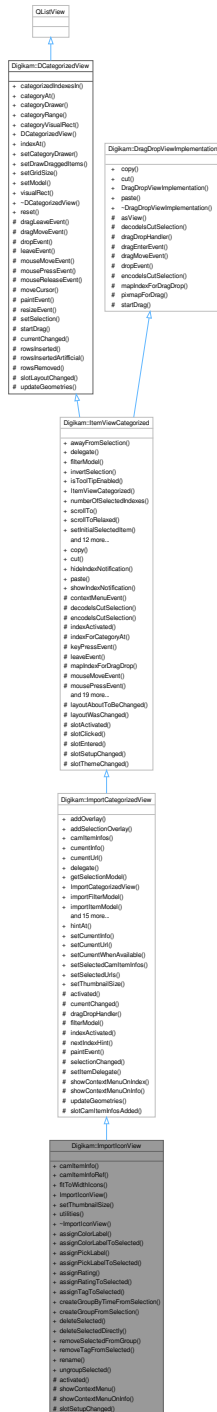
Returns true if the item `left` is less than the item `right` when sorting.

Reimplemented from [Digikam::DCategorizedSortFilterProxyModel](#).



# 6.836 Digikam::ImportIconView Class Reference

Inheritance diagram for Digikam::ImportIconView:



## Classes

- class [Private](#)

**Public Slots**

- void **assignColorLabel** (const QModelIndex &index, int colorId)
- void **assignColorLabelToSelected** (int colorId)
- void **assignPickLabel** (const QModelIndex &index, int pickId)
- void **assignPickLabelToSelected** (int pickId)
- void **assignRating** (const QList< QModelIndex > &index, int rating)
- void **assignRatingToSelected** (int rating)
- void **assignTagToSelected** (int tagID)
- void **createGroupByTimeFromSelection** ()
- void **createGroupFromSelection** ()
- void **deleteSelected** (bool permanently=false)
- void **deleteSelectedDirectly** (bool permanently=false)
- void **removeSelectedFromGroup** ()
- void **removeTagFromSelected** (int tagID)
- void **rename** ()
- void **ungroupSelected** ()

**Public Slots inherited from [Digikam::ImportCategorizedView](#)**

- void **hintAt** (const CamItemInfo &info)
- void **setCurrentInfo** (const CamItemInfo &info)
- void **setCurrentUrl** (const QUrl &url)
- void **setCurrentWhenAvailable** (qulonglong camItemId)
- void **setSelectedCamItemInfos** (const QList< CamItemInfo > &infos)
- void **setSelectedUrls** (const QList< QUrl > &urlList)
- void **setThumbnailSize** (int size)

**Public Slots inherited from [Digikam::ItemViewCategorized](#)**

- void **copy** () override
- void **cut** () override
- void **hideIndexNotification** ()
- void **paste** () override
- void **showIndexNotification** (const QModelIndex &index, const QString &message)

**Public Slots inherited from [Digikam::DCategorizedView](#)**

- void **reset** () override

**Signals**

- void **previewRequested** (const CamItemInfo &info, bool downloadPreview)

**Signals inherited from [Digikam::ImportCategorizedView](#)**

- void **camItemInfoActivated** (const CamItemInfo &info)
- void **currentChanged** (const CamItemInfo &info)
- void **deselected** (const QList< CamItemInfo > &nowDeselectedInfos)
- void **modelChanged** ()
- void **selected** (const QList< CamItemInfo > &newSelectedInfos)

## Signals inherited from [Digikam::ItemViewCategorized](#)

- void [clicked](#) (const [QMouseEvent](#) \*e, const [QModelIndex](#) &index)
- void **entered** (const [QMouseEvent](#) \*e, const [QModelIndex](#) &index)
- void [keyPressed](#) ([QKeyEvent](#) \*e)
- void [selectionChanged](#) ()
- void [selectionCleared](#) ()
- void [viewportClicked](#) (const [QMouseEvent](#) \*e)
- void **zoomInStep** ()
- void **zoomOutStep** ()

## Public Member Functions

- [CamItemInfo](#) **camItemInfo** (const [QString](#) &folder, const [QString](#) &file)
- [CamItemInfo](#) & **camItemInfoRef** (const [QString](#) &folder, const [QString](#) &file)
- int **fitToWidthIcons** ()
- [ImportIconView](#) ([QWidget](#) \*const parent=nullptr)
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &size) override
- [ItemViewUtilities](#) \* **utilities** () const

## Public Member Functions inherited from [Digikam::ImportCategorizedView](#)

- void [addOverlay](#) ([ItemDelegateOverlay](#) \*overlay, [ImportDelegate](#) \*delegate=nullptr)
- void **addSelectionOverlay** ([ImportDelegate](#) \*delegate=nullptr)
- [QList](#)< [CamItemInfo](#) > **camItemInfos** () const
- [CamItemInfo](#) **currentInfo** () const
- [QUrl](#) **currentUrl** () const
- [ImportDelegate](#) \* **delegate** () const
- [QItemSelectionModel](#) \* **getSelectionModel** () const
- [ImportCategorizedView](#) ([QWidget](#) \*const parent=nullptr)
- [ImportFilterModel](#) \* **importFilterModel** () const
- [ImportItemModel](#) \* **importItemModel** () const
- [ImportSortFilterModel](#) \* **importSortFilterModel** () const
- [ImportThumbnailModel](#) \* **importThumbnailModel** () const
- [CamItemInfo](#) **nextInfo** (const [CamItemInfo](#) &info)
- [CamItemInfo](#) **nextInOrder** (const [CamItemInfo](#) &startingPoint, int nth)
- [CamItemInfo](#) **previousInfo** (const [CamItemInfo](#) &info)
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [QList](#)< [CamItemInfo](#) > **selectedCamItemInfos** () const
- [QList](#)< [CamItemInfo](#) > **selectedCamItemInfosCurrentFirst** () const
- [QList](#)< [QUrl](#) > **selectedUrls** () const
- void **setModels** ([ImportItemModel](#) \*model, [ImportSortFilterModel](#) \*filterModel)
- [ThumbnailSize](#) **thumbnailSize** () const
- void **toIndex** (const [QUrl](#) &url)
- [QList](#)< [QUrl](#) > **urls** () const

## Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **awayFromSelection** ()
- [DItemDelegate](#) \* **delegate** () const
- void **invertSelection** ()
- bool **isToolTipEnabled** () const
- [ItemViewCategorized](#) (QWidget \*const parent=nullptr)
- int **numberOfSelectedIndexes** () const
- void **scrollTo** (const QModelIndex &index, ScrollHint hint=EnsureVisible) override
- void **scrollToRelaxed** (const QModelIndex &index, ScrollHint hint=EnsureVisible)
- void **setInitialSelectedItem** (bool enabled)
- void **setScrollCurrentToCenter** (bool enabled)
- void **setScrollStepGranularity** (int factor)
- void **setSelectedIndexes** (const QList< QModelIndex > &indexes)
- void **setSpacing** (int spacing)
- void **setToolTipEnabled** (bool enabled)
- void **setUsePointingHandCursor** (bool useCursor)
- void **toFirstIndex** ()
- void **toIndex** (const QModelIndex &index)
- void **toLastIndex** ()
- void **toNextIndex** ()
- void **toPreviousIndex** ()

## Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual QModelIndexList **categorizedIndexesIn** (const QRect &rect) const
- virtual QModelIndex **categoryAt** (const QPoint &point) const
- [DCategoryDrawer](#) \* **categoryDrawer** () const
- virtual QItemSelectionRange **categoryRange** (const QModelIndex &index) const
- virtual QRect **categoryVisualRect** (const QModelIndex &index) const
- [DCategorizedView](#) (QWidget \*const parent=nullptr)
- QModelIndex **indexAt** (const QPoint &point) const override
- void **setCategoryDrawer** ([DCategoryDrawer](#) \*categoryDrawer)
- void **setDrawDraggedItems** (bool drawDraggedItems)
- void **setGridSize** (const QSize &size)
- void **setModel** (QAbstractItemModel \*model) override
- QRect **visualRect** (const QModelIndex &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

## Protected Member Functions

- void **activated** (const [CamItemInfo](#) &info, Qt::KeyboardModifiers modifiers) override  
*Reimplement these in a subclass.*
- void **showContextMenu** (QContextMenuEvent \*event) override
- void **showContextMenuOnInfo** (QContextMenuEvent \*event, const [CamItemInfo](#) &info) override
- void **slotSetupChanged** () override

## Protected Member Functions inherited from [Digikam::ImportCategorizedView](#)

- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override
  - reimplemented from parent class*
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **paintEvent** (QPaintEvent \*e) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([ImportDelegate](#) \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override
  - Reimplement these in a subclass.*
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override
  - reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([DItemDelegate](#) \*delegate)
- void **setToolTip** ([ItemViewToolTip](#) \*tip)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

### Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

### Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

### Additional Inherited Members

### Protected Slots inherited from [Digikam::ImportCategorizedView](#)

- void **slotCamItemInfosAdded** ()

### Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void **layoutAboutToBeChanged** ()
- void **layoutWasChanged** ()
- void **slotActivated** (const QModelIndex &index)
- void **slotClicked** (const QModelIndex &index)
- void **slotEntered** (const QModelIndex &index)
- virtual void **slotThemeChanged** ()

### Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## 6.836.1 Member Function Documentation

### 6.836.1.1 activated()

```
void Digikam::ImportIconView::activated (
    const CamItemInfo & info,
    Qt::KeyboardModifiers modifiers ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ImportCategorizedView](#).

### 6.836.1.2 setThumbnailSize()

```
void Digikam::ImportIconView::setThumbnailSize (
    const ThumbnailSize & size ) [override], [virtual]
```

Reimplemented from [Digikam::ImportCategorizedView](#).

### 6.836.1.3 showContextMenu()

```
void Digikam::ImportIconView::showContextMenu (
    QContextMenuEvent * event ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

### 6.836.1.4 showContextMenuOnInfo()

```
void Digikam::ImportIconView::showContextMenuOnInfo (
    QContextMenuEvent * event,
    const CamItemInfo & info ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ImportCategorizedView](#).

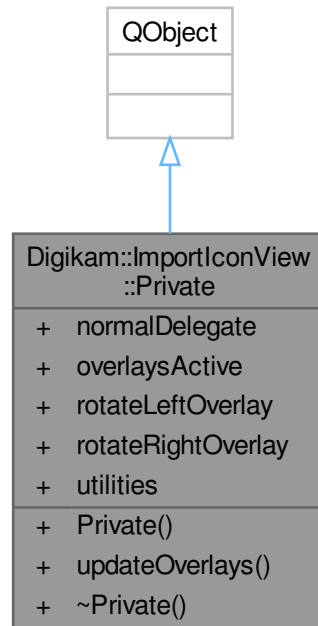
### 6.836.1.5 slotSetupChanged()

```
void Digikam::ImportIconView::slotSetupChanged ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

## 6.837 Digikam::ImportIconView::Private Class Reference

Inheritance diagram for Digikam::ImportIconView::Private:



### Public Member Functions

- `Private (ImportIconView *const qq)`
- `void updateOverlays ()`

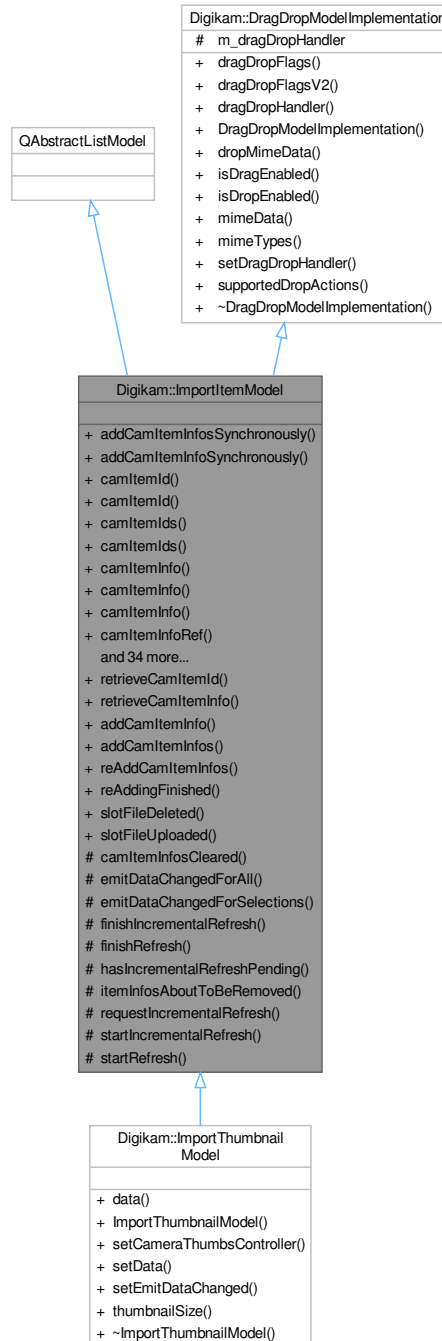
### Public Attributes

- `ImportNormalDelegate * normalDelegate = nullptr`
- `bool overlaysActive = false`
- `ImportRotateOverlay * rotateLeftOverlay = nullptr`
- `ImportRotateOverlay * rotateRightOverlay = nullptr`
- `ItemViewUtilities * utilities = nullptr`



## 6.838 Digikam::ImportItemModel Class Reference

Inheritance diagram for Digikam::ImportItemModel:



### Public Types

- enum [ImportItemModelRoles](#) {
  - [ImportItemModelPointerRole](#) = Qt::UserRole , [ImportItemModelInternalId](#) = Qt::UserRole + 1 ,
  - [ThumbnailRole](#) = Qt::UserRole + 2 , [ExtraDataRole](#) = Qt::UserRole + 3 ,
  - [ExtraDataDuplicateCount](#) = Qt::UserRole + 6 , [FilterModelRoles](#) = Qt::UserRole + 100 }

## Public Slots

- void **addCamItemInfo** (const [CamItemInfo](#) &info)
- void **addCamItemInfos** (const [CamItemInfoList](#) &infos)
- void **reAddCamItemInfos** (const [CamItemInfoList](#) &infos)
- void **reAddingFinished** ()
- void **slotFileDeleted** (const [QString](#) &folder, const [QString](#) &file, bool status)
- void **slotFileUploaded** (const [CamItemInfo](#) &info)

## Signals

- void [allRefreshingFinished](#) ()
- void [itemInfosAboutToBeAdded](#) (const [QList](#)< [CamItemInfo](#) > &infos)
- void [itemInfosAboutToBeRemoved](#) (const [QList](#)< [CamItemInfo](#) > &infos)
- void [itemInfosAdded](#) (const [QList](#)< [CamItemInfo](#) > &infos)
- void [itemInfosRemoved](#) (const [QList](#)< [CamItemInfo](#) > &infos)
- void [preprocess](#) (const [QList](#)< [CamItemInfo](#) > &infos)
- void **processAdded** (const [QList](#)< [CamItemInfo](#) > &infos)
- void [readyForIncrementalRefresh](#) ()

## Public Member Functions

- void **addCamItemInfosSynchronously** (const [Digikam::CamItemInfoList](#) &infos)
- void [addCamItemInfoSynchronously](#) (const [CamItemInfo](#) &info)
- qlonglong **camItemId** (const [QModelIndex](#) &index) const
- qlonglong **camItemId** (int row) const
- [QList](#)< qlonglong > **camItemIds** () const
- [QList](#)< qlonglong > **camItemIds** (const [QList](#)< [QModelIndex](#) > &indexes) const
- [CamItemInfo](#) **camItemInfo** (const [QModelIndex](#) &index) const
- [CamItemInfo](#) **camItemInfo** (const [QUrl](#) &fileUrl) const
- [CamItemInfo](#) **camItemInfo** (int row) const
- [CamItemInfo](#) & **camItemInfoRef** (const [QModelIndex](#) &index) const
- [CamItemInfo](#) & **camItemInfoRef** (int row) const
- [QList](#)< [CamItemInfo](#) > **camItemInfos** () const
- [CamItemInfoList](#) **camItemInfos** (const [QList](#)< [QModelIndex](#) > &indexes) const
- [QList](#)< [CamItemInfo](#) > **camItemInfos** (const [QUrl](#) &fileUrl) const
- void [clearCamItemInfos](#) ()
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role) const override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasImage** (const [CamItemInfo](#) &info) const
- bool **hasImage** (qlonglong id) const
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role) const override
- **ImportItemModel** ([QObject](#) \*const parent=nullptr)
- [QModelIndex](#) **index** (int row, int column, const [QModelIndex](#) &parent) const override
- [QList](#)< [QModelIndex](#) > **indexesForCamItemId** (qlonglong id) const
- [QList](#)< [QModelIndex](#) > **indexesForCamItemInfo** (const [CamItemInfo](#) &info) const
- [QList](#)< [QModelIndex](#) > **indexesForUrl** (const [QUrl](#) &fileUrl) const
- [QModelIndex](#) **indexForCamItemId** (qlonglong id) const
- [QModelIndex](#) **indexForCamItemInfo** (const [CamItemInfo](#) &info) const
- [QModelIndex](#) **indexForUrl** (const [QUrl](#) &fileUrl) const
- bool **isEmpty** () const
- bool [isRefreshing](#) () const
- bool **keepsFileUrlCache** () const

- int **numberOfIndexesForCamItemid** (qulonglong id) const
  - int **numberOfIndexesForCamItemInfo** (const [CamItemInfo](#) &info) const
  - void **removeCamItemInfo** (const [CamItemInfo](#) &info)
  - void **removeCamItemInfos** (const QList< [CamItemInfo](#) > &infos)
  - void **removeIndex** (const QModelIndex &index)
  - void **removeIndexes** (const QList< QModelIndex > &indexes)
  - int **rowCount** (const QModelIndex &parent) const override
- QAbstractListModel implementation.*
- virtual void **setCameraThumbsController** ([CameraThumbsCtrl](#) \*const controller)
  - void **setCamItemInfos** (const CamItemInfoList &infos)
  - void **setKeepsFileUrlCache** (bool keepCache)
  - DECLARE\_MODEL\_DRAG\_DROP\_METHODS void **setSendRemovalSignals** (bool send)
- DragDrop methods.*
- QList< [CamItemInfo](#) > **uniqueCamItemInfos** () const

## Public Member Functions inherited from [Digikam::DragDropModelImplementation](#)

- virtual Qt::ItemFlags **dragDropFlags** (const QModelIndex &index) const
- Qt::ItemFlags **dragDropFlagsV2** (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const
- [DragDropModelImplementation](#) ()=default
- bool **dropMimeData** (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool **isDragEnabled** (const QModelIndex &index) const
- virtual bool **isDropEnabled** (const QModelIndex &index) const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const
- QStringList **mimeTypes** () const
- void **setDragDropHandler** ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions **supportedDropActions** () const

## Static Public Member Functions

- static qulonglong **retrieveCamItemid** (const QModelIndex &index)
- static [CamItemInfo](#) **retrieveCamItemInfo** (const QModelIndex &index)

## Protected Member Functions

- virtual void **camItemInfosCleared** ()
- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelections** (const QItemSelection &selection)
- void **finishIncrementalRefresh** ()
- void **finishRefresh** ()
- bool **hasIncrementalRefreshPending** () const
- virtual void **itemInfosAboutToBeRemoved** (int, int)
- void **requestIncrementalRefresh** ()
- void **startIncrementalRefresh** ()
- void **startRefresh** ()

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr

## 6.838.1 Member Enumeration Documentation

### 6.838.1.1 ImportItemModelRoles

```
enum Digikam::ImportItemModel::ImportItemModelRoles
```

Enumerator

ImportItemModelPointerRole	An ImportItemModel* pointer to this model
ThumbnailRole	Returns a thumbnail pixmap. May be implemented by subclasses. Returns either a valid pixmap or a null QVariant.
ExtraDataRole	Return (optional) extraData field
ExtraDataDuplicateCount	Returns the number of duplicate indexes for the same image id

## 6.838.2 Member Function Documentation

### 6.838.2.1 addCamItemInfoSynchronously()

```
void Digikam::ImportItemModel::addCamItemInfoSynchronously (
    const CamItemInfo & info )
```

addCamItemInfo() is asynchronous if a preprocessor is set. This method first adds the info, synchronously. Only afterwards, the preprocessor will have the opportunity to process it. This method also bypasses any incremental updates.

### 6.838.2.2 allRefreshingFinished

```
void Digikam::ImportItemModel::allRefreshingFinished ( ) [signal]
```

Signals that the model has finished currently with all scheduled refreshing, full or incremental, and all preprocessing. The model is in polished, clean situation right now.

### 6.838.2.3 camItemInfo() [1/2]

```
CamItemInfo Digikam::ImportItemModel::camItemInfo (
    const QModelIndex & index ) const
```

Returns the [CamItemInfo](#) object, reference from the underlying data pointed to by the index. For camItemInfo and camItemInfoRef if the index is not valid they will return a null [CamItemInfo](#), and 0 respectively, camItemInfoRef must not be called with an invalid index as it will crash.

### 6.838.2.4 camItemInfo() [2/2]

```
CamItemInfo Digikam::ImportItemModel::camItemInfo (
    int row ) const
```

Returns the [CamItemInfo](#) object, reference from the underlying data of the given row (parent is the invalid QModelIndex, column is 0). Note that camItemInfoRef must not be called with an invalid index as it will crash.

### 6.838.2.5 camItemInfosCleared()

```
virtual void Digikam::ImportItemModel::camItemInfosCleared ( ) [inline], [protected], [virtual]
```

Called when the internal storage is cleared.

### 6.838.2.6 clearCamItemInfos()

```
void Digikam::ImportItemModel::clearCamItemInfos ( )
```

Clears the CamItemInfos and resets the model.

### 6.838.2.7 indexForCamItemInfo()

```
QModelIndex Digikam::ImportItemModel::indexForCamItemInfo (
    const CamItemInfo & info ) const
```

Return the index of a given [CamItemInfo](#), if it exists in the model.

### 6.838.2.8 indexForUrl()

```
QModelIndex Digikam::ImportItemModel::indexForUrl (
    const QUrl & fileUrl ) const
```

Returns the index or [CamItemInfo](#) object from the underlying data for the given file url. In case of multiple occurrences of the same file, the simpler overrides returns any one found first, use the QList methods to retrieve all occurrences.

### 6.838.2.9 isRefreshing()

```
bool Digikam::ImportItemModel::isRefreshing ( ) const
```

Returns true if this model is currently refreshing. For a preprocessor this means that, although the preprocessor may currently have processed all it got, more batches are to be expected.

### 6.838.2.10 itemInfosAboutToBeAdded

```
void Digikam::ImportItemModel::itemInfosAboutToBeAdded (
    const QList< CamItemInfo > & infos ) [signal]
```

Notifies that ItemInfos will be added to the model. This signal is sent before the model data is changed and views are informed.

### 6.838.2.11 itemInfosAboutToBeRemoved [1/2]

```
void Digikam::ImportItemModel::itemInfosAboutToBeRemoved (
    const QList< CamItemInfo > & infos ) [signal]
```

Notifies that CamItemInfos will be removed from the model. This signal is sent before the model data is changed and views are informed. Note: You need to explicitly enable sending of this signal. It is not sent in [clearCamItemInfos\(\)](#).

#### 6.838.2.12 itemInfosAboutToBeRemoved() [2/2]

```
virtual void Digikam::ImportItemModel::itemInfosAboutToBeRemoved (
    int ,
    int ) [inline], [protected], [virtual]
```

Called before rowsAboutToBeRemoved

#### 6.838.2.13 itemInfosAdded

```
void Digikam::ImportItemModel::itemInfosAdded (
    const QList< CamItemInfo > & infos ) [signal]
```

Informs that ItemInfos have been added to the model. This signal is sent after the model data is changed and views are informed.

#### 6.838.2.14 itemInfosRemoved

```
void Digikam::ImportItemModel::itemInfosRemoved (
    const QList< CamItemInfo > & infos ) [signal]
```

Informs that CamItemInfos have been removed from the model. This signal is sent after the model data is changed and views are informed. Note: You need to explicitly enable sending of this signal. It is not sent in [clearCamItemInfos\(\)](#).

#### 6.838.2.15 preprocess

```
void Digikam::ImportItemModel::preprocess (
    const QList< CamItemInfo > & infos ) [signal]
```

Connect to this signal only if you are the current preprocessor.

#### 6.838.2.16 readyForIncrementalRefresh

```
void Digikam::ImportItemModel::readyForIncrementalRefresh ( ) [signal]
```

Signals that the model is right now ready to start an incremental refresh. This is guaranteed only for the scope of emitting this signal.

#### 6.838.2.17 removeIndex()

```
void Digikam::ImportItemModel::removeIndex (
    const QModelIndex & index )
```

Remove the given infos or indexes directly from the model.

### 6.838.2.18 requestIncrementalRefresh()

```
void Digikam::ImportItemModel::requestIncrementalRefresh ( ) [protected]
```

As soon as the model is ready to start an incremental refresh, the signal [readyForIncrementalRefresh\(\)](#) will be emitted. The signal will be emitted inline if the model is ready right now.

### 6.838.2.19 retrieveCamItemInfo()

```
CamItemInfo Digikam::ImportItemModel::retrieveCamItemInfo (
    const QModelIndex & index ) [static]
```

Retrieve the [CamItemInfo](#) object from the `data()` function of the given index. The index may be from a [QSortFilterProxyModel](#) as long as an [ImportItemModel](#) is at the end.

### 6.838.2.20 setCameraThumbsController()

```
void Digikam::ImportItemModel::setCameraThumbsController (
    CameraThumbsCtrl *const controller ) [virtual]
```

Used to set the camera controller, and connect with it.

Reimplemented in [Digikam::ImportThumbnailModel](#).

### 6.838.2.21 setCamItemInfos()

```
void Digikam::ImportItemModel::setCamItemInfos (
    const CamItemInfoList & infos )
```

Clears and adds infos.

### 6.838.2.22 setKeepsFileUrlCache()

```
void Digikam::ImportItemModel::setKeepsFileUrlCache (
    bool keepCache )
```

If a cache is kept, lookup by file path is fast, without a cache it is  $O(n)$ . Default is false.

### 6.838.2.23 setSendRemovalSignals()

```
void Digikam::ImportItemModel::setSendRemovalSignals (
    bool send )
```

Enable sending of `itemInfosAboutToBeRemoved` and `itemInfosRemoved` signals. Default: false

**6.838.2.24 startIncrementalRefresh()**

```
void Digikam::ImportItemModel::startIncrementalRefresh ( ) [protected]
```

Starts an incremental refresh operation. You shall only call this method from a slot connected to [readyForIncrementalRefresh\(\)](#). To initiate an incremental refresh, call [requestIncrementalRefresh\(\)](#).

**6.838.2.25 startRefresh()**

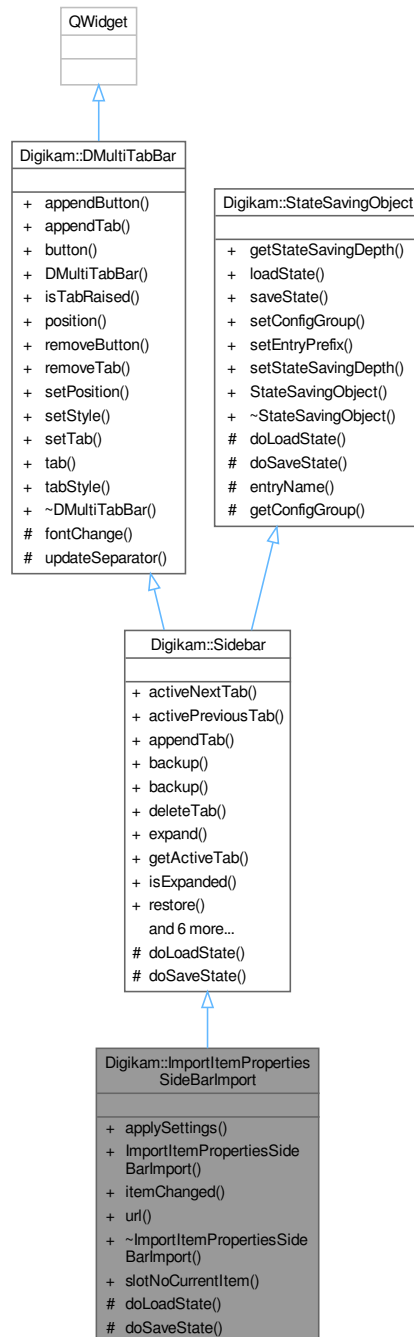
```
void Digikam::ImportItemModel::startRefresh ( ) [protected]
```

Subclasses that add CamItemInfos in batches shall call [startRefresh\(\)](#) when they start sending batches and [finishRefresh\(\)](#) when they have finished. No incremental refreshes will be started while listing. A [clearCamItemInfos\(\)](#) always stops listing, calling [finishRefresh\(\)](#) is then not necessary.



## 6.839 Digikam::ImportItemPropertiesSideBarImport Class Reference

Inheritance diagram for Digikam::ImportItemPropertiesSideBarImport:



### Public Slots

- virtual void `slotNoCurrentItem ()`

## Signals

- void **signalFirstItem** ()
- void **signalLastItem** ()
- void **signalNextItem** ()
- void **signalPrevItem** ()

## Signals inherited from [Digikam::Sidebar](#)

- void [signalChangedTab](#) (QWidget \*w)
- void [signalViewChanged](#) ()

## Public Member Functions

- void [applySettings](#) ()
- **ImportItemPropertiesSideBarImport** (QWidget \*const parent, [SidebarSplitter](#) \*const splitter, Qt::Edge side=Qt::LeftEdge, bool mimimizedDefault=false)
- void **itemChanged** (const [CamItemInfo](#) &itemInfo, const [DMetadata](#) &meta)
- [QUrl url](#) () const

## Public Member Functions inherited from [Digikam::Sidebar](#)

- void [activeNextTab](#) ()
- void [activePreviousTab](#) ()
- void [appendTab](#) (QWidget \*const w, const [QIcon](#) &pic, const [QString](#) &title)
- void [backup](#) ()
- void [backup](#) (const [QList](#)< QWidget \* > &thirdWidgetsToBackup, [QList](#)< int > \*const sizes)
- void [deleteTab](#) (QWidget \*const w)
- void [expand](#) ()
- QWidget \* [getActiveTab](#) () const
- bool [isExpanded](#) () const
- void [restore](#) ()
- void [restore](#) (const [QList](#)< QWidget \* > &thirdWidgetsToRestore, const [QList](#)< int > &sizes)
- void [setActiveTab](#) (QWidget \*const w)
- void [shrink](#) ()
- [Sidebar](#) (QWidget \*const parent, [SidebarSplitter](#) \*const sp, Qt::Edge side=Qt::LeftEdge, bool minimized↔ Default=false)
- [SidebarSplitter](#) \* [splitter](#) () const

## Public Member Functions inherited from [Digikam::DMultiTabBar](#)

- void [appendButton](#) (const [QIcon](#) &pic, int id=-1, [QMenu](#) \*const popup=nullptr, const [QString](#) &not\_used\_↔ yet=QString())
- void [appendTab](#) (const [QIcon](#) &pic, int id=-1, const [QString](#) &text=QString())
- [DMultiTabBarButton](#) \* [button](#) (int id) const
- **DMultiTabBar** (Qt::Edge pos, QWidget \*const parent=nullptr)
- bool [isTabRaised](#) (int id) const
- Qt::Edge [position](#) () const
- void [removeButton](#) (int id)
- void [removeTab](#) (int id)
- void [setPosition](#) (Qt::Edge pos)
- void [setStyle](#) ([TextStyle](#) style)
- void [setTab](#) (int id, bool state)
- [DMultiTabBarTab](#) \* [tab](#) (int id) const
- [TextStyle tabStyle](#) () const

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Protected Member Functions

- void [doLoadState](#) () override
- void [doSaveState](#) () override

## Protected Member Functions inherited from [Digikam::Sidebar](#)

- void [doLoadState](#) () override
- void [doSaveState](#) () override

## Protected Member Functions inherited from [Digikam::DMultiTabBar](#)

- virtual void [fontChange](#) (const QFont &)
- void [updateSeparator](#) ()

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## Additional Inherited Members

## Public Types inherited from [Digikam::DMultiTabBar](#)

- enum [TextStyle](#) { [ActiveIconText](#) = 0 , [AllIconsText](#) = 2 }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## 6.839.1 Member Function Documentation

### 6.839.1.1 [applySettings\(\)](#)

```
void Digikam::ImportItemPropertiesSideBarImport::applySettings ( )
```

### 6.839.1.2 doLoadState()

```
void Digikam::ImportItemPropertiesSideBarImport::doLoadState ( ) [override], [protected],  
[virtual]
```

load the last view state from disk - called by [StateSavingObject::loadState\(\)](#)

Implements [Digikam::StateSavingObject](#).

### 6.839.1.3 doSaveState()

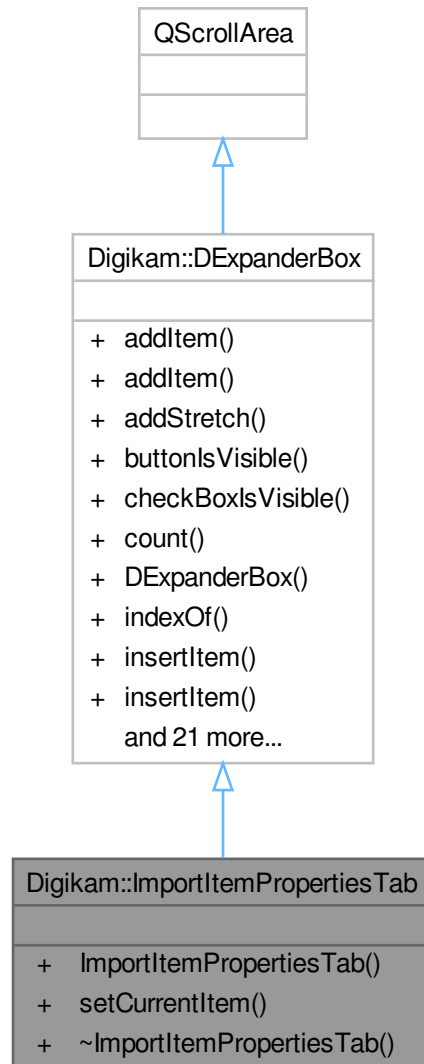
```
void Digikam::ImportItemPropertiesSideBarImport::doSaveState ( ) [override], [protected],  
[virtual]
```

save the view state to disk - called by [StateSavingObject::saveState\(\)](#)

Implements [Digikam::StateSavingObject](#).

## 6.840 Digikam::ImportItemPropertiesTab Class Reference

Inheritance diagram for Digikam::ImportItemPropertiesTab:



### Public Member Functions

- **ImportItemPropertiesTab** (QWidget \*const parent)
- void **setCurrentItem** (const [CamItemInfo](#) &itemInfo=[CamItemInfo](#)(), [DMetadata](#) \*const meta=nullptr)

### Public Member Functions inherited from [Digikam::DExpanderBox](#)

- void [addItem](#) (QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)

- void **addItem** (QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **addStretch** ()
- bool **buttonIsVisible** (int index) const
- bool **checkboxIsVisible** (int index) const
- int **count** () const
- **DExpanderBox** (QWidget \*const parent=nullptr)
- int **indexOf** (DLabelExpander \*const widget) const
- void **insertItem** (int index, QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertItem** (int index, QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertStretch** (int index)
- bool **isChecked** (int index) const
- bool **isItemEnabled** (int index) const
- bool **isItemExpanded** (int index) const
- QIcon **itemIcon** (int index) const
- QString **itemText** (int index) const
- QString **itemToolTip** (int index) const
- virtual void **readSettings** (KConfigGroup &group)
- void **removeItem** (int index)
- void **setButtonIcon** (int index, const QIcon &icon)
- void **setButtonVisible** (int index, bool b)
- void **setCheckBoxVisible** (int index, bool b)
- void **setChecked** (int index, bool b)
- void **setItemEnabled** (int index, bool enabled)
- void **setItemExpanded** (int index, bool b)
- void **setItemIcon** (int index, const QIcon &icon)
- void **setItemText** (int index, const QString &txt)
- void **setItemToolTip** (int index, const QString &tip)
- DLabelExpander \* **widget** (int index) const
- virtual void **writeSettings** (KConfigGroup &group)

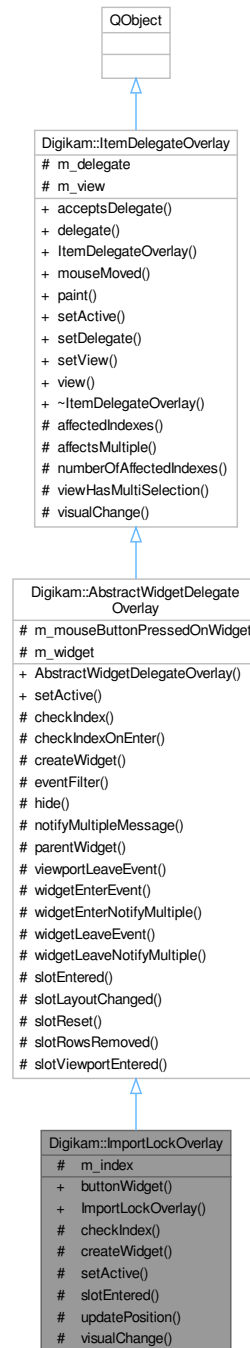
### Additional Inherited Members

### Signals inherited from [Digikam::DExpanderBox](#)

- void **signalItemButtonPressed** (int index)
- void **signalItemExpanded** (int index, bool b)
- void **signalItemToggled** (int index, bool b)

## 6.841 Digikam::ImportLockOverlay Class Reference

Inheritance diagram for Digikam::ImportLockOverlay:



### Public Member Functions

- [ImportOverlayWidget](#) \* **buttonWidget** () const
- **ImportLockOverlay** (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- QWidget \* **createWidget** () override
- void **setActive** (bool active) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **visualChange** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **checkIndexOnEnter** (const QModelIndex &index) const
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual void **hide** ()
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* **parentWidget** () const
- virtual void **viewportLeaveEvent** (QObject \*obj, QEvent \*event)
- virtual void **widgetEnterEvent** ()
- void **widgetEnterNotifyMultiple** (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool **affectsMultiple** (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool **viewHasMultiSelection** () const

## Protected Attributes

- QPersistentModelIndex **m\_index**



## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **m\_mouseButtonPressedOnWidget** = false
- QWidget \* **m\_widget** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* **m\_delegate** = nullptr
- QAbstractItemView \* **m\_view** = nullptr

## Additional Inherited Members

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

### 6.841.1 Member Function Documentation

#### 6.841.1.1 `checkIndex()`

```
bool Digikam::ImportLockOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

#### 6.841.1.2 `createWidget()`

```
QWidget * Digikam::ImportLockOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass `parentWidget()` as parent widget. Ownership of the object is passed. It will be deleted in `setActive(false)`.

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.841.1.3 setActive()

```
void Digikam::ImportLockOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.841.1.4 slotEntered()

```
void Digikam::ImportLockOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.841.1.5 visualChange()

```
void Digikam::ImportLockOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

## 6.842 Digikam::ImportNormalDelegate Class Reference

Inheritance diagram for Digikam::ImportNormalDelegate:



### Public Member Functions

- **ImportNormalDelegate** ([ImportCategorizedView](#) \*const parent)

## Public Member Functions inherited from [Digikam::ImportDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect [actualPixmapRect](#) (const QModelIndex &index) const
- int [calculatethumbSizeToFit](#) (int ws)
- [ImportCategoryDrawer](#) \* [categoryDrawer](#) () const
- QRect [coordinatesIndicatorRect](#) () const
- QRect [downloadIndicatorRect](#) () const
- QRect [groupIndicatorRect](#) () const
- QRect [imageInformationRect](#) () const override
- [ImportDelegate](#) (QWidget \*const parent)
- QRect [lockIndicatorRect](#) () const
- void [paint](#) (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setSpacing](#) (int spacing) override
- void [setView](#) ([ImportCategorizedView](#) \*view)
- QRect [tagsRect](#) () const

## Public Member Functions inherited from [Digikam::ItemViewImportDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double [displayRatio](#) () const
- QSize [gridSize](#) () const override
- [ItemViewImportDelegate](#) (QWidget \*const parent)
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect [ratingRect](#) () const
- QRect [rect](#) () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setRatingEdited](#) (const QModelIndex &index)
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from [DItemDelegate](#)*
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int [spacing](#) () const
- [ThumbnailSize](#) [thumbnailSize](#) () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- [DItemDelegate](#) (QObject \*const parent=nullptr)

## Public Member Functions inherited from Digikam::ItemDelegateOverlayContainer

- void **installOverlay** (ItemDelegateOverlay \*overlay)
- ItemDelegateOverlayContainer ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< ItemDelegateOverlay \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** (ItemDelegateOverlay \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Protected Member Functions

- **ImportNormalDelegate** (ImportNormalDelegatePrivate &dd, ImportCategorizedView \*const parent)
- void **updateRects** () override

## Protected Member Functions inherited from Digikam::ImportDelegate

- void **clearCaches** () override
  - virtual void **clearModelDataCaches** ()
  - **ImportDelegate** (ImportDelegate::ImportDelegatePrivate &dd, QWidget \*const parent)
  - void **invalidatePaintingCache** () override
- reimplement these in subclasses*
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
  - void **setModel** (QAbstractItemModel \*model)
  - virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
  - void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
  - virtual void **updateContentWidth** ()
  - void **updateSizeRectsAndPxmmaps** () override

## Protected Member Functions inherited from Digikam::ItemViewImportDelegate

- QAbstractItemDelegate \* **asDelegate** () override
- Returns the delegate, typically, the derived class.*
- void **drawColorLabelLine** (QPainter \*p, const QRect &pixRect, int colorId) const
  - void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
  - void **drawDownloadIndicator** (QPainter \*p, const QRect &r, int itemType) const
  - void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
  - void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
  - void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
  - void **drawGroupIndicator** (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
  - void **drawImageFormat** (QPainter \*p, const QRect &dimsRect, const QString &mime) const
  - void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
  - void **drawLockIndicator** (QPainter \*p, const QRect &r, int lockStatus) const
  - void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
  - void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
  - void **drawPickLabelIcon** (QPainter \*p, const QRect &r, int pickLabel) const
  - void **drawRating** (QPainter \*p, const QModelIndex &index, const QRect &ratingRect, int rating, bool isSelected) const
  - void **drawTags** (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const

- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail) const

*Use the tool methods for painting in subclasses.*

- **ItemViewImportDelegate** ([ItemViewImportDelegatePrivate](#) &dd, QWidget \*const parent)
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- void **prepareRatingPixmaps** (bool composeOverBackground=true)
- QPixmap **ratingPixmap** (int rating, bool selected) const

*Returns the relevant pixmap from the cached rating pixmaps.*

### Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)

*Declare as slot in the derived class calling this method.*

### Additional Inherited Members

### Signals inherited from [Digikam::ItemViewImportDelegate](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

### Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

### Static Public Member Functions inherited from [Digikam::ImportDelegate](#)

- static QPixmap **retrieveThumbnailPixmap** (const QModelIndex &index, int thumbnailSize)

### Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

### Protected Slots inherited from [Digikam::ImportDelegate](#)

- void `modelChanged` ()
- void `modelContentsChanged` ()

### Protected Slots inherited from [Digikam::ItemViewImportDelegate](#)

- void `overlayDestroyed` (QObject \*o) override
- void `slotSetupChanged` ()
- void `slotThemeChanged` ()

### Protected Attributes inherited from [Digikam::ItemViewImportDelegate](#)

- [ItemViewImportDelegatePrivate](#) \*const `d_ptr` = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > `m_overlays`

## 6.842.1 Member Function Documentation

### 6.842.1.1 `updateRects()`

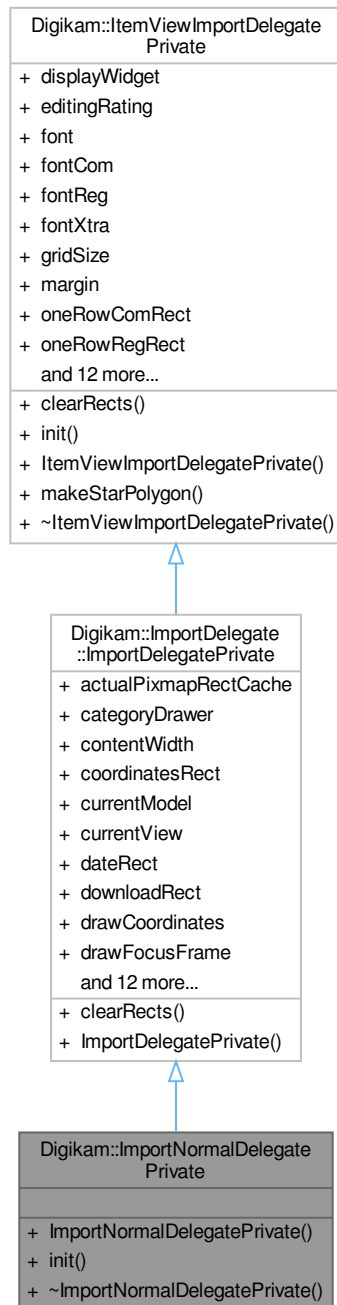
```
void Digikam::ImportNormalDelegate::updateRects ( ) [override], [protected], [virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The `paint()` method operates depending on these rects.

Implements [Digikam::ImportDelegate](#).

## 6.843 Digikam::ImportNormalDelegatePrivate Class Reference

Inheritance diagram for Digikam::ImportNormalDelegatePrivate:



### Public Member Functions

- void **init** ([ImportNormalDelegate](#) \*const q, [ImportCategorizedView](#) \*const parent)



## Public Member Functions inherited from [Digikam::ImportDelegate::ImportDelegatePrivate](#)

- void [clearRects](#) () override  
*Resets cached rects. Remember to reimplement in subclass for added rects.*

## Public Member Functions inherited from [Digikam::ItemViewImportDelegatePrivate](#)

- void [init](#) ([ItemViewImportDelegate](#) \*const \_q, [QWidget](#) \*const \_widget)
- void [makeStarPolygon](#) ()

## Additional Inherited Members

## Public Attributes inherited from [Digikam::ImportDelegate::ImportDelegatePrivate](#)

- [QCache](#)< int, [QRect](#) > [actualPixmapRectCache](#)
- [ImportCategoryDrawer](#) \* [categoryDrawer](#) = nullptr
- int [contentWidth](#) = 0
- [QRect](#) [coordinatesRect](#)
- [QAbstractItemModel](#) \* [currentModel](#) = nullptr
- [ImportCategorizedView](#) \* [currentView](#) = nullptr
- [QRect](#) [dateRect](#)
- [QRect](#) [downloadRect](#)
- bool [drawCoordinates](#) = false
- bool [drawFocusFrame](#) = true
- bool [drawImageFormat](#) = false
- bool [drawMouseOverFrame](#) = true
- [QRect](#) [groupRect](#)
- [QRect](#) [imageInformationRect](#)
- [QRect](#) [lockRect](#)
- [QRect](#) [nameRect](#)
- [QRect](#) [pickLabelRect](#)
- [QRect](#) [pixmapRect](#)
- bool [ratingOverThumbnail](#) = false
- [QRect](#) [resolutionRect](#)
- [QRect](#) [sizeRect](#)
- [QRect](#) [tagRect](#)

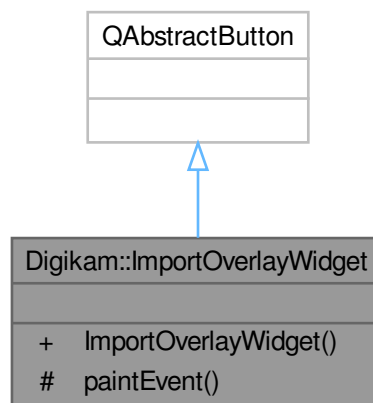
## Public Attributes inherited from [Digikam::ItemViewImportDelegatePrivate](#)

- [QWidget](#) \* [displayWidget](#) = nullptr
- [QPersistentModelIndex](#) [editingRating](#)
- [QFont](#) [font](#)
- [QFont](#) [fontCom](#)
- [QFont](#) [fontReg](#)
- [QFont](#) [fontXtra](#)
- [QSize](#) [gridSize](#)
- int [margin](#) = 5
- [QRect](#) [oneRowComRect](#)
- [QRect](#) [oneRowRegRect](#)
- [QRect](#) [oneRowXtraRect](#)

- [ItemViewImportDelegate](#) \* **q** = nullptr
- int **radius** = 3
  - constant values for drawing*
- QVector< QPixmap > **ratingPixmap** = QVector< QPixmap >(10)
- QRect **ratingRect**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- QPolygon **starPolygon**
- QSize **starPolygonSize**
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

## 6.844 Digikam::ImportOverlayWidget Class Reference

Inheritance diagram for Digikam::ImportOverlayWidget:



### Public Member Functions

- **ImportOverlayWidget** (QWidget \*const parent=nullptr)

### Protected Member Functions

- void **paintEvent** (QPaintEvent \*) override

## 6.845 Digikam::ImportPreviewView Class Reference

Inheritance diagram for Digikam::ImportPreviewView:



### Public Types

- enum **Mode** { **IconViewPreview** }

## Signals

- void **signalAssignColorLabel** (int)
- void **signalAssignPickLabel** (int)
- void **signalAssignRating** (int)
- void **signalDeleteItem** ()
- void **signalEscapePreview** ()
- void **signalNextItem** ()
- void **signalPreviewLoaded** (bool success)
- void **signalPrevItem** ()

## Signals inherited from [Digikam::GraphicsDImgView](#)

- void **activated** ()
- void **contentsMoved** (bool panningFinished)
- void **contentsMoving** (int, int)
- void **leftButtonClicked** ()
- void **leftButtonDoubleClicked** ()
- void **resized** ()
- void **rightButtonClicked** ()
- void **toNextImage** ()
- void **toPreviousImage** ()
- void **viewportRectChanged** (const QRectF &viewportRect)

## Public Member Functions

- [CamItemInfo](#) **getCamItemInfo** () const
- **ImportPreviewView** (QWidget \*const parent, Mode mode=IconViewPreview)
- void **reload** ()
- void **setCamItemInfo** (const [CamItemInfo](#) &info=[CamItemInfo](#)(), const [CamItemInfo](#) &previous=[CamItemInfo](#)(), const [CamItemInfo](#) &next=[CamItemInfo](#)())
- void **setCamItemPath** (const QString &path=QString())
- void **setPreviousNextPaths** (const QString &previous, const QString &next)
- void **showContextMenu** (const [CamItemInfo](#) &info, QGraphicsSceneContextMenuEvent \*event)

## Public Member Functions inherited from [Digikam::GraphicsDImgView](#)

- int **contentsX** () const
- int **contentsY** () const
- void **drawText** (QPainter \*p, const QRectF &rect, const QString &text)
- void **fitToWindow** ()
- **GraphicsDImgView** (QWidget \*const parent=nullptr)
- [GraphicsDImgItem](#) \* **item** () const
- [SinglePhotoPreviewLayout](#) \* **layout** () const
- [DImgPreviewItem](#) \* **previewItem** () const
- void **scrollPointOnPoint** (const QPointF &scenePos, const QPoint &viewportPos)
- void **setContentsPos** (int x, int y)
- void **setItem** ([GraphicsDImgItem](#) \*const item)
- void **toggleFullScreen** (bool set)
- QRect **visibleArea** () const

### Protected Member Functions

- bool [acceptsMouseClicked](#) (QMouseEvent \*e) override
- void [enterEvent](#) (QEnterEvent \*) override
- void [leaveEvent](#) (QEvent \*e) override
- void [showEvent](#) (QShowEvent \*e) override

### Protected Member Functions inherited from [Digikam::GraphicsDImgView](#)

- void [continuePanning](#) (const QPoint &pos)
- void [drawForeground](#) (QPainter \*painter, const QRectF &rect) override
- void [finishPanning](#) ()
- void [installPanIcon](#) ()
- void [mouseDoubleClickEvent](#) (QMouseEvent \*) override
- void [mouseMoveEvent](#) (QMouseEvent \*) override
- void [mousePressEvent](#) (QMouseEvent \*) override
- void [mouseReleaseEvent](#) (QMouseEvent \*) override
- void [resizeEvent](#) (QResizeEvent \*) override
- void [scrollContentsBy](#) (int dx, int dy) override
- void [setScaleFitToWindow](#) (bool value)
- void [setShowText](#) (bool value)
- void [startPanning](#) (const QPoint &pos)
- void [wheelEvent](#) (QWheelEvent \*) override

### Additional Inherited Members

### Protected Slots inherited from [Digikam::GraphicsDImgView](#)

- void [slotContentsMoved](#) ()
- void [slotCornerButtonPressed](#) ()
- void [slotPanIconHidden](#) ()
- virtual void [slotPanIconSelectionMoved](#) (const QRect &, bool)

## 6.845.1 Member Function Documentation

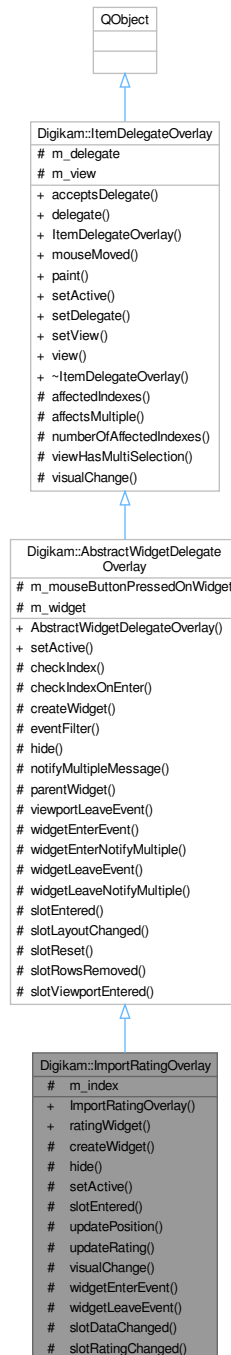
### 6.845.1.1 [acceptsMouseClicked\(\)](#)

```
bool Digikam::ImportPreviewView::acceptsMouseClicked (
    QMouseEvent * e ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::GraphicsDImgView](#).

## 6.846 Digikam::ImportRatingOverlay Class Reference

Inheritance diagram for Digikam::ImportRatingOverlay:



### Signals

- void **ratingEdited** (const QList< QModelIndex > &indexes, int rating)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- **ImportRatingOverlay** (QObject \*const parent)
- [RatingWidget](#) \* **ratingWidget** () const

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotDataChanged** (const QModelIndex &, const QModelIndex &)
- void **slotRatingChanged** (int)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

## Protected Member Functions

- QWidget \* **createWidget** () override
- void **hide** () override
- void **setActive** (bool) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **updateRating** ()
- void **visualChange** () override
- void **widgetEnterEvent** () override
- void **widgetLeaveEvent** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual bool [checkIndex](#) (const QModelIndex &index) const
- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- void [widgetLeaveNotifyMultiple](#) ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

## Protected Attributes

- QPersistentModelIndex [m\\_index](#)

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.846.1 Member Function Documentation

### 6.846.1.1 [createWidget\(\)](#)

```
QWidget * Digikam::ImportRatingOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.846.1.2 [hide\(\)](#)

```
void Digikam::ImportRatingOverlay::hide ( ) [override], [protected], [virtual]
```

Called when the widget shall be hidden (mouse cursor left index, viewport, uninstalled etc.). Default implementation [hide\(\)](#)s [m\\_widget](#).

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).



### 6.846.1.3 setActive()

```
void Digikam::ImportRatingOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.846.1.4 slotEntered()

```
void Digikam::ImportRatingOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.846.1.5 visualChange()

```
void Digikam::ImportRatingOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

### 6.846.1.6 widgetEnterEvent()

```
void Digikam::ImportRatingOverlay::widgetEnterEvent ( ) [override], [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

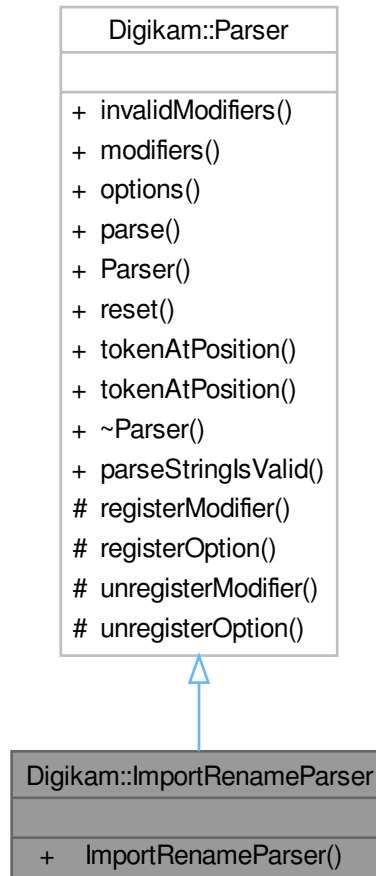
### 6.846.1.7 widgetLeaveEvent()

```
void Digikam::ImportRatingOverlay::widgetLeaveEvent ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.847 Digikam::ImportRenameParser Class Reference

Inheritance diagram for Digikam::ImportRenameParser:



### Additional Inherited Members

### Public Member Functions inherited from [Digikam::Parser](#)

- [ParseResults](#) `invalidModifiers` ([ParseSettings](#) &settings)
- RulesList `modifiers` () const
- RulesList `options` () const
- QString `parse` ([ParseSettings](#) &settings)
- void `reset` ()
- bool `tokenAtPosition` ([ParseSettings](#) &settings, int pos)
- bool `tokenAtPosition` ([ParseSettings](#) &settings, int pos, int &start, int &length)

### Static Public Member Functions inherited from [Digikam::Parser](#)

- static bool `parseStringIsValid` (const QString &str)

## Protected Member Functions inherited from Digikam::Parser

- void **registerModifier** (Rule \*modifier)
- void **registerOption** (Rule \*option)
- void **unregisterModifier** (const Rule \*modifier)
- void **unregisterOption** (const Rule \*option)

## 6.848 Digikam::ImportRotateOverlay Class Reference

Inheritance diagram for Digikam::ImportRotateOverlay:



## Signals

- void **signalRotate** (const QList< QModelIndex > &indexes)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- ImportRotateOverlayDirection **direction** () const
- **ImportRotateOverlay** (ImportRotateOverlayDirection dir, QObject \*const parent)
- bool **isLeft** () const
- bool **isRight** () const
- void **setActive** (bool active) override

## Public Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- [ItemViewHoverButton](#) \* **button** () const
- **HoverButtonDelegateOverlay** (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Static Public Member Functions

- static [ImportRotateOverlay](#) \* **left** (QObject \*const parent)
- static [ImportRotateOverlay](#) \* **right** (QObject \*const parent)

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- [ItemViewHoverButton](#) \* **createButton** () override
- void **updateButton** (const QModelIndex &index) override
- void **widgetEnterEvent** () override
- void **widgetLeaveEvent** () override

**Protected Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)**

- QWidget \* [createWidget](#) () override
- void [visualChange](#) () override

**Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- void [widgetLeaveNotifyMultiple](#) ()

**Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)**

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

**Additional Inherited Members****Protected Slots inherited from [Digikam::HoverButtonDelegateOverlay](#)**

- void [slotEntered](#) (const QModelIndex &index) override
- void [slotReset](#) () override

**Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- virtual void [slotEntered](#) (const QModelIndex &index)
- virtual void [slotLayoutChanged](#) ()
- virtual void [slotReset](#) ()
- virtual void [slotRowsRemoved](#) (const QModelIndex &parent, int start, int end)
- virtual void [slotViewportEntered](#) ()

**Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)****Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

**Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)**

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.848.1 Member Function Documentation

### 6.848.1.1 checkIndex()

```
bool Digikam::ImportRotateOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.848.1.2 createButton()

```
ItemViewHoverButton * Digikam::ImportRotateOverlay::createButton ( ) [override], [protected],
[virtual]
```

Create your widget here. Pass view() as parent.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.848.1.3 setActive()

```
void Digikam::ImportRotateOverlay::setActive (
    bool active ) [override], [virtual]
```

Will call [createButton\(\)](#).

Reimplemented from [Digikam::HoverButtonDelegateOverlay](#).

### 6.848.1.4 updateButton()

```
void Digikam::ImportRotateOverlay::updateButton (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.848.1.5 widgetEnterEvent()

```
void Digikam::ImportRotateOverlay::widgetEnterEvent ( ) [override], [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.848.1.6 widgetLeaveEvent()

```
void Digikam::ImportRotateOverlay::widgetLeaveEvent ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.849 Digikam::ImportRotateOverlayButton Class Reference

Inheritance diagram for Digikam::ImportRotateOverlayButton:



### Public Member Functions

- **ImportRotateOverlayButton** (ImportRotateOverlayDirection dir, QAbstractItemView \*const parentView)
- QSize [sizeHint](#) () const override

### Public Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- QModelIndex [index](#) () const
- void [initIcon](#) ()
- **ItemViewHoverButton** (QAbstractItemView \*const parentView)
- void [reset](#) ()
- void [setIndex](#) (const QModelIndex &index)
- void [setVisible](#) (bool visible) override

### Protected Member Functions

- QIcon [icon](#) () override
- void [updateToolTip](#) () override

### Protected Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- void [enterEvent](#) (QEnterEvent \*event)
- void [leaveEvent](#) (QEvent \*event)
- void [paintEvent](#) (QPaintEvent \*event)
- void [setup](#) ()

### Protected Attributes

- ImportRotateOverlayDirection const [m\\_direction](#)

### Protected Attributes inherited from [Digikam::ItemViewHoverButton](#)

- QTimerLine \* [m\\_fadingTimeLine](#) = nullptr
- int [m\\_fadingValue](#) = 0
- QIcon [m\\_icon](#)
- QPersistentModelIndex [m\\_index](#)
- bool [m\\_isHovered](#) = false

### Additional Inherited Members

### Protected Slots inherited from [Digikam::ItemViewHoverButton](#)

- void [refreshIcon](#) ()
- void [setFadingValue](#) (int value)
- void [startFading](#) ()
- void [stopFading](#) ()



## 6.849.1 Member Function Documentation

### 6.849.1.1 icon()

```
QIcon Digikam::ImportRotateOverlayButton::icon ( ) [override], [protected], [virtual]
```

Return your icon here. Will be queried again on toggle.

Implements [Digikam::ItemViewHoverButton](#).

### 6.849.1.2 sizeHint()

```
QSize Digikam::ImportRotateOverlayButton::sizeHint ( ) const [override], [virtual]
```

Reimplement to match the size of your icon

Implements [Digikam::ItemViewHoverButton](#).

### 6.849.1.3 updateToolTip()

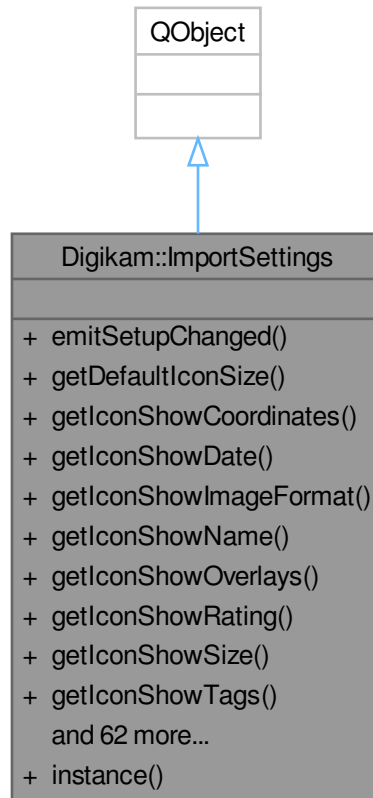
```
void Digikam::ImportRotateOverlayButton::updateToolTip ( ) [override], [protected], [virtual]
```

Optionally update tooltip here. Will be called again on state change.

Reimplemented from [Digikam::ItemViewHoverButton](#).

## 6.850 Digikam::ImportSettings Class Reference

Inheritance diagram for Digikam::ImportSettings:



### Public Types

- enum `ItemLeftClickAction` { `ShowPreview = 0` , `StartEditor` , `OpenDefault` }

### Signals

- void `setupChanged` ()

### Public Member Functions

- void `emitSetupChanged` ()
- int `getDefaultIconSize` () const
- bool `getIconShowCoordinates` () const
- bool `getIconShowDate` () const
- bool `getIconShowImageFormat` () const
- bool `getIconShowName` () const

- bool **getIconShowOverlays** () const
- bool **getIconShowRating** () const
- bool **getIconShowSize** () const
- bool **getIconShowTags** () const
- bool **getIconShowTitle** () const
- QFont **getIconViewFont** () const
- int **getImageSeparationMode** () const
- int **getImageSortBy** () const
- int **getImageSortOrder** () const
- int **getItemLeftClickAction** () const
- bool **getPreviewItemsWhileDownload** () const
- bool **getPreviewLoadFullImageSize** () const
- bool **getPreviewShowIcons** () const
- bool **getShowThumbbar** () const
- bool **getShowToolTips** () const
- QFont **getToolTipsFont** () const
- bool **getToolTipsShowFileDate** () const
- bool **getToolTipsShowFileName** () const
- bool **getToolTipsShowFileSize** () const
- bool **getToolTipsShowImageDim** () const
- bool **getToolTipsShowImageType** () const
- bool **getToolTipsShowLabelRating** () const
- bool **getToolTipsShowPhotoExpo** () const
- bool **getToolTipsShowPhotoFlash** () const
- bool **getToolTipsShowPhotoFocal** () const
- bool **getToolTipsShowPhotoLens** () const
- bool **getToolTipsShowPhotoMake** () const
- bool **getToolTipsShowPhotoWB** () const
- bool **getToolTipsShowTags** () const
- void **readSettings** ()
- void **saveSettings** ()
- void **setDefaultIconSize** (int val)
- void **setIconShowCoordinates** (bool val)
- void **setIconShowDate** (bool val)
- void **setIconShowImageFormat** (bool val)
- void **setIconShowName** (bool val)
- void **setIconShowOverlays** (bool val)
- void **setIconShowRating** (bool val)
- void **setIconShowSize** (bool val)
- void **setIconShowTags** (bool val)
- void **setIconShowTitle** (bool val)
- void **setIconViewFont** (const QFont &font)
- void **setImageSeparationMode** (int mode)
- void **setImageSortBy** (int sortBy)
- void **setImageSortOrder** (int order)
- void **setItemLeftClickAction** (int action)
- void **setPreviewItemsWhileDownload** (bool val)
- void **setPreviewLoadFullImageSize** (bool val)
- void **setPreviewShowIcons** (bool val)
- void **setShowThumbbar** (bool val)
- void **setShowToolTips** (bool val)
- void **setToolTipsFont** (const QFont &font)
- void **setToolTipsShowFileDate** (bool val)
- void **setToolTipsShowFileName** (bool val)
- void **setToolTipsShowFileSize** (bool val)

- void **setToolTipsShowImageDim** (bool val)
- void **setToolTipsShowImageType** (bool val)
- void **setToolTipsShowLabelRating** (bool val)
- void **setToolTipsShowPhotoExpo** (bool val)
- void **setToolTipsShowPhotoFlash** (bool val)
- void **setToolTipsShowPhotoFocal** (bool val)
- void **setToolTipsShowPhotoLens** (bool val)
- void **setToolTipsShowPhotoMake** (bool val)
- void **setToolTipsShowPhotoWB** (bool val)
- void **setToolTipsShowTags** (bool val)
- bool **showToolTipsIsValid** () const

#### Static Public Member Functions

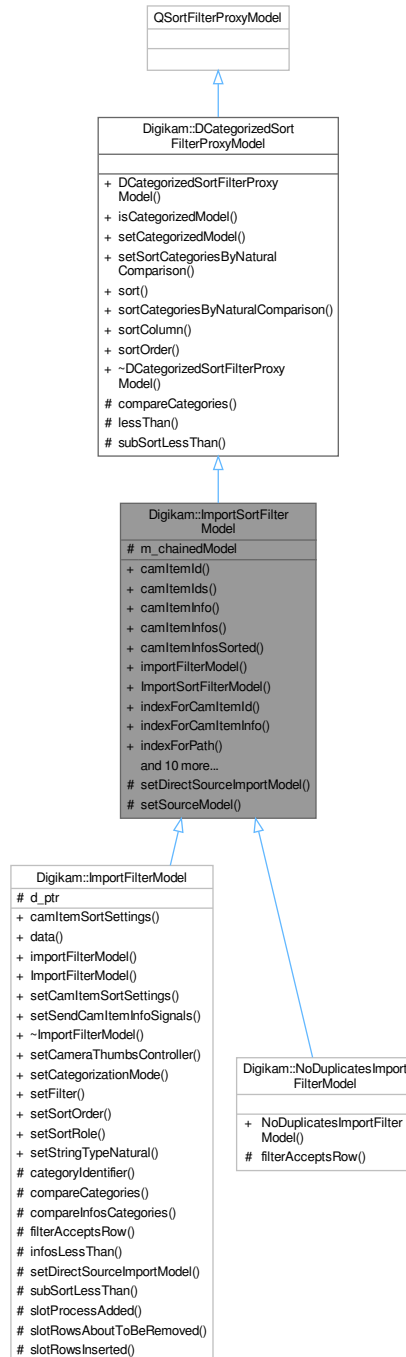
- static [ImportSettings](#) \* **instance** ()

#### Friends

- class **ImportSettingsCreator**

## 6.851 Digikam::ImportSortFilterModel Class Reference

Inheritance diagram for Digikam::ImportSortFilterModel:



### Public Member Functions

- `qulonglong camItemid` (const QModelIndex &index) const
- `QList< qulonglong > camItemids` (const QList< QModelIndex > &indexes) const

- [CamItemInfo](#) **camItemInfo** (const QModelIndex &index) const
- QList< [CamItemInfo](#) > **camItemInfos** (const QList< QModelIndex > &indexes) const
- QList< [CamItemInfo](#) > **camItemInfosSorted** () const
- virtual [ImportFilterModel](#) \* **importFilterModel** () const  
*Returns this, any chained [ImportFilterModel](#), or 0.*
- **ImportSortFilterModel** (QObject \*const parent=nullptr)
- QModelIndex **indexForCamItemId** (qulonglong id) const
- QModelIndex **indexForCamItemInfo** (const [CamItemInfo](#) &info) const
- QModelIndex **indexForPath** (const QString &filePath) const
- QModelIndex **mapFromDirectSourceToSourceImportModel** (const QModelIndex &sourceModelIndex) const
- QModelIndex **mapFromSourceImportModel** (const QModelIndex &importModelIndex) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceImportModel** (const QModelIndex &proxyIndex) const
- void **setSourceFilterModel** ([ImportSortFilterModel](#) \*const sourceModel)
- void **setSourceImportModel** ([ImportItemModel](#) \*const sourceModel)
- [ImportSortFilterModel](#) \* **sourceFilterModel** () const
- [ImportItemModel](#) \* **sourceImportModel** () const

## Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- **DCategorizedSortFilterProxyModel** (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void **setCategorizedModel** (bool categorizedModel)
- void **setSortCategoriesByNaturalComparison** (bool [sortCategoriesByNaturalComparison](#))
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool **sortCategoriesByNaturalComparison** () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

## Protected Member Functions

- virtual void **setDirectSourceImportModel** ([ImportItemModel](#) \*const sourceModel)  
*Reimplement if needed. Called only when model shall be set as (direct) sourceModel.*
- void **setSourceModel** (QAbstractItemModel \*sourceModel) override

## Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- virtual int **compareCategories** (const QModelIndex &left, const QModelIndex &right) const
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool **subSortLessThan** (const QModelIndex &left, const QModelIndex &right) const

## Protected Attributes

- [ImportSortFilterModel](#) \* **m\_chainedModel** = nullptr

## Additional Inherited Members

### Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## 6.851.1 Member Function Documentation

### 6.851.1.1 [camItemInfosSorted\(\)](#)

```
QList< CamItemInfo > Digikam::ImportSortFilterModel::camItemInfosSorted ( ) const
```

Returns a list of all camera infos, sorted according to this model. If you do not need a sorted list, use [ImportItemModel](#)'s [camItemInfo\(\)](#) method.

### 6.851.1.2 [importFilterModel\(\)](#)

```
ImportFilterModel * Digikam::ImportSortFilterModel::importFilterModel ( ) const [virtual]
```

Reimplemented in [Digikam::ImportFilterModel](#).

### 6.851.1.3 [mapToSourceImportModel\(\)](#)

```
QModelIndex Digikam::ImportSortFilterModel::mapToSourceImportModel (
    const QModelIndex & proxyIndex ) const
```

Convenience methods mapped to [ImportItemModel](#). Mentioned indexes returned come from the source import image model.

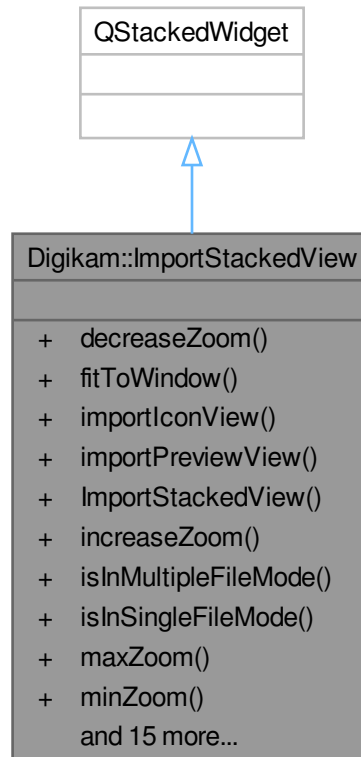
### 6.851.1.4 [setDirectSourceImportModel\(\)](#)

```
void Digikam::ImportSortFilterModel::setDirectSourceImportModel (
    ImportItemModel *const sourceModel ) [protected], [virtual]
```

Reimplemented in [Digikam::ImportFilterModel](#).

## 6.852 Digikam::ImportStackedView Class Reference

Inheritance diagram for Digikam::ImportStackedView:



### Public Types

- enum `StackedViewMode` { `PreviewCameraMode` = 0 , `PreviewImageMode` , `MediaPlayerMode` , `MapWidgetMode` }

### Signals

- void `signalEscapePreview` ()
- void `signalNextItem` ()
- void `signalPrevItem` ()
- void `signalViewModeChanged` ()
- void `signalZoomFactorChanged` (double)



## Public Member Functions

- void **decreaseZoom** ()
- void **fitToWindow** ()
- [ImportIconView](#) \* **importIconView** () const
- [ImportPreviewView](#) \* **importPreviewView** () const
- **ImportStackedView** (QWidget \*const parent=nullptr)
- void **increaseZoom** ()
- bool **isInMultipleFileMode** () const
- bool **isInSingleFileMode** () const
- bool **maxZoom** () const
- bool **minZoom** () const
- void **previewLoaded** ()
- void **setDockArea** (QMainWindow \*)
- void **setPreviewItem** (const [CamItemInfo](#) &info=[CamItemInfo](#)(), const [CamItemInfo](#) &previous=[CamItemInfo](#)(), const [CamItemInfo](#) &next=[CamItemInfo](#)())
- void **setViewMode** (const [StackedViewMode](#) mode)
- void **setZoomFactor** (double z)
- void **setZoomFactorSnapped** (double z)
- [ImportThumbnailBar](#) \* **thumbBar** () const
- [ThumbBarDock](#) \* **thumbBarDock** () const
- void **toggleFitToWindowOr100** ()
- [StackedViewMode](#) **viewMode** () const
- double **zoomFactor** () const
- double **zoomMax** () const
- double **zoomMin** () const
- void **zoomTo100Percents** ()

## 6.852.1 Member Enumeration Documentation

### 6.852.1.1 StackedViewMode

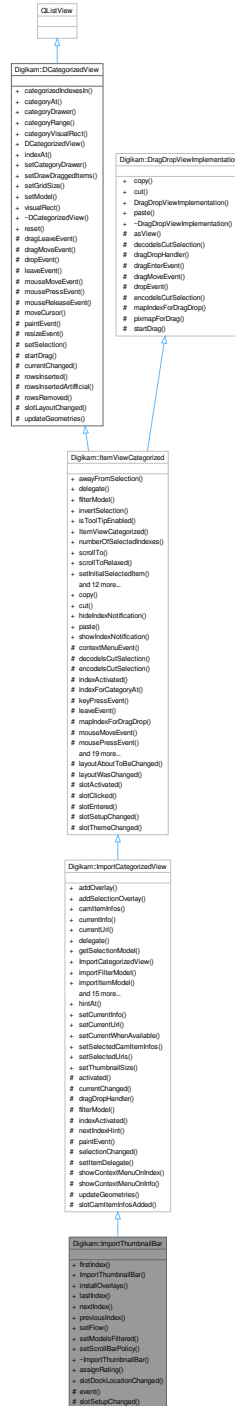
```
enum Digikam::ImportStackedView::StackedViewMode
```

#### Enumerator

PreviewCameraMode	previewing the set of items on the camera
-------------------	---

## 6.853 Digikam::ImportThumbnailBar Class Reference

Inheritance diagram for Digikam::ImportThumbnailBar:



### Public Slots

- void **assignRating** (const QList< QModelIndex > &index, int rating)
- void **slotDockLocationChanged** (Qt::DockWidgetArea area)

## Public Slots inherited from [Digikam::ImportCategorizedView](#)

- void [hintAt](#) (const [CamItemInfo](#) &info)
- void [setCurrentInfo](#) (const [CamItemInfo](#) &info)
- void [setCurrentUrl](#) (const [QUrl](#) &url)
- void [setCurrentWhenAvailable](#) (qulonglong [camItemId](#))
- void [setSelectedCamItemInfos](#) (const [QList](#)< [CamItemInfo](#) > &infos)
- void [setSelectedUrls](#) (const [QList](#)< [QUrl](#) > &urlList)
- void [setThumbnailSize](#) (int size)

## Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void [copy](#) () override
- void [cut](#) () override
- void [hideIndexNotification](#) ()
- void [paste](#) () override
- void [showIndexNotification](#) (const [QModelIndex](#) &index, const [QString](#) &message)

## Public Slots inherited from [Digikam::DCategorizedView](#)

- void [reset](#) () override

## Public Member Functions

- [QModelIndex](#) [firstIndex](#) () const
- [ImportThumbnailBar](#) ([QWidget](#) \*const parent=nullptr)
- void [installOverlays](#) ()
- [QModelIndex](#) [lastIndex](#) () const
- [QModelIndex](#) [nextIndex](#) (const [QModelIndex](#) &index) const
- [QModelIndex](#) [previousIndex](#) (const [QModelIndex](#) &index) const
- void [setFlow](#) ([QListView::Flow](#) newFlow)
- void [setModelsFiltered](#) ([ImportItemModel](#) \*model, [ImportSortFilterModel](#) \*filterModel)
- void [setScrollBarPolicy](#) ([Qt::ScrollBarPolicy](#) policy)

## Public Member Functions inherited from [Digikam::ImportCategorizedView](#)

- void [addOverlay](#) ([ItemDelegateOverlay](#) \*overlay, [ImportDelegate](#) \*delegate=nullptr)
- void [addSelectionOverlay](#) ([ImportDelegate](#) \*delegate=nullptr)
- [QList](#)< [CamItemInfo](#) > [camItemInfos](#) () const
- [CamItemInfo](#) [currentInfo](#) () const
- [QUrl](#) [currentUrl](#) () const
- [ImportDelegate](#) \* [delegate](#) () const
- [QItemSelectionModel](#) \* [getSelectionModel](#) () const
- [ImportCategorizedView](#) ([QWidget](#) \*const parent=nullptr)
- [ImportFilterModel](#) \* [importFilterModel](#) () const
- [ImportItemModel](#) \* [importItemModel](#) () const
- [ImportSortFilterModel](#) \* [importSortFilterModel](#) () const
- [ImportThumbnailModel](#) \* [importThumbnailModel](#) () const
- [CamItemInfo](#) [nextInfo](#) (const [CamItemInfo](#) &info)
- [CamItemInfo](#) [nextInOrder](#) (const [CamItemInfo](#) &startingPoint, int nth)
- [CamItemInfo](#) [previousInfo](#) (const [CamItemInfo](#) &info)

- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [QList](#)< [CamItemInfo](#) > **selectedCamItemInfos** () const
- [QList](#)< [CamItemInfo](#) > **selectedCamItemInfosCurrentFirst** () const
- [QList](#)< [QUrl](#) > **selectedUrls** () const
- void **setModels** ([ImportItemModel](#) \*model, [ImportSortFilterModel](#) \*filterModel)
- virtual void **setThumbnailSize** (const [ThumbnailSize](#) &size)
- [ThumbnailSize](#) **thumbnailSize** () const
- void **toIndex** (const [QUrl](#) &url)
- [QList](#)< [QUrl](#) > **urls** () const

## Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **awayFromSelection** ()
- [DItemDelegate](#) \* **delegate** () const
- void **invertSelection** ()
- bool **isToolTipEnabled** () const
- [ItemViewCategorized](#) ([QWidget](#) \*const parent=nullptr)
- int **numberOfSelectedIndexes** () const
- void **scrollTo** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible) override
- void **scrollToRelaxed** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible)
- void **setInitialSelectedItem** (bool enabled)
- void **setScrollCurrentToCenter** (bool enabled)
- void **setScrollStepGranularity** (int factor)
- void **setSelectedIndexes** (const [QList](#)< [QModelIndex](#) > &indexes)
- void **setSpacing** (int spacing)
- void **setToolTipEnabled** (bool enabled)
- void **setUsePointingHandCursor** (bool useCursor)
- void **toFirstIndex** ()
- void **toIndex** (const [QModelIndex](#) &index)
- void **toLastIndex** ()
- void **toNextIndex** ()
- void **toPreviousIndex** ()

## Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual [QModelIndexList](#) **categorizedIndexesIn** (const [QRect](#) &rect) const
- virtual [QModelIndex](#) **categoryAt** (const [QPoint](#) &point) const
- [DCategoryDrawer](#) \* **categoryDrawer** () const
- virtual [QItemSelectionRange](#) **categoryRange** (const [QModelIndex](#) &index) const
- virtual [QRect](#) **categoryVisualRect** (const [QModelIndex](#) &index) const
- [DCategorizedView](#) ([QWidget](#) \*const parent=nullptr)
- [QModelIndex](#) **indexAt** (const [QPoint](#) &point) const override
- void **setCategoryDrawer** ([DCategoryDrawer](#) \*categoryDrawer)
- void **setDrawDraggedItems** (bool drawDraggedItems)
- void **setGridSize** (const [QSize](#) &size)
- void **setModel** ([QAbstractItemModel](#) \*model) override
- [QRect](#) **visualRect** (const [QModelIndex](#) &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

**Protected Member Functions**

- bool **event** (QEvent \*) override
- void **slotSetupChanged** () override

**Protected Member Functions inherited from Digikam::ImportCategorizedView**

- virtual void **activated** (const CamItemInfo &info, Qt::KeyboardModifiers modifiers)
  - Reimplement these in a subclass.*
- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- AbstractItemDragDropHandler \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override
  - reimplemented from parent class*
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **paintEvent** (QPaintEvent \*e) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** (ImportDelegate \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override
  - Reimplement these in a subclass.*
- virtual void **showContextMenuOnInfo** (QContextMenuEvent \*event, const CamItemInfo &info)
- void **updateGeometries** () override

**Protected Member Functions inherited from Digikam::ItemViewCategorized**

- void **contextMenuEvent** (QContextMenuEvent \*event) override
  - reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** (DItemDelegate \*delegate)
- void **setToolTip** (ItemViewToolTip \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

### Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

### Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

### Additional Inherited Members

### Signals inherited from [Digikam::ImportCategorizedView](#)

- void **camItemInfoActivated** (const CamItemInfo &info)
- void **currentChanged** (const CamItemInfo &info)
- void **deselected** (const QList< CamItemInfo > &nowDeselectedInfos)
- void **modelChanged** ()
- void **selected** (const QList< CamItemInfo > &newSelectedInfos)

### Signals inherited from [Digikam::ItemViewCategorized](#)

- void **clicked** (const QMouseEvent \*e, const QModelIndex &index)
- void **entered** (const QMouseEvent \*e, const QModelIndex &index)
- void **keyPressed** (QKeyEvent \*e)
- void **selectionChanged** ()
- void **selectionCleared** ()
- void **viewportClicked** (const QMouseEvent \*e)
- void **zoomInStep** ()
- void **zoomOutStep** ()

### Protected Slots inherited from [Digikam::ImportCategorizedView](#)

- void **slotCamItemInfosAdded** ()

## Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void **layoutAboutToBeChanged** ()
- void **layoutWasChanged** ()
- void **slotActivated** (const QModelIndex &index)
- void **slotClicked** (const QModelIndex &index)
- void **slotEntered** (const QModelIndex &index)
- virtual void **slotThemeChanged** ()

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## 6.853.1 Member Function Documentation

### 6.853.1.1 `setModelsFiltered()`

```
void Digikam::ImportThumbnailBar::setModelsFiltered (
    ImportItemModel * model,
    ImportSortFilterModel * filterModel )
```

This installs a duplicate filter model, if the [ImportItemModel](#) may contain duplicates. Otherwise, just use `setModels()`.

### 6.853.1.2 `setScrollBarPolicy()`

```
void Digikam::ImportThumbnailBar::setScrollBarPolicy (
    Qt::ScrollBarPolicy policy )
```

Sets the policy always for the one scroll bar which is relevant, depending on orientation

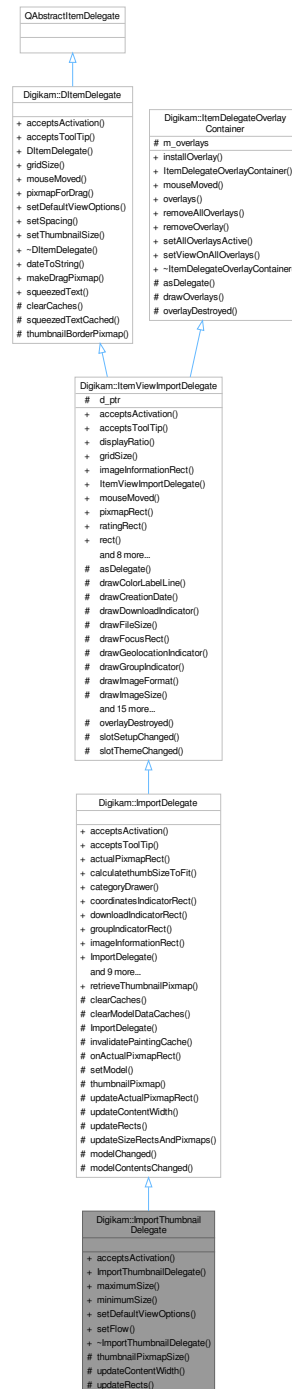
### 6.853.1.3 `slotSetupChanged()`

```
void Digikam::ImportThumbnailBar::slotSetupChanged ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

## 6.854 Digikam::ImportThumbnailDelegate Class Reference

Inheritance diagram for Digikam::ImportThumbnailDelegate:



### Public Member Functions

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect) const override



- **ImportThumbnailDelegate** ([ImportCategorizedView](#) \*const parent)
- int **maximumSize** () const
- int **minimumSize** () const
- void **setDefaultViewOptions** (const [QStyleOptionViewItem](#) &option) override
- void **setFlow** ([QListView::Flow](#) flow)

## Public Member Functions inherited from [Digikam::ImportDelegate](#)

- bool **acceptsToolTip** (const [QPoint](#) &pos, const [QRect](#) &visualRect, const [QModelIndex](#) &index, [QRect](#) \*tooltipRect=nullptr) const override
- [QRect](#) **actualPixmapRect** (const [QModelIndex](#) &index) const
- int **calculatethumbSizeToFit** (int ws)
- [ImportCategoryDrawer](#) \* **categoryDrawer** () const
- [QRect](#) **coordinatesIndicatorRect** () const
- [QRect](#) **downloadIndicatorRect** () const
- [QRect](#) **groupIndicatorRect** () const
- [QRect](#) **imageInformationRect** () const override
- **ImportDelegate** ([QWidget](#) \*const parent)
- [QRect](#) **lockIndicatorRect** () const
- void **paint** ([QPainter](#) \*painter, const [QStyleOptionViewItem](#) &option, const [QModelIndex](#) &index) const override
- [QPixmap](#) **pixmapForDrag** (const [QStyleOptionViewItem](#) &option, const [QList](#)< [QModelIndex](#) > &indexes) const override
- [QRect](#) **pixmapRect** () const override
- void **setSpacing** (int spacing) override
- void **setView** ([ImportCategorizedView](#) \*view)
- [QRect](#) **tagsRect** () const

## Public Member Functions inherited from [Digikam::ItemViewImportDelegate](#)

- bool **acceptsActivation** (const [QPoint](#) &pos, const [QRect](#) &visualRect, const [QModelIndex](#) &index, [QRect](#) \*activationRect=nullptr) const override
- bool **acceptsToolTip** (const [QPoint](#) &pos, const [QRect](#) &visualRect, const [QModelIndex](#) &index, [QRect](#) \*tooltipRect=nullptr) const override
- double **displayRatio** () const
- [QSize](#) **gridSize** () const override
- **ItemViewImportDelegate** ([QWidget](#) \*const parent)
- void **mouseMoved** ([QMouseEvent](#) \*e, const [QRect](#) &visualRect, const [QModelIndex](#) &index) override
- virtual [QRect](#) **ratingRect** () const
- [QRect](#) **rect** () const
- void **setDefaultViewOptions** (const [QStyleOptionViewItem](#) &option) override
- void **setRatingEdited** (const [QModelIndex](#) &index)
- void **setSpacing** (int spacing) override
- void **setThumbnailSize** (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from [DItemDelegate](#)*
- [QSize](#) **sizeHint** (const [QStyleOptionViewItem](#) &option, const [QModelIndex](#) &index) const override
- int **spacing** () const
- [ThumbnailSize](#) **thumbnailSize** () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- **DItemDelegate** ([QObject](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Protected Member Functions

- int **thumbnailPixmapSize** (bool withHighlight, int size)
- void **updateContentWidth** () override
- void **updateRects** () override

## Protected Member Functions inherited from [Digikam::ImportDelegate](#)

- void **clearCaches** () override
- virtual void **clearModelDataCaches** ()
- **ImportDelegate** ([ImportDelegate::ImportDelegatePrivate](#) &dd, QWidget \*const parent)
- void **invalidatePaintingCache** () override  
*reimplement these in subclasses*
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- void **updateSizeRectsAndPxmmaps** () override

## Protected Member Functions inherited from [Digikam::ItemViewImportDelegate](#)

- QAbstractItemDelegate \* **asDelegate** () override  
*Returns the delegate, typically, the derived class.*
- void **drawColorLabelLine** (QPainter \*p, const QRect &pixRect, int colorId) const
- void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawDownloadIndicator** (QPainter \*p, const QRect &r, int itemType) const
- void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
- void **drawGroupIndicator** (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
- void **drawImageFormat** (QPainter \*p, const QRect &dimsRect, const QString &mime) const
- void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawLockIndicator** (QPainter \*p, const QRect &r, int lockStatus) const
- void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
- void **drawPickLabelIcon** (QPainter \*p, const QRect &r, int pickLabel) const
- void **drawRating** (QPainter \*p, const QModelIndex &index, const QRect &ratingRect, int rating, bool isSelected) const
- void **drawTags** (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const

- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail) const

*Use the tool methods for painting in subclasses.*

- **ItemViewImportDelegate** ([ItemViewImportDelegatePrivate](#) &dd, QWidget \*const parent)
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- void **prepareRatingPixmap** (bool composeOverBackground=true)
- QPixmap **ratingPixmap** (int rating, bool selected) const

*Returns the relevant pixmap from the cached rating pixmaps.*

### Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)

*Declare as slot in the derived class calling this method.*

### Additional Inherited Members

### Signals inherited from [Digikam::ItemViewImportDelegate](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

### Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

### Static Public Member Functions inherited from [Digikam::ImportDelegate](#)

- static QPixmap **retrieveThumbnailPixmap** (const QModelIndex &index, int thumbnailSize)

### Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

## Protected Slots inherited from [Digikam::ImportDelegate](#)

- void `modelChanged` ()
- void `modelContentsChanged` ()

## Protected Slots inherited from [Digikam::ItemViewImportDelegate](#)

- void `overlayDestroyed` (QObject \*o) override
- void `slotSetupChanged` ()
- void `slotThemeChanged` ()

## Protected Attributes inherited from [Digikam::ItemViewImportDelegate](#)

- [ItemViewImportDelegatePrivate](#) \*const `d_ptr` = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > `m_overlays`

## 6.854.1 Member Function Documentation

### 6.854.1.1 `acceptsActivation()`

```
bool Digikam::ImportThumbnailDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect ) const [override], [virtual]
```

Reimplemented from [Digikam::ImportDelegate](#).

### 6.854.1.2 `maximumSize()`

```
int Digikam::ImportThumbnailDelegate::maximumSize ( ) const
```

Returns the minimum or maximum viewport size in the limiting dimension, width or height, depending on current flow.

### 6.854.1.3 `setDefaultViewOptions()`

```
void Digikam::ImportThumbnailDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. `option.rect` shall be the viewport rectangle. Call on resize, font change.

Reimplemented from [Digikam::ImportDelegate](#).

#### 6.854.1.4 updateContentWidth()

```
void Digikam::ImportThumbnailDelegate::updateContentWidth ( ) [override], [protected], [virtual]
```

Reimplement this to set contentWidth. This is the maximum width of all content rectangles, typically excluding margins on both sides.

Reimplemented from [Digikam::ImportDelegate](#).

#### 6.854.1.5 updateRects()

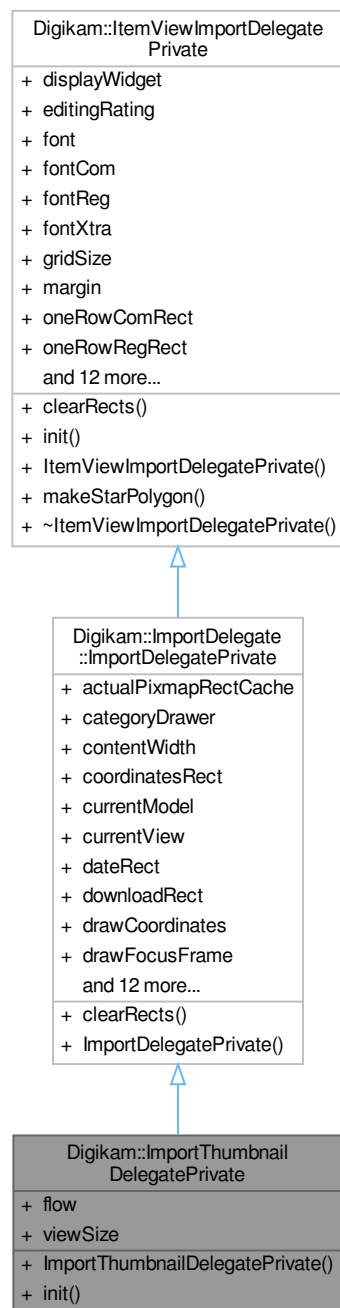
```
void Digikam::ImportThumbnailDelegate::updateRects ( ) [override], [protected], [virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The paint() method operates depending on these rects.

Implements [Digikam::ImportDelegate](#).

## 6.855 Digikam::ImportThumbnailDelegatePrivate Class Reference

Inheritance diagram for Digikam::ImportThumbnailDelegatePrivate:



### Public Member Functions

- [ImportThumbnailDelegatePrivate](#) ()
- void **init** ([ImportThumbnailDelegate](#) \*const q)

## Public Member Functions inherited from [Digikam::ImportDelegate::ImportDelegatePrivate](#)

- void `clearRects` () override  
*Resets cached rects. Remember to reimplement in subclass for added rects.*

## Public Member Functions inherited from [Digikam::ItemViewImportDelegatePrivate](#)

- void `init` ([ItemViewImportDelegate](#) \*const \_q, [QWidget](#) \*const \_widget)
- void `makeStarPolygon` ()

## Public Attributes

- [QListView::Flow](#) `flow` = [QListView::LeftToRight](#)
- [QRect](#) `viewSize`

## Public Attributes inherited from [Digikam::ImportDelegate::ImportDelegatePrivate](#)

- [QCache](#)< int, [QRect](#) > `actualPixmapRectCache`
- [ImportCategoryDrawer](#) \* `categoryDrawer` = nullptr
- int `contentWidth` = 0
- [QRect](#) `coordinatesRect`
- [QAbstractItemModel](#) \* `currentModel` = nullptr
- [ImportCategorizedView](#) \* `currentView` = nullptr
- [QRect](#) `dateRect`
- [QRect](#) `downloadRect`
- bool `drawCoordinates` = false
- bool `drawFocusFrame` = true
- bool `drawImageFormat` = false
- bool `drawMouseOverFrame` = true
- [QRect](#) `groupRect`
- [QRect](#) `imageInformationRect`
- [QRect](#) `lockRect`
- [QRect](#) `nameRect`
- [QRect](#) `pickLabelRect`
- [QRect](#) `pixmapRect`
- bool `ratingOverThumbnail` = false
- [QRect](#) `resolutionRect`
- [QRect](#) `sizeRect`
- [QRect](#) `tagRect`

## Public Attributes inherited from [Digikam::ItemViewImportDelegatePrivate](#)

- QWidget \* **displayWidget** = nullptr
- QPersistentModelIndex **editingRating**
- QFont **font**
- QFont **fontCom**
- QFont **fontReg**
- QFont **fontXtra**
- QSize **gridSize**
- int **margin** = 5
- QRect **oneRowComRect**
- QRect **oneRowRegRect**
- QRect **oneRowXtraRect**
- [ItemViewImportDelegate](#) \* **q** = nullptr
- int **radius** = 3
- constant values for drawing*
- QVector< QPixmap > **ratingPixmaps** = QVector< QPixmap >(10)
- QRect **ratingRect**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- QPolygon **starPolygon**
- QSize **starPolygonSize**
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

### 6.855.1 Constructor & Destructor Documentation

#### 6.855.1.1 ImportThumbnailDelegatePrivate()

```
Digikam::ImportThumbnailDelegatePrivate::ImportThumbnailDelegatePrivate ( ) [inline]
```

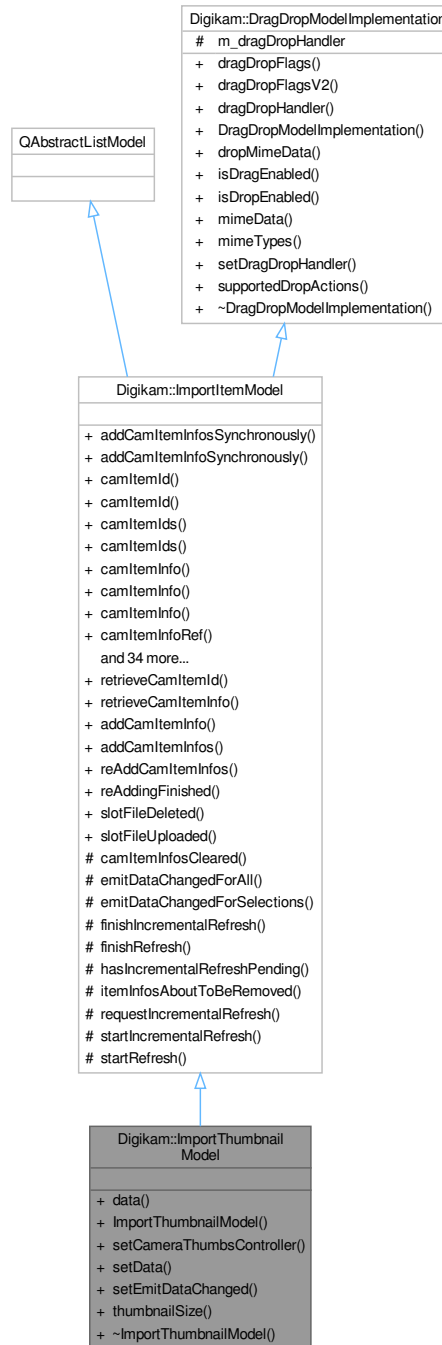
< switch off drawing of frames

< switch off composing rating over background



## 6.856 Digikam::ImportThumbnailModel Class Reference

Inheritance diagram for Digikam::ImportThumbnailModel:



### Signals

- void **thumbnailAvailable** (const QModelIndex &index, int requestedSize)
- void **thumbnailFailed** (const QModelIndex &index, int requestedSize)

## Signals inherited from [Digikam::ImportItemModel](#)

- void [allRefreshingFinished](#) ()
- void [itemInfosAboutToBeAdded](#) (const QList< [CamItemInfo](#) > &infos)
- void [itemInfosAboutToBeRemoved](#) (const QList< [CamItemInfo](#) > &infos)
- void [itemInfosAdded](#) (const QList< [CamItemInfo](#) > &infos)
- void [itemInfosRemoved](#) (const QList< [CamItemInfo](#) > &infos)
- void [preprocess](#) (const QList< [CamItemInfo](#) > &infos)
- void [processAdded](#) (const QList< [CamItemInfo](#) > &infos)
- void [readyForIncrementalRefresh](#) ()

## Public Member Functions

- QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const override
- [ImportThumbnailModel](#) (QObject \*const parent)
- void [setCameraThumbsController](#) ([CameraThumbsCtrl](#) \*const thumbsCtrl) override  
*Sets the camera thumbs controller which is used to get the thumbnails for item infos.*
- bool [setData](#) (const QModelIndex &index, const QVariant &value, int role=Qt::DisplayRole) override
- void [setEmitDataChanged](#) (bool emitSignal)
- [ThumbnailSize](#) [thumbnailSize](#) () const  
*Get the thumbnail size.*

## Public Member Functions inherited from [Digikam::ImportItemModel](#)

- void [addCamItemInfosSynchronously](#) (const Digikam::CamItemInfoList &infos)
- void [addCamItemInfoSynchronously](#) (const [CamItemInfo](#) &info)
- qlonglong [camItemId](#) (const QModelIndex &index) const
- qlonglong [camItemId](#) (int row) const
- QList< qlonglong > [camItemIds](#) () const
- QList< qlonglong > [camItemIds](#) (const QList< QModelIndex > &indexes) const
- [CamItemInfo](#) [camItemInfo](#) (const QModelIndex &index) const
- [CamItemInfo](#) [camItemInfo](#) (const QUrl &fileUrl) const
- [CamItemInfo](#) [camItemInfo](#) (int row) const
- [CamItemInfo](#) & [camItemInfoRef](#) (const QModelIndex &index) const
- [CamItemInfo](#) & [camItemInfoRef](#) (int row) const
- QList< [CamItemInfo](#) > [camItemInfos](#) () const
- [CamItemInfoList](#) [camItemInfos](#) (const QList< QModelIndex > &indexes) const
- QList< [CamItemInfo](#) > [camItemInfos](#) (const QUrl &fileUrl) const
- void [clearCamItemInfos](#) ()
- QVariant [data](#) (const QModelIndex &index, int role) const override
- Qt::ItemFlags [flags](#) (const QModelIndex &index) const override
- bool [hasImage](#) (const [CamItemInfo](#) &info) const
- bool [hasImage](#) (qlonglong id) const
- QVariant [headerData](#) (int section, Qt::Orientation orientation, int role) const override
- [ImportItemModel](#) (QObject \*const parent=nullptr)
- QModelIndex [index](#) (int row, int column, const QModelIndex &parent) const override
- QList< QModelIndex > [indexesForCamItemId](#) (qlonglong id) const
- QList< QModelIndex > [indexesForCamItemInfo](#) (const [CamItemInfo](#) &info) const
- QList< QModelIndex > [indexesForUrl](#) (const QUrl &fileUrl) const
- QModelIndex [indexForCamItemId](#) (qlonglong id) const
- QModelIndex [indexForCamItemInfo](#) (const [CamItemInfo](#) &info) const
- QModelIndex [indexForUrl](#) (const QUrl &fileUrl) const
- bool [isEmpty](#) () const

- bool `isRefreshing` () const
- bool `keepsFileUriCache` () const
- int `numberOfIndexesForCamItemId` (qulonglong id) const
- int `numberOfIndexesForCamItemInfo` (const [CamItemInfo](#) &info) const
- void `removeCamItemInfo` (const [CamItemInfo](#) &info)
- void `removeCamItemInfos` (const QList< [CamItemInfo](#) > &infos)
- void `removeIndex` (const QModelIndex &index)
- void `removeIndexes` (const QList< QModelIndex > &indexes)
- int `rowCount` (const QModelIndex &parent) const override  
*QAbstractListModel implementation.*
- void `setCamItemInfos` (const CamItemInfoList &infos)
- void `setKeepsFileUriCache` (bool keepCache)
- DECLARE\_MODEL\_DRAG\_DROP\_METHODS void `setSendRemovalSignals` (bool send)  
*DragDrop methods.*
- QList< [CamItemInfo](#) > `uniqueCamItemInfos` () const

## Public Member Functions inherited from [Digikam::DragDropModelImplementation](#)

- virtual Qt::ItemFlags `dragDropFlags` (const QModelIndex &index) const
- Qt::ItemFlags `dragDropFlagsV2` (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* `dragDropHandler` () const
- [DragDropModelImplementation](#) ()=default
- bool `dropMimeData` (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool `isDragEnabled` (const QModelIndex &index) const
- virtual bool `isDropEnabled` (const QModelIndex &index) const
- QMimeData \* `mimeData` (const QModelIndexList &indexes) const
- QStringList `mimeTypes` () const
- void `setDragDropHandler` ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions `supportedDropActions` () const

## Additional Inherited Members

## Public Types inherited from [Digikam::ImportItemModel](#)

- enum [ImportItemModelRoles](#) {  
[ImportItemModelPointerRole](#) = Qt::UserRole , [ImportItemModelInternalId](#) = Qt::UserRole + 1 ,  
[ThumbnailRole](#) = Qt::UserRole + 2 , [ExtraDataRole](#) = Qt::UserRole + 3 ,  
[ExtraDataDuplicateCount](#) = Qt::UserRole + 6 , [FilterModelRoles](#) = Qt::UserRole + 100 }

## Public Slots inherited from [Digikam::ImportItemModel](#)

- void `addCamItemInfo` (const [CamItemInfo](#) &info)
- void `addCamItemInfos` (const CamItemInfoList &infos)
- void `reAddCamItemInfos` (const CamItemInfoList &infos)
- void `reAddingFinished` ()
- void `slotFileDeleted` (const QString &folder, const QString &file, bool status)
- void `slotFileUploaded` (const [CamItemInfo](#) &info)

## Static Public Member Functions inherited from [Digikam::ImportItemModel](#)

- static qlonglong **retrieveCamItemid** (const QModelIndex &index)
- static [CamItemInfo](#) **retrieveCamItemInfo** (const QModelIndex &index)

## Protected Member Functions inherited from [Digikam::ImportItemModel](#)

- virtual void **camItemInfosCleared** ()
- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelections** (const QItemSelection &selection)
- void **finishIncrementalRefresh** ()
- void **finishRefresh** ()
- bool **hasIncrementalRefreshPending** () const
- virtual void **itemInfosAboutToBeRemoved** (int, int)
- void **requestIncrementalRefresh** ()
- void **startIncrementalRefresh** ()
- void **startRefresh** ()

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr

## 6.856.1 Constructor & Destructor Documentation

### 6.856.1.1 ImportThumbnailModel()

```
Digikam::ImportThumbnailModel::ImportThumbnailModel (
    QObject *const parent ) [explicit]
```

This model provides thumbnail loading, it uses the Camera Controller to retrieve thumbnails for CamItemInfos. It also provides preloading of thumbnails, and caching facility. Thumbnails size can be adjusted.

## 6.856.2 Member Function Documentation

### 6.856.2.1 data()

```
QVariant Digikam::ImportThumbnailModel::data (
    const QModelIndex & index,
    int role = Qt::DisplayRole ) const [override]
```

Handles the ThumbnailRole. If the pixmap is available, returns it in the QVariant. If it still needs to be loaded, returns a null QVariant and emits thumbnailAvailable() as soon as it is available.

### 6.856.2.2 setCameraThumbsController()

```
void Digikam::ImportThumbnailModel::setCameraThumbsController (
    CameraThumbsCtrl *const thumbsCtrl ) [override], [virtual]
```

Reimplemented from [Digikam::ImportItemModel](#).

### 6.856.2.3 setData()

```
bool Digikam::ImportThumbnailModel::setData (
    const QModelIndex & index,
    const QVariant & value,
    int role = Qt::DisplayRole ) [override]
```

You can override the current thumbnail size by giving an integer value for ThumbnailRole. Set a null QVariant to use the thumbnail size set by setThumbnailSize() again. The index given here is ignored for this purpose.

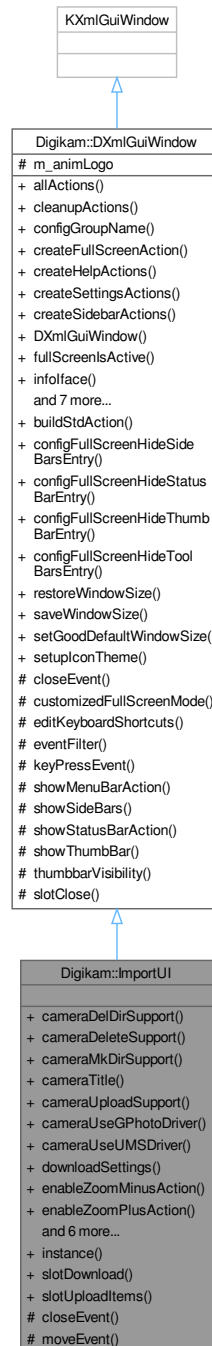
### 6.856.2.4 setEmitDataChanged()

```
void Digikam::ImportThumbnailModel::setEmitDataChanged (
    bool emitSignal )
```

Enable emitting dataChanged() when a thumbnail becomes available. The thumbnailAvailable() signal will be emitted in any case. Default is true.

## 6.857 Digikam::ImportUI Class Reference

Inheritance diagram for Digikam::ImportUI:



### Classes

- class [Private](#)

## Public Slots

- void **slotDownload** (bool onlySelected, bool deleteAfter, [Album](#) \*pAlbum=nullptr)
- void **slotUploadItems** (const QList< QUrl > &)

## Signals

- void **signalEscapePressed** ()
- void **signalLastDestination** (const QUrl &)
- void **signalNewSelection** (bool)
- void **signalPreviewRequested** (const [CamItemInfo](#) &, bool)
- void **signalWindowHasMoved** ()

## Public Member Functions

- bool **cameraDelDirSupport** () const
- bool **cameraDeleteSupport** () const
- bool **cameraMkDirSupport** () const
- QString **cameraTitle** () const
- bool **cameraUploadSupport** () const
- bool **cameraUseGPhotoDriver** () const
- bool **cameraUseUMSDriver** () const
- [DownloadSettings](#) **downloadSettings** () const
- void **enableZoomMinusAction** (bool val)
- void **enableZoomPlusAction** (bool val)
- [CameraThumbsCtrl](#) \* **getCameraThumbsCtrl** () const
- **ImportUI** (const QString &cameraTitle, const QString &model, const QString &port, const QString &path, int startIndex)
- [DInfoInterface](#) \* **infolface** ([DPluginAction](#) \*const) override
- bool **isBusy** () const
- bool **isClosed** () const

## Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- QList< QAction \* > **allActions** () const
- void **cleanupActions** ()
- QString **configGroupName** () const
- void **createFullScreenAction** (const QString &name)
- void **createHelpActions** (const QString &handbookSection, bool coreOptions=true)
- void **createSettingsActions** ()
- void **createSidebarActions** ()
- **DXmlGuiWindow** (QWidget \*const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())
- bool **fullScreensActive** () const
- void **readFullScreenSettings** (const KConfigGroup &group)
- virtual void **registerExtraPluginsActions** (QString &)
- void **registerPluginsActions** ()
- void **setConfigGroupName** (const QString &name)
- void **setFullScreenOptions** (int options)
- void **unminimizeAndActivateWindow** ()

**Static Public Member Functions**

- static [ImportUI](#) \* **instance** ()

**Static Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)**

- static QAction \* **buildStdAction** (StdActionType type, const QObject \*const recvr, const char \*const slot, QObject \*const parent)
- static QString **configFullScreenHideSideBarsEntry** ()
- static QString **configFullScreenHideStatusBarEntry** ()
- static QString **configFullScreenHideThumbBarEntry** ()
- static QString **configFullScreenHideToolBarsEntry** ()
- static void **restoreWindowSize** (QWindow \*const win, const KConfigGroup &group)
- static void **saveWindowSize** (QWindow \*const win, KConfigGroup &group)
- static void **setGoodDefaultWindowSize** (QWindow \*const win)
- static void **setupIconTheme** ()

**Protected Member Functions**

- void **closeEvent** (QCloseEvent \*e) override
- void **moveEvent** (QMoveEvent \*e) override

**Protected Member Functions inherited from [Digikam::DXmlGuiWindow](#)**

- void **closeEvent** (QCloseEvent \*e) override
- void **editKeyboardShortcuts** (KActionCollection \*const extraac=nullptr, const QString &actitle=QString())
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override
- void **keyPressEvent** (QKeyEvent \*e) override
- QAction \* **showMenuBarAction** () const
- QAction \* **showStatusBarAction** () const

**Additional Inherited Members****Protected Slots inherited from [Digikam::DXmlGuiWindow](#)**

- bool **slotClose** ()

**Protected Attributes inherited from [Digikam::DXmlGuiWindow](#)**

- [DLogoAction](#) \* **m\_animLogo** = nullptr

**6.857.1 Member Function Documentation****6.857.1.1 infoface()**

```
DInfoInterface * Digikam::ImportUI::infoIface (
    DPluginAction * const ac ) [inline], [override], [virtual]
```

Return the interface instance to access to items information.

Implements [Digikam::DXmlGuiWindow](#).



## 6.858 Digikam::ImportUI::Private Class Reference

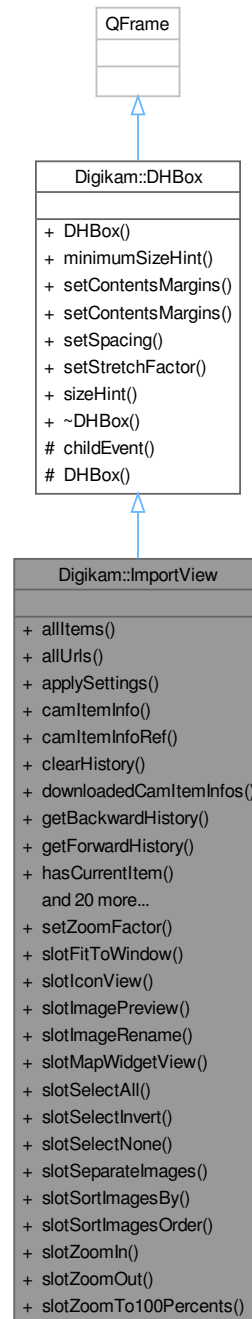
### Public Attributes

- [AdvancedSettings](#) \* **advancedSettings** = nullptr
- [DExpanderBox](#) \* **advBox** = nullptr
- [AlbumCustomizer](#) \* **albumCustomizer** = nullptr
- [FreeSpaceWidget](#) \* **albumLibraryFreeSpace** = nullptr
- bool **busy** = false
- QActionGroup \* **cameraActions** = nullptr
- QAction \* **cameraCancelAction** = nullptr
- QAction \* **cameraCaptureAction** = nullptr
- [FreeSpaceWidget](#) \* **cameraFreeSpace** = nullptr
- QAction \* **cameraInfoAction** = nullptr
- QString **cameraTitle**
- QAction \* **camItemPreviewAction** = nullptr
- [CameraThumbsCtrl](#) \* **camThumbsCtrl** = nullptr
- bool **closed** = false
- const QString **configDefaultTargetAlbumId** = QLatin1String("DefaultTargetAlbumId")
- const QString **configFileSaveConflictRule** = QLatin1String("FileSaveConflictRule")
- const QString **configGroupName** = QLatin1String("Camera Settings")
- const QString **configLastTargetAlbum** = QLatin1String("LastTargetAlbum")
- const QString **configUseDefaultTargetAlbum** = QLatin1String("UseDefaultTargetAlbum")
- const QString **configUseFileMetadata** = QLatin1String("UseFileMetadata")
- QAction \* **connectAction** = nullptr
- [CameraController](#) \* **controller** = nullptr
- QStringList **currentlyDeleting**
- QAction \* **decreaseThumbsAction** = nullptr
- QMenu \* **deleteAction** = nullptr
- bool **deleteAfter** = false
- QAction \* **deleteAllAction** = nullptr
- QMenu \* **deleteMenu** = nullptr
- QAction \* **deleteNewAction** = nullptr
- QAction \* **deleteSelectedAction** = nullptr
- QMenu \* **downloadAction** = nullptr
- QAction \* **downloadAllAction** = nullptr
- QAction \* **downloadDelAllAction** = nullptr
- QAction \* **downloadDelNewAction** = nullptr
- QAction \* **downloadDelSelectedAction** = nullptr
- QHash< QString, QDateTime > **downloadedDateHash**
- QHash< QString, QPair< QString, QString > > **downloadedInfoHash**
- QStringList **downloadedItemList**
- QMenu \* **downloadMenu** = nullptr
- QAction \* **downloadNewAction** = nullptr
- QAction \* **downloadSelectedAction** = nullptr
- [DNotificationWidget](#) \* **errorWidget** = nullptr
- [ImportFilterComboBox](#) \* **filterComboBox** = nullptr
- [FilterStatusBar](#) \* **filterStatusBar** = nullptr
- QStringList **foldersToScan**
- [DHistoryView](#) \* **historyView** = nullptr
- QAction \* **iconViewAction** = nullptr
- QMenu \* **imageMenu** = nullptr
- KSelectAction \* **imageViewSelectionAction** = nullptr
- const QString **importFiltersConfigGroupName** = QLatin1String("Import Filters")

- QAction \* **increaseThumbsAction** = nullptr
- KSelectAction \* **itemsGroupAction** = nullptr
- KSelectAction \* **itemSortAction** = nullptr
- KSelectAction \* **itemSortOrderAction** = nullptr
- QUrl **lastDestURL**
- QAction \* **lockAction** = nullptr
- QAction \* **markAsDownloadedAction** = nullptr
- QAction \* **pauseAction** = nullptr
- QTimer \* **progressTimer** = nullptr
- float **progressValue** = 0.0F
- [RenameCustomizer](#) \* **renameCustomizer** = nullptr
- QAction \* **resumeAction** = nullptr
- [ImportItemPropertiesSideBarImport](#) \* **rightSideBar** = nullptr
- [ScriptingSettings](#) \* **scriptingSettings** = nullptr
- QAction \* **selectAllAction** = nullptr
- QAction \* **selectInvertAction** = nullptr
- QAction \* **selectLockedItemsAction** = nullptr
- QAction \* **selectNewItemAction** = nullptr
- QAction \* **selectNoneAction** = nullptr
- QAction \* **showBarAction** = nullptr
- QAction \* **showLogAction** = nullptr
- QAction \* **showPreferencesAction** = nullptr
- [SidebarSplitter](#) \* **splitter** = nullptr
- QString **statusBarText**
- [StatusProgressBar](#) \* **statusProgressBar** = nullptr
- QAction \* **uploadAction** = nullptr
- [ImportView](#) \* **view** = nullptr
- QAction \* **viewCMViewAction** = nullptr
- bool **waitAutoRotate** = false
- [DZoomBar](#) \* **zoomBar** = nullptr
- QAction \* **zoomFitToWindowAction** = nullptr
- QAction \* **zoomTo100percents** = nullptr

## 6.859 Digikam::ImportView Class Reference

Inheritance diagram for Digikam::ImportView:



### Public Slots

- void **setZoomFactor** (double zoom)
- void **slotFitToWindow** ()

- void **slotIconView** ()
- void **slotImagePreview** ()
- void **slotImageRename** ()
- void **slotMapView** ()
- void **slotSelectAll** ()
- void **slotSelectInvert** ()
- void **slotSelectNone** ()
- void **slotSeparateImages** (int mode)
- void **slotSortImagesBy** (int sortBy)
- void **slotSortImagesOrder** (int order)
- void **slotZoomIn** ()
- *View Action slots.*
- void **slotZoomOut** ()
- void **slotZoomTo100Percents** ()

### Signals

- void **signalImageSelected** (const CamItemInfoList &selectedImage, const CamItemInfoList &allImages)
- void **signalNewSelection** (bool hasSelection)
- void **signalNoCurrentItem** ()
- void **signalSelectionChanged** (int numberOfSelectedItems)
- void **signalSwitchedToIconView** ()
- void **signalSwitchedToMapView** ()
- void **signalSwitchedToPreview** ()
- void **signalThumbSizeChanged** (int)
- void **signalZoomChanged** (double)

### Public Member Functions

- QList< [CamItemInfo](#) > **allItems** () const
- QList< QUrl > **allUrls** () const
- void **applySettings** ()
- [CamItemInfo](#) **camItemInfo** (const QString &folder, const QString &file) const
- [CamItemInfo](#) & **camItemInfoRef** (const QString &folder, const QString &file) const
- void **clearHistory** ()
- int **downloadedCamItemInfos** () const
- void **getBackwardHistory** (QStringList &titles)
- void **getForwardHistory** (QStringList &titles)
- bool **hasCurrentItem** () const
- bool **hasImage** (const [CamItemInfo](#) &info) const
- void **hideSideBars** ()
- [ImportFilterModel](#) \* **importFilterModel** () const
- **ImportView** ([Digikam::ImportUI](#) \*const ui, QWidget \*const parent)
- bool **isSelected** (const QUrl &url) const
- void **refreshView** ()
- void **scrollTo** (const QString &folder, const QString &file)
- QList< [CamItemInfo](#) > **selectedCamItemInfos** () const
- QList< QUrl > **selectedUrls** () const
- void **setSelectedCamItemInfos** (const CamItemInfoList &infos) const
- void **setThumbSize** (int size)
- void **showSideBars** ()
- [ThumbnailSize](#) **thumbnailSize** () const
- void **toggleFullScreen** (bool set)
- void **toggleShowBar** (bool b)
- void **updateIconView** ()
- [ImportStackedView::StackedViewMode](#) **viewMode** () const
- double **zoomMax** () const
- double **zoomMin** () const

## Public Member Functions inherited from Digikam::DHBox

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

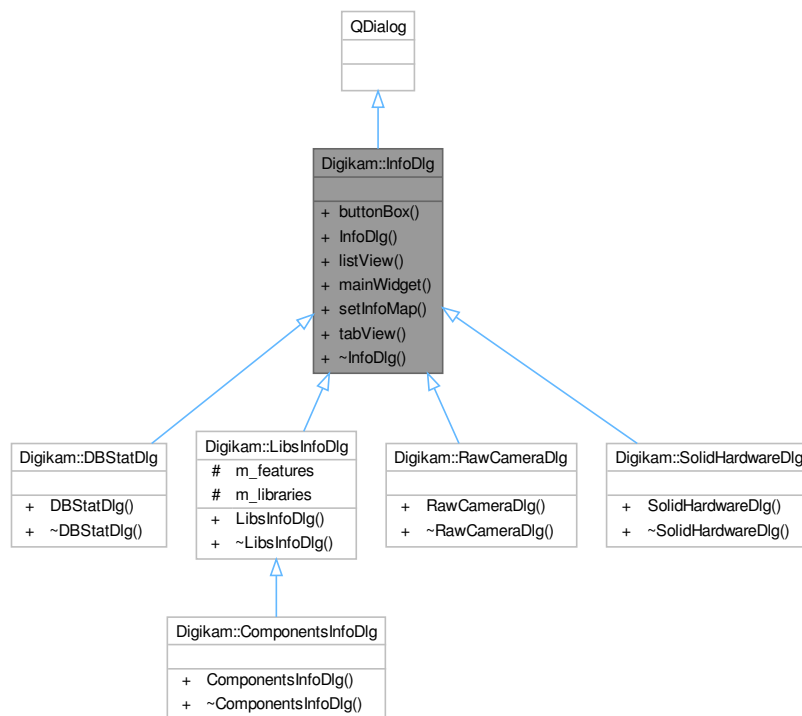
## Additional Inherited Members

## Protected Member Functions inherited from Digikam::DHBox

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.860 Digikam::InfoDlg Class Reference

Inheritance diagram for Digikam::InfoDlg:



### Public Member Functions

- QDialogButtonBox \* **buttonBox** () const
- **InfoDlg** (QWidget \*const parent)
- QTreeWidget \* **listView** () const
- QWidget \* **mainWidget** () const
- virtual void **setInfoMap** (const QMap< QString, QString > &list)
- QTabWidget \* **tabView** () const

## 6.861 Digikam::InfraredContainer Class Reference

### Public Attributes

- double **blueGain** = -0.8
- double **greenGain** = 2.1
- double **redGain** = 0.4
- int **sensibility** = 200

*Sensibility: 200..2600 ISO.*

## 6.862 Digikam::InfraredFilter Class Reference

Inheritance diagram for Digikam::InfraredFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **InfraredFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [InfraredContainer](#) &settings=[InfraredContainer](#)())
- **InfraredFilter** (QObject \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }



## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.862.1 Member Function Documentation

### 6.862.1.1 filterAction()

```
FilterAction Digikam::InfraredFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.862.1.2 filterIdentifier()

```
QString Digikam::InfraredFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

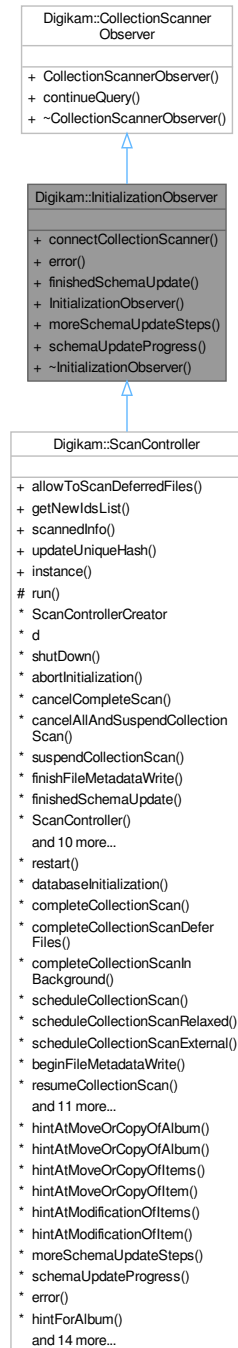
### 6.862.1.3 readParameters()

```
void Digikam::InfraredFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.863 Digikam::InitializationObserver Class Reference

Inheritance diagram for Digikam::InitializationObserver:



### Public Types

- enum **UpdateResult** { **UpdateSuccess** , **UpdateError** , **UpdateErrorMustAbort** }

### Public Member Functions

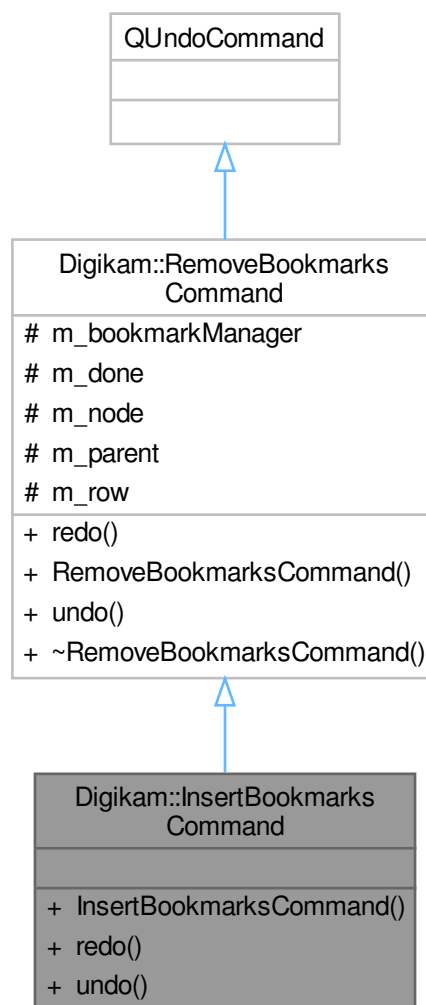
- virtual void **connectCollectionScanner** ([CollectionScanner](#) \*const scanner)=0
- virtual void **error** (const QString &errorMessage)=0
- virtual void **finishedSchemaUpdate** (UpdateResult result)=0
- virtual void **moreSchemaUpdateSteps** (int numberOfSteps)=0
- virtual void **schemaUpdateProgress** (const QString &message, int numberOfSteps=1)=0

### Public Member Functions inherited from [Digikam::CollectionScannerObserver](#)

- virtual bool **continueQuery** ()=0

## 6.864 Digikam::InsertBookmarksCommand Class Reference

Inheritance diagram for Digikam::InsertBookmarksCommand:



### Public Member Functions

- **InsertBookmarksCommand** ([BookmarksManager](#) \*const mngr, [BookmarkNode](#) \*const parent, [BookmarkNode](#) \*const node, int row)
- void **redo** () override
- void **undo** () override

### Public Member Functions inherited from [Digikam::RemoveBookmarksCommand](#)

- void **redo** () override
- **RemoveBookmarksCommand** ([BookmarksManager](#) \*const mngr, [BookmarkNode](#) \*const parent, int row)
- void **undo** () override

### Additional Inherited Members

### Protected Attributes inherited from [Digikam::RemoveBookmarksCommand](#)

- [BookmarksManager](#) \* **m\_bookmarkManager** = nullptr
- bool **m\_done** = false
- [BookmarkNode](#) \* **m\_node** = nullptr
- [BookmarkNode](#) \* **m\_parent** = nullptr
- int **m\_row** = 0

## 6.865 Digikam::InternalTagName Class Reference

### Static Public Member Functions

- static QLatin1String **colorLabelBlack** ()
- static QLatin1String **colorLabelBlue** ()
- static QLatin1String **colorLabelGray** ()
- static QLatin1String **colorLabelGreen** ()
- static QLatin1String **colorLabelMagenta** ()
- static QLatin1String **colorLabelNone** ()
- static QLatin1String **colorLabelOrange** ()
- static QLatin1String **colorLabelRed** ()
- static QLatin1String **colorLabelWhite** ()
- static QLatin1String **colorLabelYellow** ()
- static QLatin1String **currentVersion** ()
- static QLatin1String **intermediateVersion** ()
- static QLatin1String **needResolvingHistory** ()
- static QLatin1String **needTaggingHistoryGraph** ()
- static QLatin1String **originalVersion** ()
- static QLatin1String **pickLabelAccepted** ()
- static QLatin1String **pickLabelNone** ()
- static QLatin1String **pickLabelPending** ()
- static QLatin1String **pickLabelRejected** ()
- static QLatin1String **scannedForFaces** ()
- static QLatin1String **versionAlwaysVisible** ()

## 6.866 Digikam::InvertFilter Class Reference

Inheritance diagram for Digikam::InvertFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **InvertFilter** ([DImg](#) \*const orgImage, [QObject](#) \*const parent=nullptr)
- **InvertFilter** ([DImgThreadedFilter](#) \*const parentFilter, const [DImg](#) &orgImage, [DImg](#) &destImage, int progressBegin=0, int progressEnd=100)
- **InvertFilter** ([QObject](#) \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false



## 6.866.1 Member Function Documentation

### 6.866.1.1 filterAction()

```
FilterAction Digikam::InvertFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.866.1.2 filterIdentifier()

```
QString Digikam::InvertFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.866.1.3 readParameters()

```
void Digikam::InvertFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.867 Digikam::IOFileSettings Class Reference

### Public Attributes

- int **AVIFCompression** = 75  
*AVIF quality value.*
- bool **AVIFLossLess** = true  
*AVIF lossless compression.*
- int **HEIFCompression** = 75  
*HEIF quality value.*
- bool **HEIFLossLess** = true  
*HEIF lossless compression.*
- int **JPEG2000Compression** = 75  
*JPEG2000 quality value.*
- bool **JPEG2000LossLess** = true  
*JPEG2000 lossless compression.*
- int **JPEGCompression** = 75  
*JPEG quality value.*
- int **JPEGSubSampling** = 1  
*JPEG chroma sub-sampling value.*
- int **JXLCompression** = 75  
*JXL quality value.*
- bool **JXLLossLess** = true

- *JXL lossless compression.*
- int **PGFCompression** = 3  
*PGF quality value.*
- bool **PGFLossLess** = true  
*PGF lossless compression.*
- int **PNGCompression** = 9  
*PNG compression value.*
- [DRawDecoding](#) **rawDecodingSettings**  
*RAW File decoding options.*
- QString **rawImportToolId** = QLatin1String("org.kde.digikam.plugin.rawimport.Native")
- bool **TIFFCompression** = false  
*TIFF deflate compression.*
- bool **useRAWImport** = true  
*Use Raw Import tool to load a RAW picture.*
- int **WEBPCompression** = 75  
*WEBP quality value.*
- bool **WEBPLossLess** = true  
*WEBP lossless compression.*

## 6.867.1 Member Data Documentation

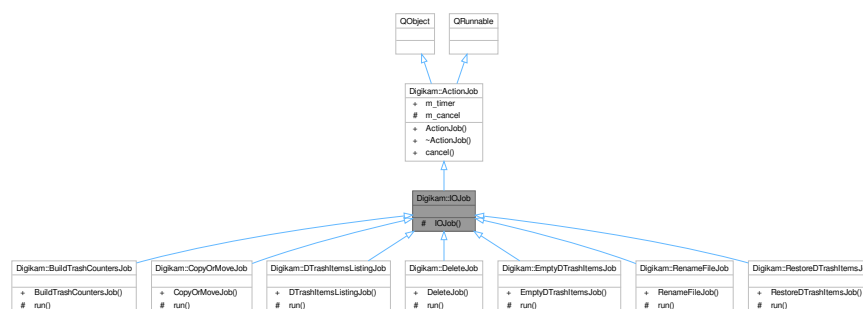
### 6.867.1.1 JPEGSubSampling

```
int Digikam::IOFileSettings::JPEGSubSampling = 1
```

Medium sub-sampling

## 6.868 Digikam::IOJob Class Reference

Inheritance diagram for Digikam::IOJob:



## Signals

- void **signalError** (const QString &errMsg)
- void **signalOneProcessed** (const QUrl &url)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.869 Digikam::IOJobData Class Reference

### Public Types

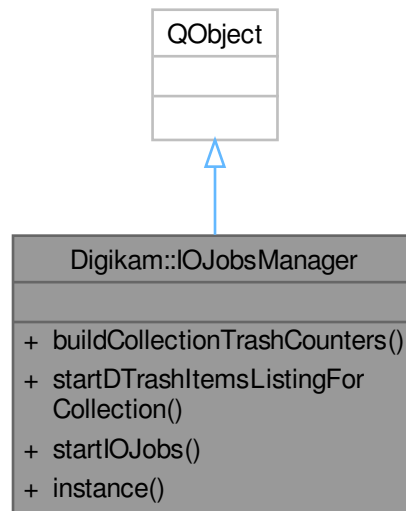
- enum [FileConflict](#) { [Continue](#) = 0 , [AutoRename](#) , [Overwrite](#) }
- enum [Operation](#) {  
    [Unknown](#) = 0 , [CopyAlbum](#) , [CopyImage](#) , [CopyFiles](#) ,  
    [CopyToExt](#) , [MoveAlbum](#) , [MoveImage](#) , [MoveFiles](#) ,  
    [Restore](#) , [Rename](#) , [Delete](#) , [Trash](#) ,  
    [Empty](#) }

## Public Member Functions

- [PAlbum](#) \* **destAlbum** () const
- QString **destName** (const QUrl &srcUrl) const
- QUrl **destUrl** (const QUrl &srcUrl=QUrl()) const
- bool **errorOrCancel** () const
- int **fileConflict** () const
- [ItemInfo](#) **findItemInfo** (const QUrl &url) const
- QUrl **getNextUrl** () const
- QString **getProgressId** () const
- **IOJobData** (int operation, const DTrashItemInfoList &infos)
- **IOJobData** (int operation, const [ItemInfo](#) &info, const QString &newName, bool overwrite=false)
- **IOJobData** (int operation, const QList< [ItemInfo](#) > &infos, const QUrl &dest)
- **IOJobData** (int operation, const QList< [ItemInfo](#) > &infos, [PAlbum](#) \*const dest=nullptr)
- **IOJobData** (int operation, const QList< QUrl > &urls, const QUrl &dest)
- **IOJobData** (int operation, const QList< QUrl > &urls, [PAlbum](#) \*const dest=nullptr)
- **IOJobData** (int operation, [PAlbum](#) \*const src, [PAlbum](#) \*const dest=nullptr)
- QList< [ItemInfo](#) > **itemInfos** () const
- QDateTime **jobTime** () const
- int **operation** () const
- void **setDestUrl** (const QUrl &srcUrl, const QUrl &destUrl)
- void **setErrorOrCancel** (bool err)
- void **setFileConflict** (int fc)
- void **setItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **setProgressId** (const QString &id)
- void **setSourceUrls** (const QList< QUrl > &urls)
- QList< QUrl > **sourceUrls** () const
- [PAlbum](#) \* **srcAlbum** () const
- QList< int > **srcAlbumIds** () const
- DTrashItemInfoList **trashItems** () const

## 6.870 Digikam::IOJobsManager Class Reference

Inheritance diagram for Digikam::IOJobsManager:



### Public Member Functions

- `IOJobsThread * buildCollectionTrashCounters ()`  
*Starts a thread for count trash items for all collections.*
- `IOJobsThread * startDTrashItemsListingForCollection (const QString &collectionPath)`  
*Starts a thread for listing items inside trash for specific collection.*
- `IOJobsThread * startIOJobs (IOJobData *const data)`  
*startIOJobs: Starts a thread to copy, move, delete or rename items*

### Static Public Member Functions

- static `IOJobsManager * instance ()`  
*instance: returns the singleton of IO Jobs Manager*

### Friends

- class `IOJobsManagerCreator`

## 6.870.1 Member Function Documentation

### 6.870.1.1 buildCollectionTrashCounters()

`IOJobsThread * Digikam::IOJobsManager::buildCollectionTrashCounters ( )`

#### Returns

`IOJobsThread` pointer for signal/slot connection

### 6.870.1.2 instance()

```
IOJobsManager * Digikam::IOJobsManager::instance ( ) [static]
```

#### Returns

[IOJobsManager](#) global instance

### 6.870.1.3 startDTrashItemsListingForCollection()

```
IOJobsThread * Digikam::IOJobsManager::startDTrashItemsListingForCollection (
    const QString & collectionPath )
```

#### Parameters

<i>collectionPath</i>	the path for collection to list items for it's trash
-----------------------	--

#### Returns

[IOJobsThread](#) pointer for signal/slot connection

### 6.870.1.4 startIOJobs()

```
IOJobsThread * Digikam::IOJobsManager::startIOJobs (
    IOJobData *const data )
```

#### Parameters

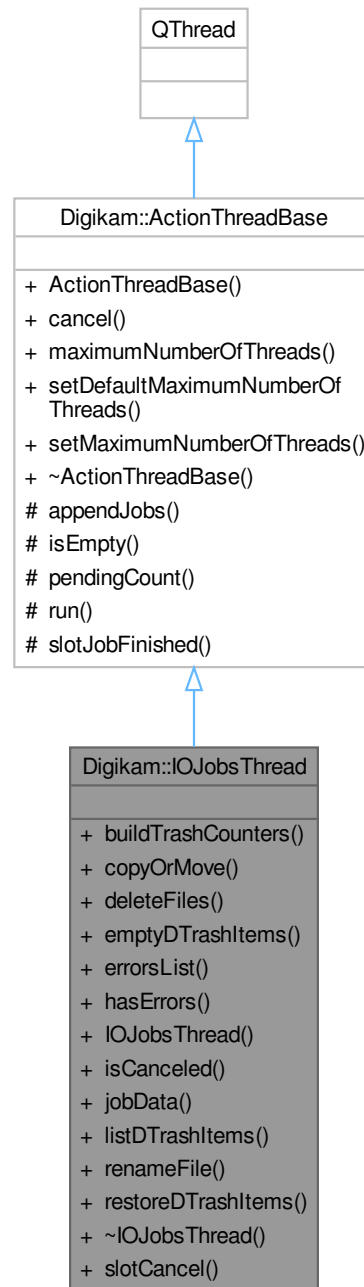
<i>data</i>	<a href="#">IOJobData</a> container with source and destination url
-------------	---

#### Returns

[IOJobsThread](#) pointer for signal/slot connection

## 6.871 Digikam::IOJobsThread Class Reference

Inheritance diagram for Digikam::IOJobsThread:



### Public Slots

- void **slotCancel** ()  
*cancels thread execution*

## Signals

- void **collectionTrashItemInfo** (const [DTrashItemInfo](#) &trashItemInfo)
- void **signalFinished** ()
- void **signalOneProcessed** (const [QUrl](#) &url)
- void **signalRenameFailed** (const [QUrl](#) &url)
- void **signalTrashCountersMap** (const [QMap](#)< [QString](#), int > &counterMap)

## Public Member Functions

- void **buildTrashCounters** ()  
*creates a job for count trash items from all collections*
- void **copyOrMove** ([IOJobData](#) \*const data)  
*Starts a number of jobs to copy or move source files to destination.*
- void **deleteFiles** ([IOJobData](#) \*const data)  
*Starts a number of jobs to delete multiple files.*
- void **emptyDTrashItems** ([IOJobData](#) \*const data)  
*creates a job for every item to delete from collection trash*
- [QStringList](#) & **errorsList** () const
- bool **hasErrors** () const  
*hasErrors*
- **IOJobsThread** ([QObject](#) \*const parent)
- bool **isCanceled** () const  
*isCanceled*
- [IOJobData](#) \* **jobData** () const
- void **listDTrashItems** (const [QString](#) &collectionPath)  
*Starts a job for listing trash items in a collection.*
- void **renameFile** ([IOJobData](#) \*const data)  
*Starts one job to rename a file to a new name.*
- void **restoreDTrashItems** ([IOJobData](#) \*const data)  
*creates a job for every item to restore back to album*

## Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- **ActionThreadBase** ([QObject](#) \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void **slotJobFinished** ()



## Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void [appendJobs](#) (const [ActionJobCollection](#) &jobs)
- bool [isEmpty](#) () const
- int [pendingCount](#) () const
- void [run](#) () override

### 6.871.1 Member Function Documentation

#### 6.871.1.1 [copyOrMove\(\)](#)

```
void Digikam::IOJobsThread::copyOrMove (
    IOJobData *const data )
```

##### Parameters

<i>data</i>	IOJobsData container
-------------	----------------------

#### 6.871.1.2 [deleteFiles\(\)](#)

```
void Digikam::IOJobsThread::deleteFiles (
    IOJobData *const data )
```

##### Parameters

<i>data</i>	IOJobsData container
-------------	----------------------

#### 6.871.1.3 [emptyDTrashItems\(\)](#)

```
void Digikam::IOJobsThread::emptyDTrashItems (
    IOJobData *const data )
```

##### Parameters

<i>data</i>	IOJobsData container
-------------	----------------------

#### 6.871.1.4 [errorsList\(\)](#)

```
QStringList & Digikam::IOJobsThread::errorsList ( ) const
```

##### Returns

the current errors list

### 6.871.1.5 hasErrors()

```
bool Digikam::IOJobsThread::hasErrors ( ) const
```

#### Returns

true if string list was not empty

### 6.871.1.6 isCanceled()

```
bool Digikam::IOJobsThread::isCanceled ( ) const
```

#### Returns

true if the thread was interrupted

### 6.871.1.7 jobData()

```
IOJobData * Digikam::IOJobsThread::jobData ( ) const
```

#### Returns

the current data job instance

### 6.871.1.8 listDTrashItems()

```
void Digikam::IOJobsThread::listDTrashItems (
    const QString & collectionPath )
```

#### Parameters

<i>collectionPath</i>	
-----------------------	--

### 6.871.1.9 renameFile()

```
void Digikam::IOJobsThread::renameFile (
    IOJobData *const data )
```

#### Parameters

<i>data</i>	IOJobsData container
-------------	----------------------

### 6.871.1.10 restoreDTrashItems()

```
void Digikam::IOJobsThread::restoreDTrashItems (
    IOJobData *const data )
```

#### Parameters

<i>data</i>	IOJobsData container
-------------	----------------------

## 6.872 Digikam::IptcCoreContactInfo Class Reference

### Public Member Functions

- bool **isEmpty** () const
- bool **isNull** () const
- void **merge** (const [IptcCoreContactInfo](#) &t)
- bool **operator==** (const [IptcCoreContactInfo](#) &t) const

### Public Attributes

- QString **address**
- QString **city**
- QString **country**
- QString **email**
- QString **phone**
- QString **postalCode**
- QString **provinceState**
- QString **webUrl**

## 6.873 Digikam::IptcCoreLocationInfo Class Reference

### Public Member Functions

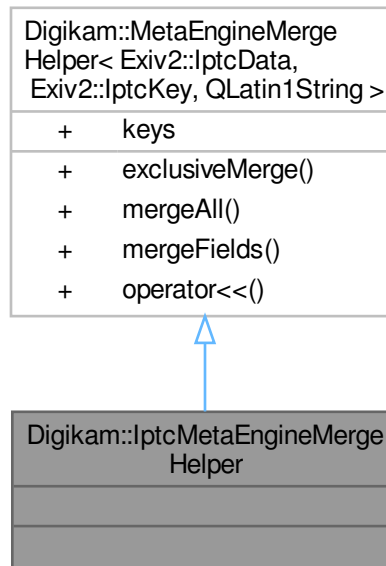
- bool **isEmpty** () const
- bool **isNull** () const
- void **merge** (const [IptcCoreLocationInfo](#) &t)
- bool **operator==** (const [IptcCoreLocationInfo](#) &t) const

### Public Attributes

- QString **city**
- QString **country**
- QString **countryCode**
- QString **location**
- QString **provinceState**

## 6.874 Digikam::IptcMetaEngineMergeHelper Class Reference

Inheritance diagram for Digikam::IptcMetaEngineMergeHelper:



### Additional Inherited Members

#### Public Member Functions inherited from

[Digikam::MetaEngineMergeHelper< Exiv2::IptcData, Exiv2::IptcKey, QLatin1String >](#)

- void [exclusiveMerge](#) (const Exiv2::IptcData &src, Exiv2::IptcData &dest)
- void [mergeAll](#) (const Exiv2::IptcData &src, Exiv2::IptcData &dest)
- void [mergeFields](#) (const Exiv2::IptcData &src, Exiv2::IptcData &dest)
- [MetaEngineMergeHelper](#) & [operator<<](#) (const QLatin1String &key)

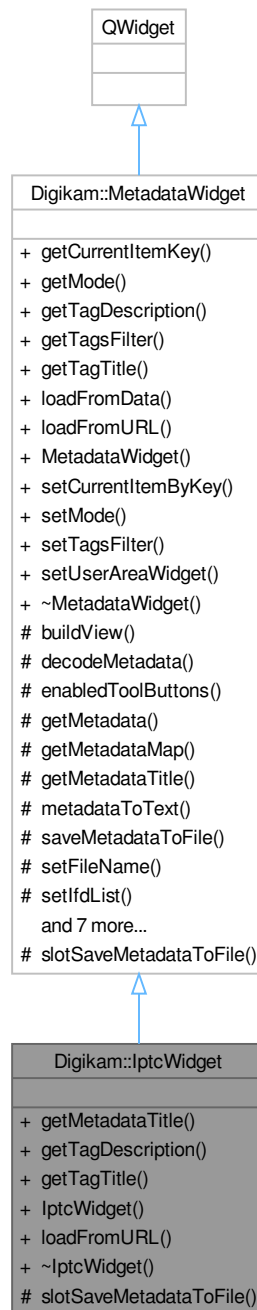
#### Public Attributes inherited from

[Digikam::MetaEngineMergeHelper< Exiv2::IptcData, Exiv2::IptcKey, QLatin1String >](#)

- QList< QLatin1String > **keys**

## 6.875 Digikam::IptcWidget Class Reference

Inheritance diagram for Digikam::IptcWidget:



### Public Member Functions

- `QString` `getMetadataTitle ()` const override
- `QString` `getTagDescription (const QString &key)` override

- QString [getTagTitle](#) (const QString &key) override
- **lptcWidget** (QWidget \*const parent, const QString &name=QString())
- bool [loadFromURL](#) (const QUrl &url) override

### Public Member Functions inherited from [Digikam::MetadataWidget](#)

- QString [getCurrentItemKey](#) () const
- int [getMode](#) () const
- QStringList [getTagsFilter](#) () const
- virtual bool [loadFromData](#) (const QString &fileName, const [DMetadata](#) &data=[DMetadata](#)())
- **MetadataWidget** (QWidget \*const parent, const QString &name=QString())
- void [setCurrentItemByKey](#) (const QString &itemKey)
- void [setMode](#) (int mode)
- void [setTagsFilter](#) (const QStringList &list)
- void [setUserAreaWidget](#) (QWidget \*const w)

### Protected Slots

- void [slotSaveMetadataToFile](#) () override

### Protected Slots inherited from [Digikam::MetadataWidget](#)

- virtual void [slotSaveMetadataToFile](#) ()=0

### Additional Inherited Members

### Public Types inherited from [Digikam::MetadataWidget](#)

- enum [TagFilters](#) { NONE = 0 , PHOTO , CUSTOM }

### Signals inherited from [Digikam::MetadataWidget](#)

- void [signalSetupMetadataFilters](#) ()

### Protected Member Functions inherited from [Digikam::MetadataWidget](#)

- void [enabledToolButtons](#) (bool)
- [DMetadata](#) \* [getMetadata](#) () const
- const [DMetadata::MetaDatum](#) & [getMetadataMap](#) ()
- QString [metadataToText](#) () const
- QUrl [saveMetadataToFile](#) (const QString &caption, const QString &fileFilter)
- void [setFileName](#) (const QString &fileName)
- void [setIfdList](#) (const [DMetadata::MetaDatum](#) &ifds, const QStringList &keysFilter, const QStringList &tagsFilter)
- void [setIfdList](#) (const [DMetadata::MetaDatum](#) &ifds, const QStringList &tagsFilter=QStringList())
- bool [setMetadata](#) (const [DMetadata](#) &data=[DMetadata](#)())
- virtual void [setMetadataEmpty](#) ()
- void [setMetadataMap](#) (const [DMetadata::MetaDatum](#) &data=[DMetadata::MetaDatum](#)())
- void [setup](#) ()
- bool [storeMetadataToFile](#) (const QUrl &url, const QByteArray &metaData)
- [MetadataListView](#) \* [view](#) () const

## 6.875.1 Member Function Documentation

### 6.875.1.1 getMetadataTitle()

```
QString Digikam::IptcWidget::getMetadataTitle ( ) const [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

### 6.875.1.2 getTagDescription()

```
QString Digikam::IptcWidget::getTagDescription (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

### 6.875.1.3 getTagTitle()

```
QString Digikam::IptcWidget::getTagTitle (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

### 6.875.1.4 loadFromURL()

```
bool Digikam::IptcWidget::loadFromURL (
    const QUrl & url ) [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

## 6.876 Digikam::ItemAlbumFilterModel Class Reference

Inheritance diagram for Digikam::ItemAlbumFilterModel:



### Public Member Functions

- **ItemAlbumFilterModel** (QObject \*const parent=nullptr)
- void **setItemFilterSettings** (const **ItemFilterSettings** &settings) override
- void **setSourceItemModel** (**ItemAlbumModel** \*model)
- **ItemAlbumModel** \* **sourceModel** () const



## Public Member Functions inherited from Digikam::ItemFilterModel

- void [addPrepareHook](#) ([ItemFilterModelPrepareHook](#) \*const hook)
- QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const override
- [GroupItemFilterSettings](#) **groupItemFilterSettings** () const
- [ItemFilterModel](#) \* [imageFilterModel](#) () const override
- [ItemFilterSettings](#) **imageFilterSettings** () const
- [ItemSortSettings](#) **imageSortSettings** () const
- void **infosToProcess** (const QList< [ItemInfo](#) > &infos)
- void **infosToProcess** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues, bool forRe←Add=true)
- void **init** ([ItemFilterModel](#) \*qq)
- bool **isAllGroupsOpen** () const
- bool **isGroupOpen** (qulonglong group) const
- [ItemFilterModel](#) (QObject \*const parent=nullptr)
- [ItemFilterModelPrivate](#) ()
- void **removePrepareHook** ([ItemFilterModelPrepareHook](#) \*const hook)
- void [setSendItemInfoSignals](#) (bool sendSignals)
- void **setupWorkers** ()
- [DatabaseFields::Set suggestedWatchFlags](#) () const
- [VersionItemFilterSettings](#) **versionItemFilterSettings** () const

## Public Member Functions inherited from Digikam::ImageSortFilterModel

- qulonglong **imageId** (const QModelIndex &index) const
- QList< qulonglong > **imageIds** (const QList< QModelIndex > &indexes) const
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- QList< [ItemInfo](#) > **imageInfos** (const QList< QModelIndex > &indexes) const
- QList< [ItemInfo](#) > **imageInfosSorted** () const
- [ImageSortFilterModel](#) (QObject \*const parent=nullptr)
- QModelIndex **indexForImageId** (qulonglong id) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info) const
- QModelIndex **indexForPath** (const QString &filePath) const
- QModelIndex **mapFromDirectSourceToSourceItemModel** (const QModelIndex &sourceModel\_index) const
- QModelIndex **mapFromSourceItemModel** (const QModelIndex &imagemodel\_index) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceItemModel** (const QModelIndex &index) const
- void **setSourceFilterModel** ([ImageSortFilterModel](#) \*const model)
- void **setSourceItemModel** ([ItemModel](#) \*const model)
- [ImageSortFilterModel](#) \* **sourceFilterModel** () const
- [ItemModel](#) \* **sourceItemModel** () const

## Public Member Functions inherited from Digikam::DCategorizedSortFilterProxyModel

- [DCategorizedSortFilterProxyModel](#) (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void [setCategorizedModel](#) (bool categorizedModel)
- void [setSortCategoriesByNaturalComparison](#) (bool [sortCategoriesByNaturalComparison](#))
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool [sortCategoriesByNaturalComparison](#) () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

### Protected Slots

- void **slotAlbumAboutToBeDeleted** ([Album](#) \*album)
- void **slotAlbumAdded** ([Album](#) \*album)
- void **slotAlbumRenamed** ([Album](#) \*album)
- void **slotAlbumsCleared** ()

### Protected Slots inherited from [Digikam::ItemFilterModel](#)

- void **slotImageChange** (const [ImageChangeset](#) &changeset)
- void **slotImageTagChange** (const [ImageTagChangeset](#) &changeset)
- void **slotModelReset** ()
- void **slotRowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end)
- void **slotRowsInserted** (const QModelIndex &parent, int start, int end)
- void **slotUpdateFilter** ()

### Protected Member Functions

- int **compareInfosCategories** (const [ItemInfo](#) &left, const [ItemInfo](#) &right) const override
- int **compareInfosCategories** (const [ItemInfo](#) &left, const [ItemInfo](#) &right, const [FaceTagsIface](#) &leftFace, const [FaceTagsIface](#) &rightFace) const override

### Protected Member Functions inherited from [Digikam::ItemFilterModel](#)

- virtual QString **categoryIdentifier** (const [ItemInfo](#) &info, const [FaceTagsIface](#) &face) const
- int **compareCategories** (const QModelIndex &left, const QModelIndex &right) const override
- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override
- virtual bool **infosLessThan** (const [ItemInfo](#) &left, const [ItemInfo](#) &right) const
- **ItemFilterModel** ([ItemFilterModelPrivate](#) &dd, QObject \*const parent)
- void **setDirectSourceItemModel** ([ItemModel](#) \*const model) override
- bool **subSortLessThan** (const QModelIndex &left, const QModelIndex &right) const override

### Protected Member Functions inherited from [Digikam::ImageSortFilterModel](#)

- void **setSourceModel** ([QAbstractItemModel](#) \*const model) override
- NOTE: made protected.*

### Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override

### Additional Inherited Members

### Public Types inherited from [Digikam::ItemFilterModel](#)

- enum [ItemFilterModelRoles](#) {  
[CategorizationModeRole](#) = [ItemModel::FilterModelRoles](#) + 1 , [SortOrderRole](#) = [ItemModel::FilterModelRoles](#) + 2 , [CategoryAlbumIdRole](#) = [ItemModel::FilterModelRoles](#) + 3 , [CategoryFormatRole](#) = [ItemModel::FilterModelRoles](#) + 4 ,  
[CategoryDateRole](#) = [ItemModel::FilterModelRoles](#) + 5 , [CategoryFaceRole](#) = [ItemModel::FilterModelRoles](#) + 6 , [GroupsOpenRole](#) = [ItemModel::FilterModelRoles](#) + 7 , [ItemFilterModelPointerRole](#) = [ItemModel::FilterModelRoles](#) + 50 }

## Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## Public Slots inherited from [Digikam::ItemFilterModel](#)

- void **packageDiscarded** (const [ItemFilterModelTodoPackage](#) &package)
- void **packageFinished** (const [ItemFilterModelTodoPackage](#) &package)
- void **preprocessInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void **processAddedInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void **setAllGroupsOpen** (bool open)
- void **setCategorizationMode** ([ItemSortSettings::CategorizationMode](#) mode)
- void **setCategorizationSortOrder** ([ItemSortSettings::SortOrder](#) order)
- void **setDayFilter** (const QList< QDateTime > &days)
- void **setExceptionList** (const QList< qlonglong > &idlist, const QString &id)
- void **setGeolocationFilter** (const [ItemFilterSettings::GeolocationCondition](#) &condition)
- void **setGroupItemFilterSettings** (const [GroupItemFilterSettings](#) &settings)
- void **setGroupOpen** (qlonglong group, bool open)
- void **setIdWhitelist** (const QList< qlonglong > &idList, const QString &id)
- virtual void **setItemSortSettings** (const [ItemSortSettings](#) &settings)
- void **setMimeTypeFilter** (int mimeTypeFilter)
- void **setRatingFilter** (int rating, [ItemFilterSettings::RatingCondition](#) ratingCond, bool isUnratedExcluded)
- void **setSortOrder** ([ItemSortSettings::SortOrder](#) order)
- void **setSortRole** ([ItemSortSettings::SortRole](#) role)
- void **setStringTypeNatural** (bool natural)
- void **setTagFilter** (const QList< int > &includedTags, const QList< int > &excludedTags, [ItemFilterSettings::MatchingCondition](#) matchingCond, bool showUnTagged, const QList< int > &clTagIds, const QList< int > &plTagIds)
- void **setTextFilter** (const [SearchTextFilterSettings](#) &settings)
- void **setUrlWhitelist** (const QList< QUrl > &urlList, const QString &id)
- void **setVersionItemFilterSettings** (const [VersionItemFilterSettings](#) &settings)
- void **setVersionManagerSettings** (const [VersionManagerSettings](#) &settings)
- void **toggleGroupOpen** (qlonglong group)

## Signals inherited from [Digikam::ItemFilterModel](#)

- void **filterMatches** (bool matches)
- void **filterMatchesForText** (bool matchesByText)
- void **filterSettingsChanged** (const [ItemFilterSettings](#) &settings)
- void **imageInfosAboutToBeRemoved** (const QList< [ItemInfo](#) > &infos)
- void **imageInfosAdded** (const QList< [ItemInfo](#) > &infos)
- void **packageToFilter** (const [ItemFilterModelTodoPackage](#) &package)
- void **packageToPrepare** (const [ItemFilterModelTodoPackage](#) &package)
- void **reAddingFinished** ()
- void **reAddItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)

## Public Attributes inherited from [Digikam::ItemFilterModel](#)

- [ItemFilterSettings](#) **filter**
- [ItemFilterSettings](#) **filterCopy**
- [ItemFilterModelFilterer](#) \* **filterer** = nullptr
- [QHash](#)< [qlonglong](#), [bool](#) > **filterResults**
- [GroupItemFilterSettings](#) **groupFilter**
- [GroupItemFilterSettings](#) **groupFilterCopy**
- [bool](#) **hasOneMatch** = false
- [bool](#) **hasOneMatchForText** = false
- [ItemModel](#) \* **imageModel** = nullptr
- [unsigned int](#) **lastDiscardVersion** = 0
- [unsigned int](#) **lastFilteredVersion** = 0
- [QMutex](#) **mutex**
- [bool](#) **needPrepare** = false
- [bool](#) **needPrepareComments** = false
- [bool](#) **needPrepareGroups** = false
- [bool](#) **needPrepareTags** = false
- [QList](#)< [ItemFilterModelPrepareHook](#) \* > **prepareHooks**
- [ItemFilterModelPreparer](#) \* **preparer** = nullptr
- [ItemFilterModel](#) \* **q** = nullptr
- [int](#) **sentOut** = 0
- [int](#) **sentOutForReAdd** = 0
- [ItemSortSettings](#) **sorter**
- [QTimer](#) \* **updateFilterTimer** = nullptr
- [volatile unsigned int](#) **version** = 0
- [VersionItemFilterSettings](#) **versionFilter**
- [VersionItemFilterSettings](#) **versionFilterCopy**

## Protected Attributes inherited from [Digikam::ItemFilterModel](#)

- [ItemFilterModelPrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ImageSortFilterModel](#)

- [ImageSortFilterModel](#) \* **m\_chainedModel** = nullptr

## 6.876.1 Member Function Documentation

### 6.876.1.1 `compareInfosCategories()` [1/2]

```
int Digikam::ItemAlbumFilterModel::compareInfosCategories (
    const ItemInfo & left,
    const ItemInfo & right ) const [override], [protected], [virtual]
```

Reimplement to customize category sorting. Return negative if category of left < category right, Return 0 if left and right are in the same category, else return positive.

Reimplemented from [Digikam::ItemFilterModel](#).

### 6.876.1.2 compareInfosCategories() [2/2]

```
int Digikam::ItemAlbumFilterModel::compareInfosCategories (
    const ItemInfo & left,
    const ItemInfo & right,
    const FaceTagsIface & leftFace,
    const FaceTagsIface & rightFace ) const [override], [protected], [virtual]
```

In order to be able to Categorize by Faces, it's necessary to pass in the face as well. One image may have multiple Faces in it, hence just the [ItemInfo](#) isn't sufficient.

Reimplemented from [Digikam::ItemFilterModel](#).

### 6.876.1.3 setItemFilterSettings()

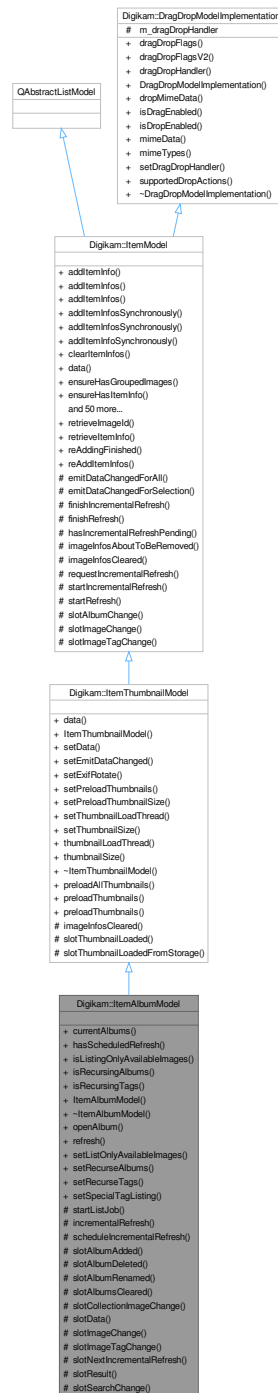
```
void Digikam::ItemAlbumFilterModel::setItemFilterSettings (
    const ItemFilterSettings & settings ) [override], [virtual]
```

Changes the current image filter settings and refilters.

Reimplemented from [Digikam::ItemFilterModel](#).

## 6.877 Digikam::ItemAlbumModel Class Reference

Inheritance diagram for Digikam::ItemAlbumModel:



### Public Slots

- void `openAlbum` (const QList< Album \* > &albums)
- void `refresh` ()

- void **setListOnlyAvailableImages** (bool onlyAvailable)
- void **setRecurseAlbums** (bool recursiveListing)
- void **setRecurseTags** (bool recursiveListing)
- void **setSpecialTagListing** (const QString &specialListing)

### Public Slots inherited from [Digikam::ItemThumbnailModel](#)

- void **preloadAllThumbnails** ()
- void **preloadThumbnails** (const QList< [ItemInfo](#) > &)
- void **preloadThumbnails** (const QList< [QModelIndex](#) > &)

### Public Slots inherited from [Digikam::ItemModel](#)

- void **reAddingFinished** ()
- void **reAddItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< [QVariant](#) > &extraValues)

### Public Member Functions

- [QList](#)< [Album](#) \* > **currentAlbums** () const
- bool **hasScheduledRefresh** () const
- bool **isListingOnlyAvailableImages** () const
- bool **isRecurseAlbums** () const
- bool **isRecurseTags** () const
- [ItemAlbumModel](#) ([QWidget](#) \*const parent)

### Public Member Functions inherited from [Digikam::ItemThumbnailModel](#)

- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [ItemThumbnailModel](#) ([QWidget](#) \*const parent)
- bool **setData** (const [QModelIndex](#) &index, const [QVariant](#) &value, int role=[Qt::DisplayRole](#)) override
- void **setEmitDataChanged** (bool emitSignal)
- void **setExifRotate** (bool rotate)
- void **setPreloadThumbnails** (bool preload)
- void **setPreloadThumbnailSize** (const [ThumbnailSize](#) &thumbSize)
  - If you want to fix a size for preloading, do it here.*
- void **setThumbnailLoadThread** ([ThumbnailLoadThread](#) \*const thread)
- void **setThumbnailSize** (const [ThumbnailSize](#) &thumbSize)
  - Set the thumbnail size to use.*
- [ThumbnailLoadThread](#) \* **thumbnailLoadThread** () const
- [ThumbnailSize](#) **thumbnailSize** () const

## Public Member Functions inherited from [Digikam::ItemModel](#)

- void [addItemInfo](#) (const [ItemInfo](#) &info)
- void **addItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **addItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void **addItemInfosSynchronously** (const QList< [ItemInfo](#) > &infos)
- void **addItemInfosSynchronously** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void [addItemInfoSynchronously](#) (const [ItemInfo](#) &info)
- void [clearItemInfos](#) ()
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- void [ensureHasGroupedImages](#) (const [ItemInfo](#) &groupLeader)
- void [ensureHasItemInfo](#) (const [ItemInfo](#) &info)
- void **ensureHasItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **ensureHasItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **hasImage** (const [ItemInfo](#) &info) const
- bool **hasImage** (const [ItemInfo](#) &info, const QVariant &extraValue) const
- bool **hasImage** (qulonglong id) const
- bool **hasImage** (qulonglong id, const QVariant &extraValue) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- qulonglong **imageld** (const QModelIndex &index) const
- qulonglong **imageld** (int row) const
- QList< qulonglong > **imagelds** () const
- QList< qulonglong > **imagelds** (const QList< QModelIndex > &indexes) const
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfo](#) **imageInfo** (const QString &filePath) const
- [ItemInfo](#) **imageInfo** (int row) const
- [ItemInfo](#) & **imageInfoRef** (const QModelIndex &index) const
- [ItemInfo](#) & **imageInfoRef** (int row) const
- QList< [ItemInfo](#) > **imageInfos** () const
- QList< [ItemInfo](#) > **imageInfos** (const QList< QModelIndex > &indexes) const
- QList< [ItemInfo](#) > **imageInfos** (const QString &filePath) const
- QModelIndex **index** (int row, int column=0, const QModelIndex &parent=QModelIndex()) const override
- QList< QModelIndex > **indexesForImageld** (qulonglong id) const
- QList< QModelIndex > **indexesForItemInfo** (const [ItemInfo](#) &info) const
- QList< QModelIndex > **indexesForPath** (const QString &filePath) const
- QModelIndex **indexForImageld** (qulonglong id) const
- QModelIndex **indexForImageld** (qulonglong id, const QVariant &extraValue) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info, const QVariant &extraValue) const
- QModelIndex **indexForPath** (const QString &filePath) const
- bool **isEmpty** () const
- bool **isRefreshing** () const
- int **itemCount** () const
- [ItemModel](#) (QObject \*const parent=nullptr)
- bool **keepsFilePathCache** () const
- int **numberOfIndexesForImageld** (qulonglong id) const
- int **numberOfIndexesForItemInfo** (const [ItemInfo](#) &info) const
- void [removeIndex](#) (const QModelIndex &indexes)
- void **removeIndexes** (const QList< QModelIndex > &indexes)
- void **removeItemInfo** (const [ItemInfo](#) &info)
- void **removeItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **removeItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void [setItemInfos](#) (const QList< [ItemInfo](#) > &infos)



- void [setKeepsFilePathCache](#) (bool keepCache)
- DECLARE\_MODEL\_DRAG\_DROP\_METHODS void [setPreprocessor](#) (QObject \*const processor)
- void [setSendRemovalSignals](#) (bool send)
- void [setWatchFlags](#) (const [DatabaseFields::Set](#) &set)
- QList< [ItemInfo](#) > [uniqueItemInfos](#) () const
- void [unsetPreprocessor](#) (QObject \*const processor)

## Public Member Functions inherited from [Digikam::DragDropModelImplementation](#)

- virtual Qt::ItemFlags [dragDropFlags](#) (const QModelIndex &index) const
- Qt::ItemFlags [dragDropFlagsV2](#) (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* [dragDropHandler](#) () const
- [DragDropModelImplementation](#) ()=default
- bool [dropMimeData](#) (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool [isDragEnabled](#) (const QModelIndex &index) const
- virtual bool [isDropEnabled](#) (const QModelIndex &index) const
- QMimeData \* [mimeData](#) (const QModelIndexList &indexes) const
- QStringList [mimeTypes](#) () const
- void [setDragDropHandler](#) ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions [supportedDropActions](#) () const

## Protected Slots

- void [incrementalRefresh](#) ()
- void [scheduleIncrementalRefresh](#) ()
- void [slotAlbumAdded](#) ([Album](#) \*album)
- void [slotAlbumDeleted](#) ([Album](#) \*album)
- void [slotAlbumRenamed](#) ([Album](#) \*album)
- void [slotAlbumsCleared](#) ()
- void [slotCollectionImageChange](#) (const [CollectionImageChangeset](#) &changeset)
- void [slotData](#) (const QList< [ItemLISTERRecord](#) > &records)
- void [slotImageChange](#) (const [ImageChangeset](#) &changeset) override
- void [slotImageTagChange](#) (const [ImageTagChangeset](#) &changeset) override
- void [slotNextIncrementalRefresh](#) ()
- void [slotResult](#) ()
- void [slotSearchChange](#) (const [SearchChangeset](#) &changeset)

## Protected Slots inherited from [Digikam::ItemThumbnailModel](#)

- void [slotThumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const QPixmap &thumb)
- void [slotThumbnailLoadedFromStorage](#) (const [LoadingDescription](#) &loadingDescription, const QPixmap &thumb)

## Protected Slots inherited from [Digikam::ItemModel](#)

- virtual void [slotAlbumChange](#) (const [AlbumChangeset](#) &changeset)
- virtual void [slotImageChange](#) (const [ImageChangeset](#) &changeset)
- virtual void [slotImageTagChange](#) (const [ImageTagChangeset](#) &changeset)

### Protected Member Functions

- void **startListJob** (const QList< Album \* > &albums)

### Protected Member Functions inherited from [Digikam::ItemThumbnailModel](#)

- void [imageInfosCleared](#) () override

### Protected Member Functions inherited from [Digikam::ItemModel](#)

- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelection** (const QItemSelection &selection)
- void **finishIncrementalRefresh** ()
- void **finishRefresh** ()
- bool **hasIncrementalRefreshPending** () const
- virtual void [imageInfosAboutToBeRemoved](#) (int, int)
- void [requestIncrementalRefresh](#) ()
- void [startIncrementalRefresh](#) ()
- void [startRefresh](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::ItemModel](#)

- enum [ItemModelRoles](#) {  
[ItemModelPointerRole](#) = Qt::UserRole , [ItemModelInternalId](#) = Qt::UserRole + 1 , [ThumbnailRole](#) = Qt::UserRole + 2 , [CreationDateRole](#) = Qt::UserRole + 3 ,  
[ExtraDataRole](#) = Qt::UserRole + 5 , [ExtraDataDuplicateCount](#) = Qt::UserRole + 6 , [LTLeftPanelRole](#) = Qt::UserRole + 50 , [LTRightPanelRole](#) = Qt::UserRole + 51 ,  
[SubclassRoles](#) = Qt::UserRole + 100 , [FilterModelRoles](#) = Qt::UserRole + 500 }

### Signals inherited from [Digikam::ItemThumbnailModel](#)

- void **thumbnailAvailable** (const QModelIndex &index, int requestedSize)
- void **thumbnailFailed** (const QModelIndex &index, int requestedSize)

### Signals inherited from [Digikam::ItemModel](#)

- void [allRefreshingFinished](#) ()
- void [imageChange](#) (const [ImageChangeset](#) &, const QItemSelection &)
- void [imageInfosAboutToBeAdded](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosAboutToBeRemoved](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosAdded](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosRemoved](#) (const QList< [ItemInfo](#) > &infos)
- void [imageTagChange](#) (const [ImageTagChangeset](#) &, const QItemSelection &)
- void [preprocess](#) (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &)
- void **processAdded** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &)
- void [readyForIncrementalRefresh](#) ()

## Static Public Member Functions inherited from [Digikam::ItemModel](#)

- static qlonglong [retrievelmageld](#) (const QModelIndex &index)
- static [ItemInfo](#) [retrievelitemInfo](#) (const QModelIndex &index)

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* `m_dragDropHandler` = nullptr

## 6.877.1 Member Function Documentation

### 6.877.1.1 `openAlbum`

```
void Digikam::ItemAlbumModel::openAlbum (
    const QList< Album * > & albums ) [slot]
```

Call this method to populate the model with data from the given album. If called with 0, the model will be empty. Opening the same album again is a no-op. Extra safety, ensure that no null pointers are added

### 6.877.1.2 `refresh`

```
void Digikam::ItemAlbumModel::refresh ( ) [slot]
```

Reloads the current album

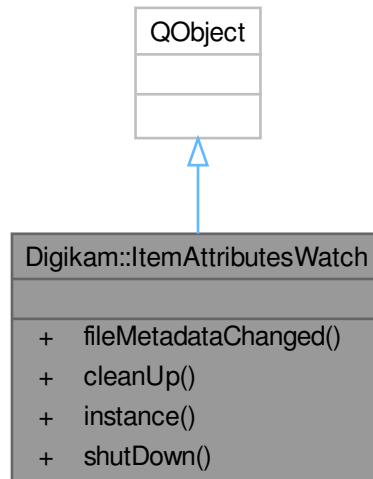
### 6.877.1.3 `slotImageChange`

```
void Digikam::ItemAlbumModel::slotImageChange (
    const ImageChangeset & changeset ) [override], [protected], [slot]
```

QList is designed for multiple selection, for now, only tags are supported for [SAlbum](#) it will be a list with one element

## 6.878 Digikam::ItemAttributesWatch Class Reference

Inheritance diagram for Digikam::ItemAttributesWatch:



### Signals

- void [signalFileMetadataChanged](#) (const `QUrl` &url)
- void **signalImageCaptionChanged** (qulonglong imageId)
- void **signalImageDateChanged** (qulonglong imageId)
- void [signalImageRatingChanged](#) (qulonglong imageId)
- void [signalImagesChanged](#) (int albumId)
- void [signalImageTagsChanged](#) (qulonglong imageId)

### Public Member Functions

- void **fileMetadataChanged** (const `QUrl` &url)

### Static Public Member Functions

- static void **cleanUp** ()
- static [ItemAttributesWatch](#) \* **instance** ()
- static void **shutDown** ()

## 6.878.1 Member Function Documentation

### 6.878.1.1 signalFileMetadataChanged

```
void Digikam::ItemAttributesWatch::signalFileMetadataChanged (
    const QUrl & url ) [signal]
```

Indicates that the metadata if the given file has been changed (a write operation on the file on disk). Usually, the database is updated accordingly, so then this signal is sent in combination with one or more of the above signals.

### 6.878.1.2 signalImageRatingChanged

```
void Digikam::ItemAttributesWatch::signalImageRatingChanged (
    qlonglong imageId ) [signal]
```

These signals indicated that the rating, data or caption of the image with given *imageId* was set. There is no guarantee that it actually changed.

### 6.878.1.3 signalImagesChanged

```
void Digikam::ItemAttributesWatch::signalImagesChanged (
    int albumId ) [signal]
```

Indicates that images in the given *albumId* may have changed their tags. This signal, the signal above, or both may be sent.

### 6.878.1.4 signalImageTagsChanged

```
void Digikam::ItemAttributesWatch::signalImageTagsChanged (
    qlonglong imageId ) [signal]
```

Indicates that tags have been assigned or removed for image with given *imageId*. There is no guarantee that the tags were actually changed. This signal, the signal below, or both may be sent.



- void [setCurrentInfo](#) (const [ItemInfo](#) &info)
- void [setCurrentUrl](#) (const [QUrl](#) &url)
- void [setCurrentUrlWhenAvailable](#) (const [QUrl](#) &url)
- void [setCurrentWhenAvailable](#) (qulonglong imageld)
- void [setSelectedItemInfos](#) (const [QList](#)< [ItemInfo](#) > &infos)
- void [setSelectedUrls](#) (const [QList](#)< [QUrl](#) > &urlList)
- void [setThumbnailSize](#) (int size)

### Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void [copy](#) () override
- void [cut](#) () override
- void [hideIndexNotification](#) ()
- void [paste](#) () override
- void [showIndexNotification](#) (const [QModelIndex](#) &index, const [QString](#) &message)

### Public Slots inherited from [Digikam::DCategorizedView](#)

- void [reset](#) () override

### Signals

- void [currentChanged](#) (const [ItemInfo](#) &info)
- void [deselected](#) (const [QList](#)< [ItemInfo](#) > &nowDeselectedInfos)  
*Emitted when items are deselected. There may be other selected infos left. This signal is not emitted when the model is reset; then only selectionCleared is emitted.*
- void [imageActivated](#) (const [ItemInfo](#) &info)  
*Emitted when the given image is activated. Info is never null.*
- void [modelChanged](#) ()  
*Emitted when a new model is set.*
- void [selected](#) (const [QList](#)< [ItemInfo](#) > &newSelectedInfos)  
*Emitted when new items are selected. The parameter includes only the newly selected infos, there may be other already selected infos.*

### Signals inherited from [Digikam::ItemViewCategorized](#)

- void [clicked](#) (const [QMouseEvent](#) \*e, const [QModelIndex](#) &index)
- void [entered](#) (const [QMouseEvent](#) \*e, const [QModelIndex](#) &index)
- void [keyPressed](#) ([QKeyEvent](#) \*e)
- void [selectionChanged](#) ()
- void [selectionCleared](#) ()
- void [viewportClicked](#) (const [QMouseEvent](#) \*e)
- void [zoomInStep](#) ()
- void [zoomOutStep](#) ()

## Public Member Functions

- void **addOverlay** ([ItemDelegateOverlay](#) \*overlay, [ItemDelegate](#) \*delegate=nullptr)
 

*Add and remove an overlay. It will as well be removed automatically when destroyed. Unless you pass a different delegate, the current delegate will be used.*
- void **addSelectionOverlay** ([ItemDelegate](#) \*delegate=nullptr)
- [Album](#) \* **albumAt** (const [QPoint](#) &pos) const
- [ItemInfoList](#) **allItemInfos** () const
- [QList](#)< [QUrl](#) > **allUrls** () const
- [Album](#) \* **currentAlbum** () const
- [ItemInfo](#) **currentInfo** () const
- [QUrl](#) **currentUrl** () const
- [ItemDelegate](#) \* **delegate** () const
- [QItemSelectionModel](#) \* **getSelectionModel** () const
- [ItemAlbumFilterModel](#) \* **imageAlbumFilterModel** () const
- [ItemAlbumModel](#) \* **imageAlbumModel** () const
 

*Returns 0 if the [ItemModel](#) is not an [ItemAlbumModel](#).*
- [ItemFilterModel](#) \* **imageFilterModel** () const
 

*Returns any [ItemFilterMode](#) in chain. May not be [sourceModel\(\)](#)*
- [ItemModel](#) \* **imageModel** () const
- [ImageSortFilterModel](#) \* **imageSortFilterModel** () const
- [ItemThumbnailModel](#) \* **imageThumbnailModel** () const
 

*Returns 0 if the [ItemModel](#) is not an [ItemThumbnailModel](#).*
- [QModelIndex](#) **indexForInfo** (const [ItemInfo](#) &info) const
- [ItemCategorizedView](#) ([QWidget](#) \*const parent=nullptr)
- [ItemInfo](#) **nextInfo** (const [ItemInfo](#) &info)
- [ItemInfo](#) **nextInOrder** (const [ItemInfo](#) &startingPoint, int nth)
- [ItemInfo](#) **previousInfo** (const [ItemInfo](#) &info)
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemInfoList](#) **selectedItemInfos** () const
- [ItemInfoList](#) **selectedItemInfosCurrentFirst** () const
- void **setModels** ([ItemModel](#) \*model, [ImageSortFilterModel](#) \*filterModel)
- virtual void **setThumbnailSize** (const [ThumbnailSize](#) &size)
- [ThumbnailSize](#) **thumbnailSize** () const
- void **toIndex** (const [QUrl](#) &url)

## Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **awayFromSelection** ()
- [DItemDelegate](#) \* **delegate** () const
- void **invertSelection** ()
- bool **isToolTipEnabled** () const
- [ItemViewCategorized](#) ([QWidget](#) \*const parent=nullptr)
- int **numberOfSelectedIndexes** () const
- void **scrollTo** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible) override
- void **scrollToRelaxed** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible)
- void **setInitialSelectedItem** (bool enabled)
- void **setScrollCurrentToCenter** (bool enabled)
- void **setScrollStepGranularity** (int factor)
- void **setSelectedIndexes** (const [QList](#)< [QModelIndex](#) > &indexes)
- void **setSpacing** (int spacing)
- void **setToolTipEnabled** (bool enabled)
- void **setUsePointingHandCursor** (bool useCursor)
- void **toFirstIndex** ()
- void **toIndex** (const [QModelIndex](#) &index)
- void **toLastIndex** ()
- void **toNextIndex** ()
- void **toPreviousIndex** ()



## Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual QModelIndexList [categorizedIndexesIn](#) (const QRect &rect) const
- virtual QModelIndex [categoryAt](#) (const QPoint &point) const
- [DCategoryDrawer](#) \* [categoryDrawer](#) () const
- virtual QItemSelectionRange [categoryRange](#) (const QModelIndex &index) const
- virtual QRect [categoryVisualRect](#) (const QModelIndex &index) const
- [DCategorizedView](#) (QWidget \*const parent=nullptr)
- QModelIndex [indexAt](#) (const QPoint &point) const override
- void [setCategoryDrawer](#) ([DCategoryDrawer](#) \*categoryDrawer)
- void [setDrawDraggedItems](#) (bool drawDraggedItems)
- void [setGridSize](#) (const QSize &size)
- void [setModel](#) (QAbstractItemModel \*model) override
- QRect [visualRect](#) (const QModelIndex &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void [copy](#) ()
- virtual void [cut](#) ()
- virtual void [paste](#) ()

## Protected Slots

- void [slotCurrentUriTimer](#) ()
- void [slotItemInfosAdded](#) ()

## Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void [layoutAboutToBeChanged](#) ()
- void [layoutWasChanged](#) ()
- void [slotActivated](#) (const QModelIndex &index)
- void [slotClicked](#) (const QModelIndex &index)
- void [slotEntered](#) (const QModelIndex &index)
- virtual void [slotSetupChanged](#) ()
- virtual void [slotThemeChanged](#) ()

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void [currentChanged](#) (const QModelIndex &current, const QModelIndex &previous) override
- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- virtual void [rowsInsertedArtificial](#) (const QModelIndex &parent, int start, int end)
- virtual void [slotLayoutChanged](#) ()
- void [updateGeometries](#) () override

## Protected Member Functions

- virtual void **activated** (const [ItemInfo](#) &info, Qt::KeyboardModifiers modifiers)
  - Reimplement these in a subclass.*
- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfoList](#) **imageInfos** (const QList< QModelIndex > &indexes) const
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- void **installDefaultModels** ()
  - install default [ItemAlbumModel](#) and filter model, ready for use*
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([ItemDelegate](#) \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override
  - Reimplement these in a subclass.*
- virtual void **showContextMenuOnInfo** (QContextMenuEvent \*event, const [ItemInfo](#) &info)
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override
  - reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([DItemDelegate](#) \*delegate)
- void **setToolTip** ([ItemViewToolTip](#) \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

## Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

### 6.879.1 Member Function Documentation

#### 6.879.1.1 activated()

```
void Digikam::ItemCategorizedView::activated (
    const ItemInfo & info,
    Qt::KeyboardModifiers modifiers ) [protected], [virtual]
```

Reimplemented in [Digikam::DigikamItemView](#).

#### 6.879.1.2 albumAt()

```
Album * Digikam::ItemCategorizedView::albumAt (
    const QPoint & pos ) const
```

If the model is categorized by an album, returns the album of the category that contains the position. If this is not applicable, return the current album. May return 0.

#### 6.879.1.3 dragDropHandler()

```
AbstractItemDragDropHandler * Digikam::ItemCategorizedView::dragDropHandler ( ) const [override],
[protected], [virtual]
```

You need to implement these three methods Returns the drag drop handler.

Implements [Digikam::DragDropViewImplementation](#).

#### 6.879.1.4 filterModel()

```
QSortFilterProxyModel * Digikam::ItemCategorizedView::filterModel ( ) const [override], [protected], [virtual]
```

Implements [Digikam::ItemViewCategorized](#).

#### 6.879.1.5 hintAt

```
void Digikam::ItemCategorizedView::hintAt (
    const ItemInfo & info ) [slot]
```

Does something to gain attention for info, but not changing current selection.

#### 6.879.1.6 indexActivated()

```
void Digikam::ItemCategorizedView::indexActivated (
    const QModelIndex & index,
    Qt::KeyboardModifiers modifiers ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

#### 6.879.1.7 nextIndexHint()

```
QModelIndex Digikam::ItemCategorizedView::nextIndexHint (
    const QModelIndex & indexToAnchor,
    const QItemSelectionRange & removed ) const [override], [protected], [virtual]
```

Assuming the given indexes would be removed (hypothetically!), return the index to be selected instead, starting from anchor. The default implementation returns the next remaining sibling.

Reimplemented from [Digikam::ItemViewCategorized](#).

#### 6.879.1.8 nextInOrder()

```
ItemInfo Digikam::ItemCategorizedView::nextInOrder (
    const ItemInfo & startingPoint,
    int nth )
```

Returns the n-th info after the given one. Specifically, return the previous info for nth = -1 and the next info for n = 1. Returns a null info if either startingPoint or the nth info are not contained in the model.

#### 6.879.1.9 setCurrentInfo

```
void Digikam::ItemCategorizedView::setCurrentInfo (
    const ItemInfo & info ) [slot]
```

Set as current item the item identified by the imageinfo.

#### 6.879.1.10 setCurrentUrl

```
void Digikam::ItemCategorizedView::setCurrentUrl (
    const QUrl & url ) [slot]
```

Set as current item the item identified by its file url.

#### 6.879.1.11 setCurrentUrlWhenAvailable

```
void Digikam::ItemCategorizedView::setCurrentUrlWhenAvailable (
    const QUrl & url ) [slot]
```

Set as current item when it becomes available, the item identified by its file url.

#### 6.879.1.12 setCurrentWhenAvailable

```
void Digikam::ItemCategorizedView::setCurrentWhenAvailable (
    qlonglong imageId ) [slot]
```

Scroll the view to the given item when it becomes available.

#### 6.879.1.13 setSelectedItemInfos

```
void Digikam::ItemCategorizedView::setSelectedItemInfos (
    const QList< ItemInfo > & infos ) [slot]
```

Set selected items.

#### 6.879.1.14 setSelectedUrls

```
void Digikam::ItemCategorizedView::setSelectedUrls (
    const QList< QUrl > & urlList ) [slot]
```

Set selected items identified by their file urls.

#### 6.879.1.15 showContextMenuOnIndex()

```
void Digikam::ItemCategorizedView::showContextMenuOnIndex (
    QContextMenuEvent * event,
    const QModelIndex & index ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

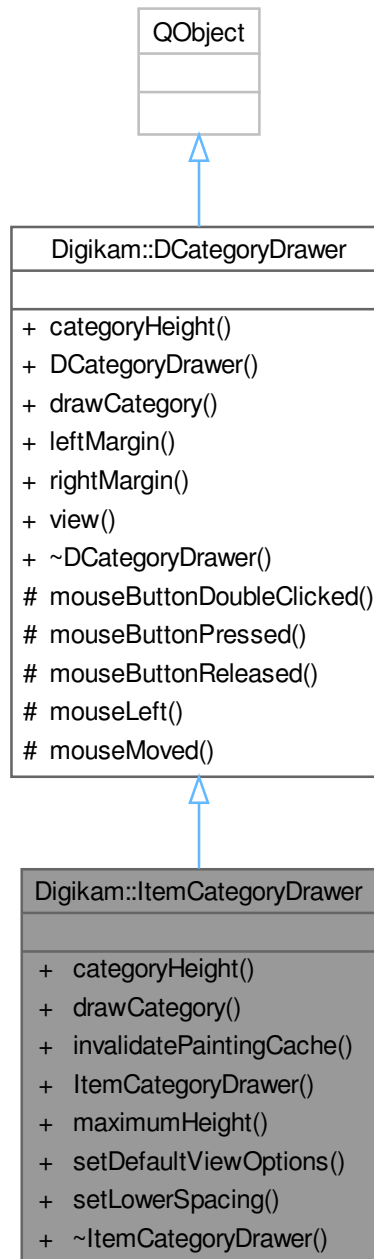
#### 6.879.1.16 toIndex()

```
void Digikam::ItemCategorizedView::toIndex (
    const QUrl & url )
```

Selects the index as current and scrolls to it.

## 6.880 Digikam::ItemCategoryDrawer Class Reference

Inheritance diagram for Digikam::ItemCategoryDrawer:



### Public Member Functions

- int [categoryHeight](#) (const QModelIndex &index, const QStyleOption &option) const override
- void [drawCategory](#) (const QModelIndex &index, int sortRole, const QStyleOption &option, QPainter \*painter) const override

- void `invalidatePaintingCache` ()
- `ItemCategoryDrawer` (`ItemCategorizedView` \*const parent)
- virtual int `maximumHeight` () const
- void `setDefaultViewOptions` (const `QStyleOptionViewItem` &option)
- void `setLowerSpacing` (int spacing)

## Public Member Functions inherited from `Digikam::DCategoryDrawer`

- `DCategoryDrawer` (`DCategorizedView` \*const view)
- virtual int `leftMargin` () const
- virtual int `rightMargin` () const
- `DCategorizedView` \* view () const

## Additional Inherited Members

## Signals inherited from `Digikam::DCategoryDrawer`

- void `actionRequested` (int action, const `QModelIndex` &index)
- void `collapseOrExpandClicked` (const `QModelIndex` &index)

## Protected Member Functions inherited from `Digikam::DCategoryDrawer`

- virtual void `mouseButtonDoubleClicked` (const `QModelIndex` &index, const `QRect` &blockRect, `QMouseEvent` \*event)
- virtual void `mouseButtonPressed` (const `QModelIndex` &index, const `QRect` &blockRect, `QMouseEvent` \*event)
- virtual void `mouseButtonReleased` (const `QModelIndex` &index, const `QRect` &blockRect, `QMouseEvent` \*event)
- virtual void `mouseLeft` (const `QModelIndex` &index, const `QRect` &blockRect)
- virtual void `mouseMoved` (const `QModelIndex` &index, const `QRect` &blockRect, `QMouseEvent` \*event)

## 6.880.1 Member Function Documentation

### 6.880.1.1 `categoryHeight()`

```
int Digikam::ItemCategoryDrawer::categoryHeight (
    const QModelIndex & index,
    const QStyleOption & option ) const [override], [virtual]
```

#### Returns

The category height for the category represented by index `index` with style options `option`.

Reimplemented from `Digikam::DCategoryDrawer`.

### 6.880.1.2 `drawCategory()`

```
void Digikam::ItemCategoryDrawer::drawCategory (
    const QModelIndex & index,
    int sortRole,
    const QStyleOption & option,
    QPainter * painter ) const [override], [virtual]
```

This method purpose is to draw a category represented by the given

## Parameters

<i>index</i>	with the given
<i>sortRole</i>	sorting role
<i>option</i>	painter style options
<i>painter</i>	painter instance

## Note

This method will be called one time per category, always with the first element in that category

Reimplemented from [Digikam::DCategoryDrawer](#).

## 6.881 Digikam::ItemChangeHint Class Reference

### Public Types

- enum [ChangeType](#) { [ItemModified](#) , [ItemRescan](#) }

### Public Member Functions

- [ChangeType](#) **changeType** () const
- [QList](#)< [qlonglong](#) > **ids** () const
- bool **isId** ([qlonglong](#) id) const
- bool **isModified** () const
- **ItemChangeHint** (const [QList](#)< [qlonglong](#) > &srcIds, [ChangeType](#) type=[ItemModified](#))
- bool **needsRescan** () const

### Protected Attributes

- [QList](#)< [qlonglong](#) > **m\_ids**
- [ChangeType](#) **m\_type** = [ItemModified](#)

### 6.881.1 Member Enumeration Documentation

#### 6.881.1.1 ChangeType

enum [Digikam::ItemChangeHint::ChangeType](#)

An [ItemCopyMoveHint](#) describes a list of existing items that should be updated although the modification date may not have changed.

#### Enumerator

ItemModified	treat as if modification date changed
ItemRescan	reread metadata



## 6.882 Digikam::ItemComments Class Reference

### Public Types

- enum [LanguageChoiceBehavior](#) { [ReturnMatchingLanguageOnly](#) , [ReturnMatchingOrDefaultLanguage](#) , [ReturnMatchingDefaultOrFirstLanguage](#) }
- enum [UniqueBehavior](#) { [UniquePerLanguage](#) , [UniquePerLanguageAndAuthor](#) }

### Public Member Functions

- void [addComment](#) (const QString &comment, const QString &language=QString(), const QString &author=QString(), const QDateTime &date=QDateTime(), DatabaseComment::Type [type](#)=DatabaseComment::Comment)
- void [addHeadline](#) (const QString &headline, const QString &language=QString(), const QString &author=QString(), const QDateTime &date=QDateTime())
- void [addTitle](#) (const QString &title, const QString &language=QString(), const QString &author=QString(), const QDateTime &date=QDateTime())
- void [apply](#) ()
- void [apply](#) (CoreDbAccess &access)
- QString [author](#) (int index) const
- void [changeAuthor](#) (int index, const QString &author)
- void [changeComment](#) (int index, const QString &comment)
- void [changeDate](#) (int index, const QDateTime &date)
- void [changeLanguage](#) (int index, const QString &language)
- void [changeType](#) (int index, DatabaseComment::Type [type](#))
- QString [comment](#) (int index) const
- QString [commentForLanguage](#) (const QString &languageCode, int \*const index=nullptr, [LanguageChoiceBehavior](#) behavior=[ReturnMatchingDefaultOrFirstLanguage](#)) const
- QDateTime [date](#) (int index) const
- QString [defaultComment](#) (DatabaseComment::Type [type](#)=DatabaseComment::Comment) const
- QString [defaultComment](#) (int \*const index, Digikam::DatabaseComment::Type [type](#)=DatabaseComment::Comment) const
- bool [isNull](#) () const
- [ItemComments](#) ()
- [ItemComments](#) (const CoreDbAccess &access, qlonglong imageid)
- [ItemComments](#) (const [ItemComments](#) &other)
- [ItemComments](#) (qlonglong imageid)
- QString [language](#) (int index) const
- int [numberOfComments](#) () const
- [ItemComments](#) & [operator=](#) (const [ItemComments](#) &other)
- void [remove](#) (int index)
- void [removeAll](#) ()
- void [removeAll](#) (DatabaseComment::Type [type](#))
- void [removeAllComments](#) ()
- void [replaceComments](#) (const [CaptionsMap](#) &comments, DatabaseComment::Type [type](#)=DatabaseComment::Comment)
- void [replaceFrom](#) (const [ItemComments](#) &source)
- void [setUniqueBehavior](#) ([UniqueBehavior](#) behavior)
- [CaptionsMap toCaptionsMap](#) (DatabaseComment::Type=DatabaseComment::Comment) const
- DatabaseComment::Type [type](#) (int index) const

## Protected Member Functions

- void **addCommentDirectly** (const QString &comment, const QString &language, const QString &author, DatabaseComment::Type type, const QDateTime &date)

## Protected Attributes

- QSharedPointer< Private > d

## 6.882.1 Member Enumeration Documentation

### 6.882.1.1 LanguageChoiceBehavior

```
enum Digikam::ItemComments::LanguageChoiceBehavior
```

The [ItemComments](#) class shall provide short-lived objects that provide read/write access to the comments stored in the database. It is a mere wrapper around the less convenient access methods in [CoreDB](#). Database results are cached, but the object will not listen to database changes from other places.

Changes are applied to the database only after calling [apply\(\)](#), which you can call any time and which will in any case be called from the destructor.

#### Enumerator

ReturnMatchingLanguageOnly	Return only a comment if the language code (at least the language code, the country part may differ) is identical. Else returns a null QString.
ReturnMatchingOrDefaultLanguage	If no matching language as above is found, return the default language.
ReturnMatchingDefaultOrFirstLanguage	If no matching or default language is found, return the first comment. Returns a null string only if no comment is available.

### 6.882.1.2 UniqueBehavior

```
enum Digikam::ItemComments::UniqueBehavior
```

#### Enumerator

UniquePerLanguage	Allow only one comment per language. Default setting.
UniquePerLanguageAndAuthor	Allow multiple comments per language, each with a different author

## 6.882.2 Constructor & Destructor Documentation

### 6.882.2.1 ItemComments() [1/3]

```
Digikam::ItemComments::ItemComments ( )
```

Create a null [ItemComments](#) object

### 6.882.2.2 ItemComments() [2/3]

```
Digikam::ItemComments::ItemComments (
    qlonglong imageid ) [explicit]
```

Create a [ItemComments](#) object for the image with the specified id.

### 6.882.2.3 ItemComments() [3/3]

```
Digikam::ItemComments::ItemComments (
    const CoreDbAccess & access,
    qlonglong imageid )
```

Create a [ItemComments](#) object for the image with the specified id. The existing [CoreDbAccess](#) object will be used to access the database.

## 6.882.3 Member Function Documentation

### 6.882.3.1 addComment()

```
void Digikam::ItemComments::addComment (
    const QString & comment,
    const QString & language = QString(),
    const QString & author = QString(),
    const QDateTime & date = QDateTime(),
    DatabaseComment::Type type = DatabaseComment::Comment )
```

Add a new comment to the list of normal image comments, specified with language and author. Checking for unique comments is done as set by `setUniqueBehavior`. If you pass a null string as language, it will be translated to the language code designating the default language ("x-default"). If you just want to change the one comment of the image, call `addComment(myComment)`;

### 6.882.3.2 addHeadline()

```
void Digikam::ItemComments::addHeadline (
    const QString & headline,
    const QString & language = QString(),
    const QString & author = QString(),
    const QDateTime & date = QDateTime() )
```

Convenience method to add a comment of type Headline. Calls `addComment`, see above for more info.

### 6.882.3.3 addTitle()

```
void Digikam::ItemComments::addTitle (
    const QString & title,
    const QString & language = QString(),
    const QString & author = QString(),
    const QDateTime & date = QDateTime() )
```

Convenience method to add a comment of type Headline. Calls `addComment`, see above for more info.

#### 6.882.3.4 apply()

```
void Digikam::ItemComments::apply ( )
```

Apply all changes. Also called in destructor, so you typically do not need to call this.

#### 6.882.3.5 changeComment()

```
void Digikam::ItemComments::changeComment (
    int index,
    const QString & comment )
```

Access individual properties. Please ensure that the specified index is a valid index

#### 6.882.3.6 commentForLanguage()

```
QString Digikam::ItemComments::commentForLanguage (
    const QString & languageCode,
    int *const index = nullptr,
    LanguageChoiceBehavior behavior = ReturnMatchingDefaultOrFirstLanguage ) const
```

Returns a comment for the specified language. Matching behavior can be specified. Optionally also returns the index with which you can access further information about the comment.

#### 6.882.3.7 defaultComment()

```
QString Digikam::ItemComments::defaultComment (
    DatabaseComment::Type type = DatabaseComment::Comment ) const
```

This methods presents one of the comment strings of the available comment as the default value, when you just want to have one string. Optionally also returns the index with which you can access further information about the comment.

#### 6.882.3.8 language()

```
QString Digikam::ItemComments::language (
    int index ) const
```

RFC 3066 notation, or "x-default"

#### 6.882.3.9 numberOfComments()

```
int Digikam::ItemComments::numberOfComments ( ) const
```

Returns the number of comments available.

### 6.882.3.10 remove()

```
void Digikam::ItemComments::remove (
    int index )
```

Remove the entry referred to by index.

### 6.882.3.11 removeAll() [1/2]

```
void Digikam::ItemComments::removeAll ( )
```

Remove all entries of all types: Comments, Headlines, Titles

### 6.882.3.12 removeAll() [2/2]

```
void Digikam::ItemComments::removeAll (
    DatabaseComment::Type type )
```

Remove all entries of the given type

### 6.882.3.13 removeAllComments()

```
void Digikam::ItemComments::removeAllComments ( )
```

Convenience method: remove all entries of type Comment

### 6.882.3.14 replaceComments()

```
void Digikam::ItemComments::replaceComments (
    const CaptionsMap & comments,
    DatabaseComment::Type type = DatabaseComment::Comment )
```

Replaces all existing comments with the given set of comments and associated language. Optionally date and author can be specified in [CaptionsMap](#) container.

### 6.882.3.15 replaceFrom()

```
void Digikam::ItemComments::replaceFrom (
    const ItemComments & source )
```

Replaces all entries in this object with all entries from source.

### 6.882.3.16 setUniqueBehavior()

```
void Digikam::ItemComments::setUniqueBehavior (
    UniqueBehavior behavior )
```

Changes the behavior to unique comments per language, see the enum above for possible values. Default value is UniquePerLanguage. Note: This is *not* a property of the database, but only of this single [ItemComments](#) object,

**6.882.3.17 toCaptionsMap()**

```
CaptionsMap Digikam::ItemComments::toCaptionsMap (  
    DatabaseComment::Type type = DatabaseComment::Comment ) const
```

Returns all entries of the given type in a [CaptionsMap](#) container.

**6.882.3.18 type()**

```
DatabaseComment::Type Digikam::ItemComments::type (  
    int index ) const
```

Access individual properties. Please ensure that the specified index is a valid index

## 6.883 Digikam::ItemCoordinatesOverlay Class Reference

Inheritance diagram for Digikam::ItemCoordinatesOverlay:



### Public Member Functions

- `CoordinatesOverlayWidget * buttonWidget () const`
- `ItemCoordinatesOverlay (QObject *const parent)`

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- QWidget \* **createWidget** () override
- void **setActive** (bool active) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **visualChange** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **checkIndexOnEnter** (const QModelIndex &index) const
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual void **hide** ()
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* **parentWidget** () const
- virtual void **viewportLeaveEvent** (QObject \*obj, QEvent \*event)
- virtual void **widgetEnterEvent** ()
- void **widgetEnterNotifyMultiple** (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool **affectsMultiple** (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool **viewHasMultiSelection** () const

## Protected Attributes

- QPersistentModelIndex **m\_index**



**Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- bool **m\_mouseButtonPressedOnWidget** = false
- QWidget \* **m\_widget** = nullptr

**Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)**

- QAbstractItemDelegate \* **m\_delegate** = nullptr
- QAbstractItemView \* **m\_view** = nullptr

**Additional Inherited Members****Signals inherited from [Digikam::ItemDelegateOverlay](#)**

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

**Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

**Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)****6.883.1 Member Function Documentation****6.883.1.1 [checkIndex\(\)](#)**

```
bool Digikam::ItemCoordinatesOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

**6.883.1.2 [createWidget\(\)](#)**

```
QWidget * Digikam::ItemCoordinatesOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.883.1.3 setActive()

```
void Digikam::ItemCoordinatesOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.883.1.4 slotEntered()

```
void Digikam::ItemCoordinatesOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.883.1.5 visualChange()

```
void Digikam::ItemCoordinatesOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

## 6.884 Digikam::ItemCopyMoveHint Class Reference

### Public Member Functions

- int **albumIdDst** () const
- int **albumRootIdDst** () const
- CollectionScannerHints::Album **dst** () const
- QString **dstName** (qlonglong id) const
- QStringList **dstNames** () const
- bool **isDstAlbum** (int albumRootId, int albumId) const
- bool **isSrcId** (qlonglong id) const
- [ItemCopyMoveHint](#) ()=default
- **ItemCopyMoveHint** (const QList< qlonglong > &srcIds, int dstAlbumRootId, int albumId, const QStringList &dstNames)
- **operator const CollectionScannerHints::Album &** () const
- bool **operator==** (const CollectionScannerHints::Album &dst) const
- QList< qlonglong > **srcIds** () const

### Protected Attributes

- CollectionScannerHints::Album **m\_dst**
- QStringList **m\_dstNames**
- QList< qlonglong > **m\_srcIds**

## 6.884.1 Constructor & Destructor Documentation

### 6.884.1.1 ItemCopyMoveHint()

```
Digikam::ItemCopyMoveHint::ItemCopyMoveHint ( ) [default]
```

An [ItemCopyMoveHint](#) describes a list of existing items that will be copied, moved or renamed to an album given by album root id and album id. In the new album, the items will have the filenames given in dstNames.

## 6.885 Digikam::ItemCopyright Class Reference

### Public Types

- enum [ReplaceMode](#) { [ReplaceAllEntries](#) , [ReplaceLanguageEntry](#) , [AddEntryToExisting](#) }

### Public Member Functions

- [MetaEngine::AltLangMap](#) **allCopyrightNotices** ()
- [MetaEngine::AltLangMap](#) **allRightsUsageTerms** ()
- [QStringList](#) **author** () const
- [QString](#) **authorsPosition** () const
- [QStringList](#) **byLine** () const
- [QString](#) **byLineTitle** () const
- [IptcCoreContactInfo](#) **contactInfo** ()
- [QString](#) **copyrightNotice** (const [QString](#) &languageCode=[QString](#)())
- [QStringList](#) **creator** () const
- [QString](#) **creatorJobTitle** () const
- [QString](#) **credit** () const
- void **fillTemplate** ([Template](#) &t)
- [QString](#) **instructions** ()
- [ItemCopyright](#) ()=default
- **ItemCopyright** (const [ItemCopyright](#) &other)
- **ItemCopyright** (qulonglong imageid)
- [ItemCopyright](#) & **operator=** (const [ItemCopyright](#) &other)
- [QString](#) **provider** () const
- void **removeAll** ()
- void **removeContactInfo** ()
- void **removeCopyrightNotices** ()
- void **removeCreatorJobTitle** ()
- void **removeCreators** ()
- void **removeInstructions** ()
- void **removeProvider** ()
- void **removeRightsUsageTerms** ()
- void **removeSource** ()
- void **replaceFrom** (const [ItemCopyright](#) &source)
- [QString](#) **rights** (const [QString](#) &languageCode=[QString](#)())
- [QString](#) **rightsUsageTerms** (const [QString](#) &languageCode=[QString](#)())
- void **setAuthor** (const [QString](#) &author, [ReplaceMode](#) mode=[ReplaceAllEntries](#))
- void **setAuthorsPosition** (const [QString](#) &position)
- void **setByLine** (const [QString](#) &byline, [ReplaceMode](#) mode=[ReplaceAllEntries](#))
- void **setByLineTitle** (const [QString](#) &title)

- void **setContactInfo** (const [IptcCoreContactInfo](#) &info)
- void **setCopyrightNotice** (const QString &notice, const QString &languageCode=QString(), [ReplaceMode](#) mode=[ReplaceLanguageEntry](#))
- void **setCreator** (const QString &creator, [ReplaceMode](#) mode=[ReplaceAllEntries](#))
- void **setCreatorJobTitle** (const QString &title)
- void **setCredit** (const QString &credit)
- void **setFromTemplate** (const [Template](#) &t)
- void **setInstructions** (const QString &instructions)
- void **setProvider** (const QString &provider)
- void **setRights** (const QString &notice, const QString &languageCode=QString(), [ReplaceMode](#) mode=[ReplaceLanguageEntry](#))
- void **setRightsUsageTerms** (const QString &term, const QString &languageCode=QString(), [ReplaceMode](#) mode=[ReplaceLanguageEntry](#))
- void **setSource** (const QString &source)
- QString **source** ()

### Protected Member Functions

- [CopyrightInfo](#) **copyrightInfo** (const QString &property) const
- QList< [CopyrightInfo](#) > **copyrightInfos** (const QString &property) const
- int **languageMatch** (const QList< [CopyrightInfo](#) > &infos, const QString &languageCode) const
- [MetaEngine::AltLangMap](#) **readLanguageProperties** (const QString &property)
- QString **readLanguageProperty** (const QString &property, const QString &languageCode)
- QString **readSimpleProperty** (const QString &property) const
- void **removeLanguageProperty** (const QString &property, const QString &languageCode)
- void **removeProperties** (const QString &property)
- void **setLanguageProperty** (const QString &property, const QString &value, const QString &languageCode, [ReplaceMode](#) mode)
- void **setSimpleProperty** (const QString &property, const QString &value)

### Protected Attributes

- ItemCopyrightCache \* **m\_cache** = nullptr
- qulonglong **m\_id** = 0

### Friends

- class [ItemCopyrightCache](#)

## 6.885.1 Member Enumeration Documentation

### 6.885.1.1 ReplaceMode

```
enum Digikam::ItemCopyright::ReplaceMode
```

#### Enumerator

<a href="#">ReplaceAllEntries</a>	Remove entries for all languages and add one new entry.
<a href="#">ReplaceLanguageEntry</a>	Only replace the entry with the given language.
<a href="#">AddEntryToExisting</a>	No constraints on adding the entry.

## 6.885.2 Constructor & Destructor Documentation

### 6.885.2.1 ItemCopyright()

```
Digikam::ItemCopyright::ItemCopyright ( ) [default]
```

Create a null [ItemCopyright](#) object

## 6.885.3 Member Function Documentation

### 6.885.3.1 contactInfo()

```
IptcCoreContactInfo Digikam::ItemCopyright::contactInfo ( )
```

Returns the creator's contact info. This is `Iptc4xmpCore:CreatorContactInfo` in XMP. The creator's contact information provides all necessary information to get in contact with the creator of this news object and comprises a set of sub-properties for proper addressing.

### 6.885.3.2 copyrightNotice()

```
QString Digikam::ItemCopyright::copyrightNotice (
    const QString & languageCode = QString() )
```

Returns the copyright notice. This is Photoshop Copyright Notice. This is IPTC Copyright Notice. This is DC Rights. This is `dc:rights` in XMP. Contains any necessary copyright notice for claiming the intellectual property for this news object and should identify the current owner of the copyright for the news object. Other entities like the creator of the news object may be added. Notes on usage rights should be provided in Rights usage terms. Note on language matching: You can specify a language code. If the requested language is not available, the entry with default language code is returned. If a default-language entry is not available, the first entry is returned. If you pass a null string as `languageCode`, the local language is returned.

### 6.885.3.3 creator()

```
QStringList Digikam::ItemCopyright::creator ( ) const
```

Returns the author/creator/byline. This is Photoshop Author. This is IPTC By-line. This is DC creator. This is `dc:creator` in XMP. Contains preferably the name of the person who created the content of this news object, a photographer for photos, a graphic artist for graphics, or a writer for textual news. If it is not appropriate to add the name of a person the name of a company or organization could be applied as well. Aligning with IIM notions IPTC Core intends to have only one creator for this news object despite the underlying XMP property `dc:creator` allows for more than one item to be included. If there are more than one item in this array the first one should be considered as the IPTC Core Creator value.

### 6.885.3.4 creatorJobTitle()

```
QString Digikam::ItemCopyright::creatorJobTitle ( ) const
```

Returns the creator's job title. This is Photoshop AuthorsPosition. This is IPTC By-line Title. This is `photoshop:AuthorsPosition` in XMP. Contains the job title of the person who created the content of this news object. As this is sort of a qualifier the Creator element has to be filled in as mandatory prerequisite for using Creator's Jobtitle.

### 6.885.3.5 fillTemplate()

```
void Digikam::ItemCopyright::fillTemplate (
    Template & t )
```

Fills the information fields in template concerning copyright info (note there are other fields in the a [Template](#). There will not be touched)

### 6.885.3.6 instructions()

```
QString Digikam::ItemCopyright::instructions ( )
```

Returns the instructions. This is Photoshop Instructions. This is IPTC Special Instruction. This is photoshop:Instructions in XMP. Any of a number of instructions from the provider or creator to the receiver of the news object which might include any of the following: embargoes (NewsMagazines OUT) and other restrictions not covered by the Rights Usage Terms field; information regarding the original means of capture (scanning notes, colorspace info) or other specific text information that the user may need for accurate reproduction; additional permissions or credits required when publishing.

### 6.885.3.7 provider()

```
QString Digikam::ItemCopyright::provider ( ) const
```

Returns the credit/provider. This is Photoshop Credit. This is IPTC Credit. This is photoshop:Credit in XMP Identifies the provider of the news object, who is not necessarily the owner/creator.

### 6.885.3.8 removeAll()

```
void Digikam::ItemCopyright::removeAll ( )
```

Calls all remove...() methods in this class

### 6.885.3.9 replaceFrom()

```
void Digikam::ItemCopyright::replaceFrom (
    const ItemCopyright & source )
```

Removes all entries and replaces them with the entries from source.

### 6.885.3.10 rightsUsageTerms()

```
QString Digikam::ItemCopyright::rightsUsageTerms (
    const QString & languageCode = QString() )
```

Returns the right usage terms. This has no equivalent in Photoshop, IPTC, or DC. This is xmpRights:UsageTerms in XMP. Language matching is done as with [copyrightNotice\(\)](#). Free text instructions on how this news object can be legally used.

### 6.885.3.11 setCopyrightNotice()

```
void Digikam::ItemCopyright::setCopyrightNotice (
    const QString & notice,
    const QString & languageCode = QString(),
    ReplaceMode mode = ReplaceLanguageEntry )
```

Sets the copyright notice. If you supply a null QString as language code, this is regarded as an entry for the default language ("x-default"). The ReplaceMode determines how existing entries are handled.

### 6.885.3.12 setCreator()

```
void Digikam::ItemCopyright::setCreator (
    const QString & creator,
    ReplaceMode mode = ReplaceAllEntries )
```

Sets the creator. If you want to specify only one creator, set the replace mode to ReplaceAllEntries. If you want to add it to a list of existing entries, pass AddEntryToExisting. You shall not use ReplaceLanguageEntry for this method, creators have no language associated.

### 6.885.3.13 setFromTemplate()

```
void Digikam::ItemCopyright::setFromTemplate (
    const Template & t )
```

Sets all database copyright fields from the template. This does not clear any fields before.

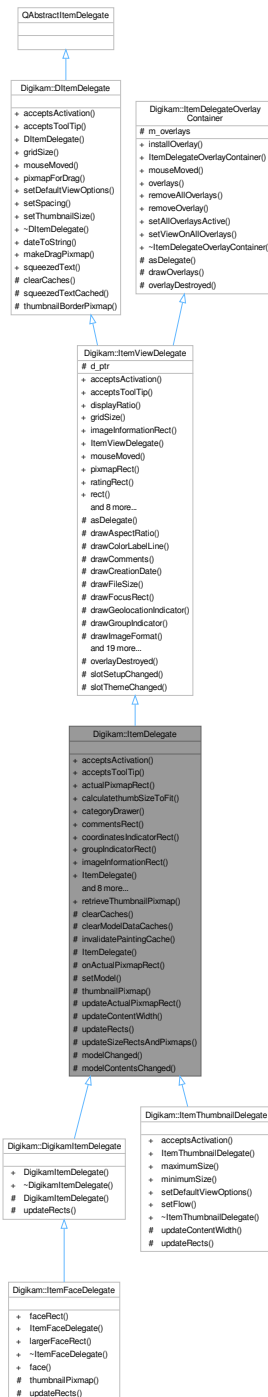
### 6.885.3.14 source()

```
QString Digikam::ItemCopyright::source ( )
```

Returns the source. This is Photoshop Source. This is IPTC Source. This is photoshop::Source in XMP. Identifies the original owner of the copyright for the intellectual content of the news object. This could be an agency, a member of an agency or an individual. Source could be different from Creator and from the entities in the CopyrightNotice. As the original owner can not change the content of this property should never be changed or deleted after the information is entered following the news object's initial creation.

## 6.886 Digikam::ItemDelegate Class Reference

Inheritance diagram for Digikam::ItemDelegate:



### Classes

- class [ItemDelegatePrivate](#)



**Public Member Functions**

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect [actualPixmapRect](#) (const QModelIndex &index) const
- int [calculatethumbSizeToFit](#) (int ws)
- [ItemCategoryDrawer](#) \* [categoryDrawer](#) () const
- QRect [commentsRect](#) () const
- QRect [coordinatesIndicatorRect](#) () const
- QRect [groupIndicatorRect](#) () const
- QRect [imageInformationRect](#) () const override
- [ItemDelegate](#) (QWidget \*const parent)
- void [paint](#) (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setSpacing](#) (int spacing) override
- void [setView](#) ([ItemCategorizedView](#) \*view)
- QRect [tagsRect](#) () const

**Public Member Functions inherited from [Digikam::ItemViewDelegate](#)**

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double [displayRatio](#) () const
- QSize [gridSize](#) () const override
- [ItemViewDelegate](#) (QWidget \*const parent)
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect [ratingRect](#) () const
- QRect [rect](#) () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setRatingEdited](#) (const QModelIndex &index)
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int [spacing](#) () const
- [ThumbnailSize](#) [thumbnailSize](#) () const

**Public Member Functions inherited from [Digikam::DItemDelegate](#)**

- [DItemDelegate](#) (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Static Public Member Functions

- static QPixmap [retrieveThumbnailPixmap](#) (const QModelIndex &index, int thumbnailSize)

## Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

## Protected Slots

- void **modelChanged** ()
- void **modelContentsChanged** ()

## Protected Slots inherited from [Digikam::ItemViewDelegate](#)

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

## Protected Member Functions

- void [clearCaches](#) () override
- virtual void [clearModelDataCaches](#) ()
- void [invalidatePaintingCache](#) () override
- **ItemDelegate** ([ItemDelegate::ItemDelegatePrivate](#) &dd, QWidget \*const parent)
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- virtual void [updateContentWidth](#) ()
- virtual void [updateRects](#) ()=0
- void [updateSizeRectsAndPixmaps](#) () override

## Protected Member Functions inherited from [Digikam::ItemViewDelegate](#)

- `QAbstractItemDelegate * asDelegate ()` override  
*Returns the delegate, typically, the derived class.*
- void **`drawAspectRatio`** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **`drawColorLabelLine`** (QPainter \*p, const QRect &pixRect, int colorId) const
- void **`drawComments`** (QPainter \*p, const QRect &commentsRect, const QString &comments) const
- void **`drawCreationDate`** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **`drawFileSize`** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **`drawFocusRect`** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **`drawGeolocationIndicator`** (QPainter \*p, const QRect &r) const
- void **`drawGroupIndicator`** (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
- void **`drawImageFormat`** (QPainter \*p, const QRect &r, const QString &f, bool drawTop) const
- void **`drawImageSize`** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **`drawModificationDate`** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **`drawMouseOverRect`** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **`drawName`** (QPainter \*p, const QRect &nameRect, const QString &name) const
- void **`drawPanelSidelcon`** (QPainter \*p, bool left, bool right) const
- void **`drawPickLabelIcon`** (QPainter \*p, const QRect &r, int pickLabel) const
- void **`drawRating`** (QPainter \*p, const QModelIndex &index, const QRect &[ratingRect](#), int rating, bool is← Selected) const
- void **`drawSpecialInfo`** (QPainter \*p, const QRect &r, const QString &text) const
- void **`drawTags`** (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const
- QRect **`drawThumbnail`** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail, bool isGrouped) const
- void **`drawTitle`** (QPainter \*p, const QRect &titleRect, const QString &title) const
- `ItemViewDelegate (ItemViewDelegatePrivate &dd, QWidget *const parent)`
- void **`prepareBackground`** ()
- void **`prepareFonts`** ()
- void **`prepareMetrics`** (int maxWidth)
- void **`prepareRatingPixmap`** (bool composeOverBackground=true)
- QPixmap **`ratingPixmap`** (int rating, bool selected) const

## Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString **`squeezedTextCached`** (QPainter \*const p, int width, const QString &text) const
- QPixmap **`thumbnailBorderPixmap`** (const QSize &pixSize, bool isGrouped=false) const

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **`drawOverlays`** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **`overlayDestroyed`** (QObject \*o)  
*Declare as slot in the derived class calling this method.*

## Additional Inherited Members

## Signals inherited from [Digikam::ItemViewDelegate](#)

- void **`hideNotification`** ()
- void **`requestNotification`** (const QModelIndex &index, const QString &message)

## Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Protected Attributes inherited from [Digikam::ItemViewDelegate](#)

- [ItemViewDelegatePrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > **m\_overlays**

## 6.886.1 Member Function Documentation

### 6.886.1.1 `acceptsActivation()`

```
bool Digikam::ItemDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect = nullptr ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

### 6.886.1.2 `acceptsToolTip()`

```
bool Digikam::ItemDelegate::acceptsToolTip (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * tooltipRect = nullptr ) const [override], [virtual]
```

These methods take four parameters: The position on viewport, the rect on viewport, the index, and optionally a parameter into which, if the return value is true, a rectangle can be written for which the return value will be true as well.

Implements [Digikam::DItemDelegate](#).

### 6.886.1.3 `clearCaches()`

```
void Digikam::ItemDelegate::clearCaches ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::DItemDelegate](#).

#### 6.886.1.4 clearModelDataCaches()

```
void Digikam::ItemDelegate::clearModelDataCaches ( ) [protected], [virtual]
```

Reimplement to clear caches based on model indexes (hash on row number etc.) Change signals are listened to this is called whenever such properties become invalid.

#### 6.886.1.5 imageInformationRect()

```
QRect Digikam::ItemDelegate::imageInformationRect ( ) const [override], [virtual]
```

Returns the area where the image information is drawn, or null if empty / not supported. The image information is textual or graphical information, but not the pixmap. The [ratingRect\(\)](#) will e.g. typically be contained in this area.

Reimplemented from [Digikam::ItemViewDelegate](#).

#### 6.886.1.6 invalidatePaintingCache()

```
void Digikam::ItemDelegate::invalidatePaintingCache ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewDelegate](#).

#### 6.886.1.7 pixmapForDrag()

```
QPixmap Digikam::ItemDelegate::pixmapForDrag (
    const QStyleOptionViewItem & option,
    const QList< QModelIndex > & indexes ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

#### 6.886.1.8 pixmapRect()

```
QRect Digikam::ItemDelegate::pixmapRect ( ) const [override], [virtual]
```

Returns the area where the pixmap is drawn, or null if not supported.

Reimplemented from [Digikam::ItemViewDelegate](#).

#### 6.886.1.9 retrieveThumbnailPixmap()

```
QPixmap Digikam::ItemDelegate::retrieveThumbnailPixmap (
    const QModelIndex & index,
    int thumbnailSize ) [static]
```

Retrieve the thumbnail pixmap in given size for the [ItemModel::ThumbnailRole](#) for the given index from the given index, which must adhere to [ItemThumbnailModel](#) semantics.

#### 6.886.1.10 `setDefaultViewOptions()`

```
void Digikam::ItemDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. `option.rect` shall be the viewport rectangle. Call on resize, font change.

Implements [Digikam::DItemDelegate](#).

Reimplemented in [Digikam::ItemThumbnailDelegate](#).

#### 6.886.1.11 `setSpacing()`

```
void Digikam::ItemDelegate::setSpacing (
    int spacing ) [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

#### 6.886.1.12 `updateContentWidth()`

```
void Digikam::ItemDelegate::updateContentWidth ( ) [protected], [virtual]
```

Reimplement this to set `contentWidth`. This is the maximum width of all content rectangles, typically excluding margins on both sides.

Reimplemented in [Digikam::ItemThumbnailDelegate](#).

#### 6.886.1.13 `updateRects()`

```
virtual void Digikam::ItemDelegate::updateRects ( ) [protected], [pure virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The `paint()` method operates depending on these rects.

Implemented in [Digikam::DigikamItemDelegate](#), [Digikam::ItemFaceDelegate](#), and [Digikam::ItemThumbnailDelegate](#).

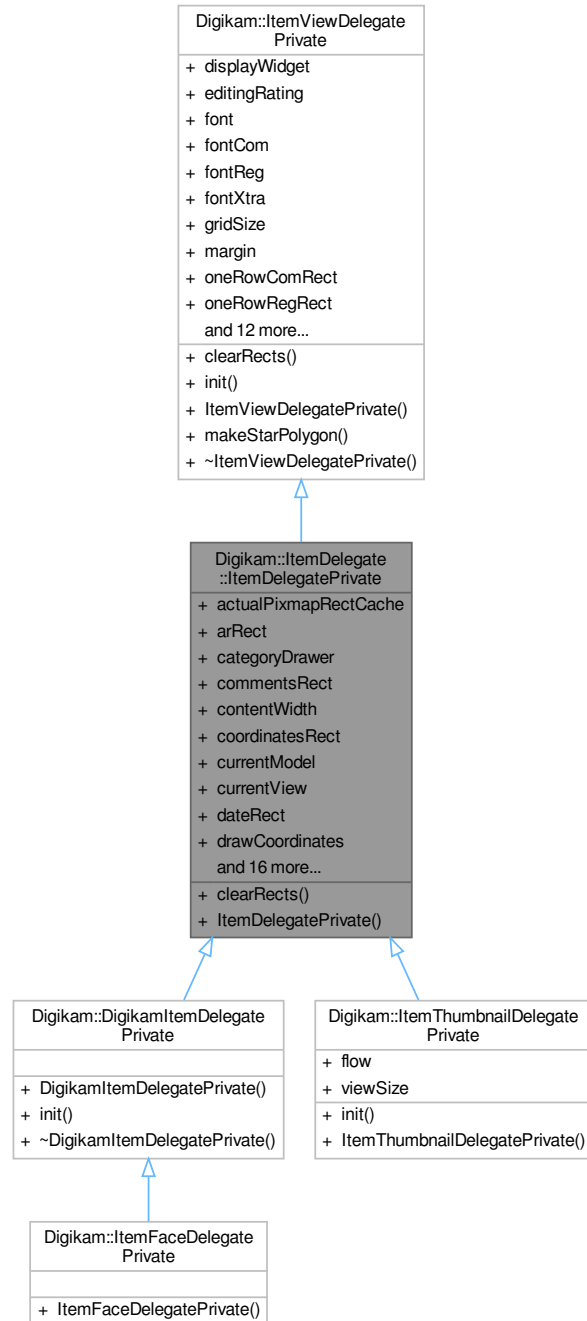
#### 6.886.1.14 `updateSizeRectsAndPixmap()`

```
void Digikam::ItemDelegate::updateSizeRectsAndPixmap ( ) [override], [protected], [virtual]
```

Implements [Digikam::ItemViewDelegate](#).

## 6.887 Digikam::ItemDelegate::ItemDelegatePrivate Class Reference

Inheritance diagram for Digikam::ItemDelegate::ItemDelegatePrivate:



### Public Member Functions

- void `clearRects` () override

## Public Member Functions inherited from [Digikam::ItemViewDelegatePrivate](#)

- void **init** ([ItemViewDelegate](#) \*const \_q, QWidget \*const \_widget)
- void **makeStarPolygon** ()

## Public Attributes

- QCache< int, QRect > **actualPixmapRectCache**
- QRect **arRect**
- [ItemCategoryDrawer](#) \* **categoryDrawer**
- QRect **commentsRect**
- int **contentWidth**
- QRect **coordinatesRect**
- QAbstractItemModel \* **currentModel**
- [ItemCategorizedView](#) \* **currentView**
- QRect **dateRect**
- bool **drawCoordinates**
- bool **drawFocusFrame**
- bool **drawImageFormat**
- bool **drawImageFormatTop**
- bool **drawMouseOverFrame**
- QRect **groupRect**
- QRect **imageInformationRect**
- QRect **modDateRect**
- QRect **nameRect**
- QRect **pickLabelRect**
- QRect **pixmapRect**
- bool **ratingOverThumbnail**
- QRect **resolutionRect**
- QRect **sizeRect**
- QRect **specialInfoRect**
- QRect **tagRect**
- QRect **titleRect**

## Public Attributes inherited from [Digikam::ItemViewDelegatePrivate](#)

- QWidget \* **displayWidget** = nullptr
- QPersistentModelIndex **editingRating**
- QFont **font**
- QFont **fontCom**
- QFont **fontReg**
- QFont **fontXtra**
- QSize **gridSize**
- int **margin** = 5
- QRect **oneRowComRect**
- QRect **oneRowRegRect**
- QRect **oneRowXtraRect**
- [ItemViewDelegate](#) \* **q** = nullptr
- int **radius** = 3
- *constant values for drawing*
- QVector< QPixmap > **ratingPixmaps** = QVector< QPixmap >(10)
- QRect **ratingRect**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- QPolygon **starPolygon**
- QSize **starPolygonSize**
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)





## Protected Slots

- virtual void [visualChange](#) ()

## Protected Member Functions

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

## Protected Attributes

- QAbstractItemDelegate \* **m\_delegate** = nullptr
- QAbstractItemView \* **m\_view** = nullptr

## 6.888.1 Member Function Documentation

### 6.888.1.1 [affectsMultiple\(\)](#)

```
bool Digikam::ItemDelegateOverlay::affectsMultiple (
    const QModelIndex & index ) const [protected]
```

For the context that an overlay can affect multiple items: Assuming the currently overlaid index is given. Will an operation affect only the single item, or multiple? If multiple, retrieve the affected selection.

### 6.888.1.2 [mouseMoved\(\)](#)

```
void Digikam::ItemDelegateOverlay::mouseMoved (
    QMouseEvent * e,
    const QRect & visualRect,
    const QModelIndex & index ) [virtual]
```

Only these two methods are implemented as virtual methods. For all other events, connect to the view's signals. There are a few signals specifically for overlays and all QAbstractItemView standard signals.

### 6.888.1.3 [setActive\(\)](#)

```
void Digikam::ItemDelegateOverlay::setActive (
    bool active ) [virtual]
```

Called when the overlay was installed and shall begin working, and before it is removed and shall stop. [Setup](#) your connections to view and delegate here. You will be disconnected automatically on removal.

Reimplemented in [Digikam::FaceRejectionOverlay](#), [Digikam::ItemCoordinatesOverlay](#), [Digikam::ItemFullScreenOverlay](#), [Digikam::ItemRotateOverlay](#), [Digikam::ItemSelectionOverlay](#), [Digikam::ShowHideVersionsOverlay](#), [Digikam::ActionVersionsOverlay](#), [Digikam::AbstractWidgetDelegateOverlay](#), [Digikam::HoverButtonDelegateOverlay](#), [Digikam::PersistentWidgetDelegateOverlay](#), [ShowFoto::ShowfotoCoordinatesOverlay](#), [Digikam::ImportCoordinatesOverlay](#), [Digikam::ImportLockOverlay](#), [Digikam::ImportDownloadOverlay](#), [Digikam::ImportRotateOverlay](#), [Digikam::AssignNameOverlay](#), [Digikam::GroupIndicatorOverlay](#), [Digikam::ItemRatingOverlay](#), [Digikam::TagsLineEditOverlay](#), and [Digikam::ImportRatingOverlay](#).

#### 6.888.1.4 viewHasMultiSelection()

```
bool Digikam::ItemDelegateOverlay::viewHasMultiSelection ( ) const [protected]
```

Utility method

#### 6.888.1.5 visualChange

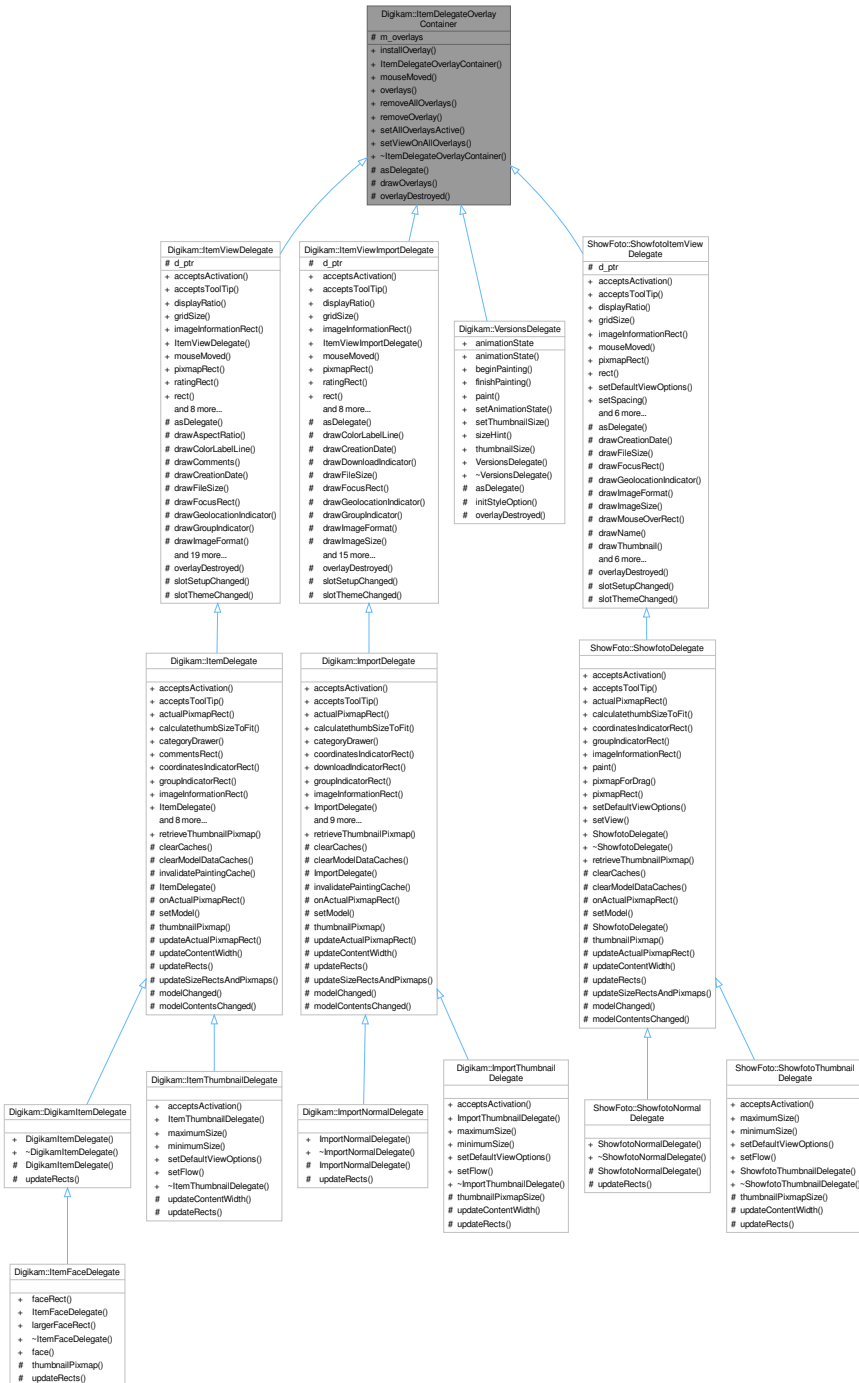
```
void Digikam::ItemDelegateOverlay::visualChange ( ) [protected], [virtual], [slot]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented in [Digikam::AssignNameOverlay](#), [Digikam::GroupIndicatorOverlay](#), [Digikam::ItemCoordinatesOverlay](#), [Digikam::ItemRatingOverlay](#), [Digikam::TagsLineEditOverlay](#), [Digikam::HoverButtonDelegateOverlay](#), [ShowFoto::ShowfotoCoordinatesOverlay](#), [Digikam::ImportCoordinatesOverlay](#), [Digikam::ImportLockOverlay](#), [Digikam::ImportDownloadOverlay](#), and [Digikam::ImportRatingOverlay](#).

## 6.889 Digikam::ItemDelegateOverlayContainer Class Reference

Inheritance diagram for Digikam::ItemDelegateOverlayContainer:



### Public Member Functions

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)

- `QList< ItemDelegateOverlay * > overlays ()` const
- `void removeAllOverlays ()`
- `void removeOverlay (ItemDelegateOverlay *overlay)`
- `void setAllOverlaysActive (bool active)`
- `void setViewOnAllOverlays (QAbstractItemView *view)`

### Protected Member Functions

- virtual `QAbstractItemDelegate * asDelegate ()=0`  
*Returns the delegate, typically, the derived class.*
- virtual void **drawOverlays** (`QPainter *p`, const `QStyleOptionViewItem &option`, const `QModelIndex &index`) const
- virtual void **overlayDestroyed** (`QObject *o`)  
*Declare as slot in the derived class calling this method.*

### Protected Attributes

- `QList< ItemDelegateOverlay * > m_overlays`

## 6.889.1 Constructor & Destructor Documentation

### 6.889.1.1 ItemDelegateOverlayContainer()

```
Digikam::ItemDelegateOverlayContainer::ItemDelegateOverlayContainer ( ) [default]
```

This is a sample implementation for delegate management methods, to be inherited by a delegate. Does not inherit `QObject`, the delegate already does.

## 6.889.2 Member Function Documentation

### 6.889.2.1 asDelegate()

```
virtual QAbstractItemDelegate * Digikam::ItemDelegateOverlayContainer::asDelegate ( ) [protected],  
[pure virtual]
```

Implemented in [Digikam::VersionsDelegate](#), [Digikam::ItemViewDelegate](#), [ShowFoto::ShowfotoItemViewDelegate](#), and [Digikam::ItemViewImportDelegate](#).

## 6.890 Digikam::ItemDescEditTab Class Reference

Inheritance diagram for Digikam::ItemDescEditTab:



### Classes

- class [Private](#)

## Public Types

- enum **DescEditTab** { **DESCRIPTIONS** = 0 , **TAGS** , **INFOS** }

## Signals

- void **signalAskToApplyChanges** (const QList< [ItemInfo](#) > &infos, [DisjointMetadata](#) \*hub)
- void **signalNextItem** ()
- void **signalPrevItem** ()
- void **signalProgressFinished** ()
- void **signalProgressMessageChanged** (const QString &actionDescription)
- void **signalProgressValueChanged** (float percent)
- void **signalRightSideBarBusy** (bool busy)

## Public Member Functions

- int **currentTab** () const
- bool **isModified** () const
- **ItemDescEditTab** (QWidget \*const parent)
- void **readSettings** (KConfigGroup &group)
- void **setCurrentTab** (int)
- void **setItem** (const [ItemInfo](#) &info=[ItemInfo](#)())
- void **setItems** (const [ItemInfoList](#) &infos)
- void **writeSettings** (KConfigGroup &group)

## Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentMargins** (const QMargins &margins)
- void **setContentMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Protected Member Functions

- bool **eventFilter** (QObject \*o, QEvent \*e) override

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)
  
- class **Private**
  
- void **assignColorLabel** (int colorId)
  - Description view methods (itemdesceditab\_descview.cpp)*
- void **assignPickLabel** (int pickId)
- void **assignRating** (int rating)
- void **setFocusToTitlesEdit** ()
- void **setFocusToCommentsEdit** ()
- void **replaceColorLabel** (int colorId)
- void **replacePickLabel** (int pickId)
- void **replaceRating** (int rating)
  
- void **populateTags** ()
  - Tags view methods (itemdesceditab\_tagsview.cpp)*
- void **setFocusToTagsView** ()
- void **setFocusToNewTagEdit** ()
- void **activateAssignedTagsButton** ()
- [AddTagsLineEdit](#) \* **getNewTagEdit** () const
- void **signalTagFilterMatch** (bool)

## 6.891 Digikam::ItemDescEditTab::Private Class Reference

### Public Member Functions

- void **initProgressIndicator** ()
- void **metadataChange** (qulonglong imageId)
- **Private** ([ItemDescEditTab](#) \*const tab)
- void **reset** ()
- void **resetMetadataChangeInfo** ()
- void **setFocusToLastSelectedWidget** ()
- void **setInfos** (const [ItemInfoList](#) &infos)
- void **setMetadataWidgetStatus** (int status, QWidget \*const widget)
- void **setupConnections** ()
- void **setupEventFilters** ()
- bool **singleSelection** () const

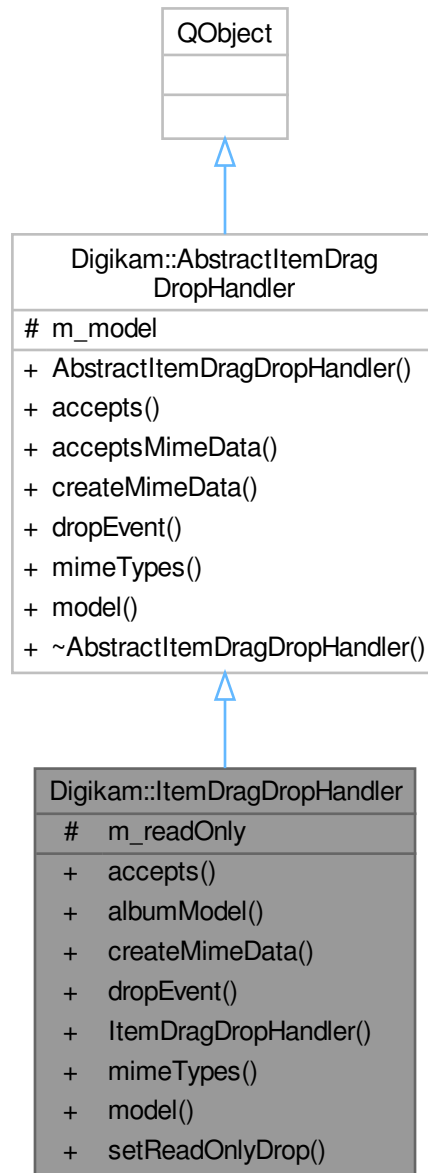


## Public Attributes

- QPushButton \* **applyBtn** = nullptr
- QPushButton \* **applyToAllVersionsButton** = nullptr
- QToolButton \* **assignedTagsBtn** = nullptr
- CaptionEdit \* **captionsEdit** = nullptr
- ColorLabelSelector \* **colorLabelSelector** = nullptr
- ItemInfoList **currInfos**
- QDateTimeEdit \* **dateTimeEdit** = nullptr
- DisjointMetadata \* **hub** = nullptr
- bool **ignoreItemAttributesWatch** = false
- bool **ignoreTagChanges** = false
- QWidget \* **lastSelectedWidget** = nullptr
- QList< int > **metadataChangeIds**
- QTimer \* **metadataChangeTimer** = nullptr
- bool **modified** = false
- QPushButton \* **moreButton** = nullptr
- QMenu \* **moreMenu** = nullptr
- AddTagsLineEdit \* **newTagEdit** = nullptr
- QPushButton \* **openTagMgr** = nullptr
- PickLabelSelector \* **pickLabelSelector** = nullptr
- RatingWidget \* **ratingWidget** = nullptr
- QToolButton \* **recentTagsBtn** = nullptr
- QToolButton \* **revertBtn** = nullptr
- int **spacing** = 0
- QTabWidget \* **tabWidget** = nullptr
- TagCheckView \* **tagCheckView** = nullptr
- TagModel \* **tagModel** = nullptr
- SearchTextBarDb \* **tagsSearchBar** = nullptr
- TemplateSelector \* **templateSelector** = nullptr
- TemplateViewer \* **templateViewer** = nullptr
- AltLangStrEdit \* **titleEdit** = nullptr
- bool **toggleTagsSearchSettings** = false

## 6.892 Digikam::ItemDragDropHandler Class Reference

Inheritance diagram for Digikam::ItemDragDropHandler:



### Signals

- void **addToGroup** (const [ItemInfo](#) &pick, const QList< [ItemInfo](#) > &infos)
- void **assignTags** (const QList< [ItemInfo](#) > &list, const QList< int > &tagIDs)
- void **dragDropSort** (const [ItemInfo](#) &pick, const QList< [ItemInfo](#) > &infos)
- void **itemInfosDropped** (const QList< [ItemInfo](#) > &infos)
- void **urlsDropped** (const QList< QUrl > &urls)

## Public Member Functions

- Qt::DropAction [accepts](#) (const QDropEvent \*e, const QModelIndex &dropIndex) override
- [ItemAlbumModel](#) \* [albumModel](#) () const
- QMimeData \* [createMimeData](#) (const QList< QModelIndex > &) override
- bool [dropEvent](#) (QAbstractItemView \*view, const QDropEvent \*e, const QModelIndex &droppedOn) override
- [ItemDragDropHandler](#) ([ItemModel](#) \*const model)
- QStringList [mimeTypes](#) () const override
- [ItemModel](#) \* [model](#) () const
- void [setReadOnlyDrop](#) (bool readOnly)

## Public Member Functions inherited from [Digikam::AbstractItemDragDropHandler](#)

- [AbstractItemDragDropHandler](#) (QAbstractItemModel \*const model)
- virtual bool [acceptsMimeData](#) (const QMimeData \*data)
- QAbstractItemModel \* [model](#) () const

## Protected Attributes

- bool [m\\_readOnly](#) = false

## Protected Attributes inherited from [Digikam::AbstractItemDragDropHandler](#)

- QAbstractItemModel \* [m\\_model](#) = nullptr

## 6.892.1 Member Function Documentation

### 6.892.1.1 [accepts\(\)](#)

```
Qt::DropAction Digikam::ItemDragDropHandler::accepts (
    const QDropEvent * e,
    const QModelIndex & dropIndex ) [override], [virtual]
```

Returns if the given mime data is accepted for drop on dropIndex. Returns the proposed action, or Qt::IgnoreAction if not accepted.

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.892.1.2 [createMimeData\(\)](#)

```
QMimeData * Digikam::ItemDragDropHandler::createMimeData (
    const QList< QModelIndex > & ) [override], [virtual]
```

Create a mime data object for starting a drag from the given Albums

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.892.1.3 dropEvent()

```
bool Digikam::ItemDragDropHandler::dropEvent (
    QAbstractItemView * view,
    const QDropEvent * e,
    const QModelIndex & droppedOn ) [override], [virtual]
```

Gives the view and the occurring drop event. The index is the index where the drop was dropped on. It may be invalid (dropped on decoration, viewport) Returns true if the event is to be accepted.

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.892.1.4 mimeTypees()

```
QStringList Digikam::ItemDragDropHandler::mimeTypees ( ) const [override], [virtual]
```

Returns the supported mime types. Called by the default implementation of model's [mimeTypees\(\)](#).

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.892.1.5 setReadOnlyDrop()

```
void Digikam::ItemDragDropHandler::setReadOnlyDrop (
    bool readOnly )
```

Enables a mode in which dropping will never start an operation which copies or moves files on disk. Only the signals are emitted.

## 6.893 Digikam::ItemExtendedProperties Class Reference

### Public Member Functions

- QString [intellectualGenre](#) ()
- [ItemExtendedProperties](#) ()=default
- **ItemExtendedProperties** (qulonglong imageid)
- QString [jobId](#) ()
- [IptcCoreLocationInfo](#) [location](#) ()
- void **removeIntellectualGenre** ()
- void **removeJobId** ()
- void **removeLocation** ()
- void **removeScene** ()
- void **removeSimilarityTo** (const qulonglong imageId)
- void **removeSubjectCode** ()
- QStringList [scene](#) ()
- void **setIntellectualGenre** (const QString &[intellectualGenre](#))
- void **setJobId** (const QString &[jobId](#))
- void **setLocation** (const [IptcCoreLocationInfo](#) &[location](#))
- void **setScene** (const QStringList &[scene](#))
- void **setSimilarityTo** (const qulonglong imageId, const double value)
- void **setSubjectCode** (const QStringList &[subjectCode](#))
- double [similarityTo](#) (const qulonglong imageId)
- QStringList [subjectCode](#) ()

## Protected Member Functions

- QStringList **readFakeListProperty** (const QString &property)
- QString **readProperty** (const QString &property)
- void **removeProperty** (const QString &property)
- void **setFakeListProperty** (const QString &property, const QStringList &value)
- void **setProperty** (const QString &property, const QString &value)

## Protected Attributes

- qlonglong **m\_id** = 0

## 6.893.1 Constructor & Destructor Documentation

### 6.893.1.1 ItemExtendedProperties()

```
Digikam::ItemExtendedProperties::ItemExtendedProperties ( ) [default]
```

Create a null [ItemExtendedProperties](#) object

## 6.893.2 Member Function Documentation

### 6.893.2.1 intellectualGenre()

```
QString Digikam::ItemExtendedProperties::intellectualGenre ( )
```

Return the Intellectual Genre. This is Photoshop Object Attribute Reference. “ Describes the nature, intellectual or journalistic characteristic of a news object, not specifically its content. Note / Examples: Journalistic genres: actuality, interview, background, feature, summary, wrapup News category related genres: daybook, obituary, press release, transcript It is advised to use terms from a controlled vocabulary.”

### 6.893.2.2 jobId()

```
QString Digikam::ItemExtendedProperties::jobId ( )
```

Returns the Job ID. This is Photoshop Transmission Reference. This is IPTC Original Transmission Reference “ Number or identifier for the purpose of improved workflow handling. This ID should be added by the creator or provider for transmission and routing purposes only and should have no significance for archiving.”

### 6.893.2.3 location()

```
IptcCoreLocationInfo Digikam::ItemExtendedProperties::location ( )
```

Return the IPTC Core Location. This includes Country, Country Code, City, Location and ProvinceState. This includes IPTC Country Name, Country Code, City, SubLocation and ProvinceState.

**6.893.2.4 scene()**

```
QStringList Digikam::ItemExtendedProperties::scene ( )
```

Returns the Scene. “ Describes the scene of a photo content. Specifies one ore more terms from the IPTC ‘Scene-NewsCodes’. Each Scene is represented as a string of 6 digits in an unordered list.”

**6.893.2.5 similarityTo()**

```
double Digikam::ItemExtendedProperties::similarityTo (
    const qlonglong imageId )
```

Returns the similarity. of the image to the given image.

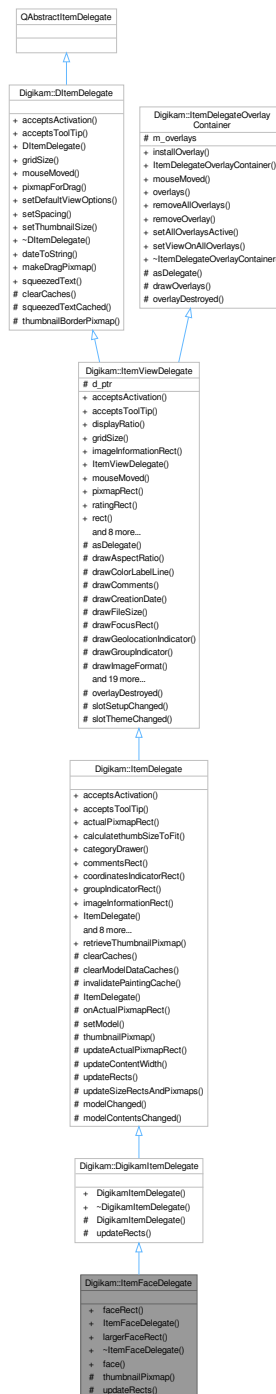
**6.893.2.6 subjectCode()**

```
QStringList Digikam::ItemExtendedProperties::subjectCode ( )
```

Returns the Subject Code. This is IPTC Subject Reference. “ Specifies one or more Subjects from the IPTC ‘Subject-NewsCodes’ taxonomy to categorize the content. Each Subject is represented as a string of 8 digits in an unordered list. Note: Only Subjects from a controlled vocabulary should be used in this metadata element, free text has to be put into the Keyword element. More about IPTC Subject-NewsCodes at [www.newscodes.org](http://www.newscodes.org).”

## 6.894 Digikam::ItemFaceDelegate Class Reference

Inheritance diagram for Digikam::ItemFaceDelegate:



### Public Member Functions

- QRect **faceRect** (const QModelIndex &index) const
- **ItemFaceDelegate** (**ItemCategorizedView** \*const parent)
- QRect **largerFaceRect** (const QModelIndex &index) const

## Public Member Functions inherited from [Digikam::DigikamItemDelegate](#)

- [DigikamItemDelegate](#) ([ItemCategorizedView](#) \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect [actualPixmapRect](#) (const QModelIndex &index) const
- int [calculatethumbSizeToFit](#) (int ws)
- [ItemCategoryDrawer](#) \* [categoryDrawer](#) () const
- QRect [commentsRect](#) () const
- QRect [coordinatesIndicatorRect](#) () const
- QRect [groupIndicatorRect](#) () const
- QRect [imageInformationRect](#) () const override
- [ItemDelegate](#) (QWidget \*const parent)
- void [paint](#) (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setSpacing](#) (int spacing) override
- void [setView](#) ([ItemCategorizedView](#) \*view)
- QRect [tagsRect](#) () const

## Public Member Functions inherited from [Digikam::ItemViewDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double [displayRatio](#) () const
- QSize [gridSize](#) () const override
- [ItemViewDelegate](#) (QWidget \*const parent)
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect [ratingRect](#) () const
- QRect [rect](#) () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setRatingEdited](#) (const QModelIndex &index)
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int [spacing](#) () const
- [ThumbnailSize](#) [thumbnailSize](#) () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- [DItemDelegate](#) (QObject \*const parent=nullptr)



## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Static Public Member Functions

- static [FaceTagsIface](#) **face** (const QModelIndex &index)

## Static Public Member Functions inherited from [Digikam::ItemDelegate](#)

- static QPixmap [retrieveThumbnailPixmap](#) (const QModelIndex &index, int thumbnailSize)

## Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

## Protected Member Functions

- QPixmap [thumbnailPixmap](#) (const QModelIndex &index) const override
- void [updateRects](#) () override

## Protected Member Functions inherited from [Digikam::DigikamItemDelegate](#)

- [DigikamItemDelegate](#) ([DigikamItemDelegatePrivate](#) &dd, [ItemCategorizedView](#) \*parent)

## Protected Member Functions inherited from [Digikam::ItemDelegate](#)

- void [clearCaches](#) () override
- virtual void [clearModelDataCaches](#) ()
- void [invalidatePaintingCache](#) () override
- [ItemDelegate](#) ([ItemDelegate::ItemDelegatePrivate](#) &dd, QWidget \*const parent)
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- virtual void [updateContentWidth](#) ()
- void [updateSizeRectsAndPixmaps](#) () override

## Protected Member Functions inherited from [Digikam::ItemViewDelegate](#)

- `QAbstractItemDelegate * asDelegate ()` override  
*Returns the delegate, typically, the derived class.*
- void **`drawAspectRatio`** (`QPainter *p`, `const QRect &dimsRect`, `const QSize &dims`) `const`
- void **`drawColorLabelLine`** (`QPainter *p`, `const QRect &pixRect`, `int colorId`) `const`
- void **`drawComments`** (`QPainter *p`, `const QRect &commentsRect`, `const QString &comments`) `const`
- void **`drawCreationDate`** (`QPainter *p`, `const QRect &dateRect`, `const QDateTime &date`) `const`
- void **`drawFileSize`** (`QPainter *p`, `const QRect &r`, `qulonglong bytes`) `const`
- void **`drawFocusRect`** (`QPainter *p`, `const QStyleOptionViewItem &option`, `bool isSelected`) `const`
- void **`drawGeolocationIndicator`** (`QPainter *p`, `const QRect &r`) `const`
- void **`drawGroupIndicator`** (`QPainter *p`, `const QRect &r`, `int numberOfGroupedImages`, `bool open`) `const`
- void **`drawImageFormat`** (`QPainter *p`, `const QRect &r`, `const QString &f`, `bool drawTop`) `const`
- void **`drawImageSize`** (`QPainter *p`, `const QRect &dimsRect`, `const QSize &dims`) `const`
- void **`drawModificationDate`** (`QPainter *p`, `const QRect &dateRect`, `const QDateTime &date`) `const`
- void **`drawMouseOverRect`** (`QPainter *p`, `const QStyleOptionViewItem &option`) `const`
- void **`drawName`** (`QPainter *p`, `const QRect &nameRect`, `const QString &name`) `const`
- void **`drawPanelSidelcon`** (`QPainter *p`, `bool left`, `bool right`) `const`
- void **`drawPickLabelIcon`** (`QPainter *p`, `const QRect &r`, `int pickLabel`) `const`
- void **`drawRating`** (`QPainter *p`, `const QModelIndex &index`, `const QRect &ratingRect`, `int rating`, `bool isSelected`) `const`
- void **`drawSpecialInfo`** (`QPainter *p`, `const QRect &r`, `const QString &text`) `const`
- void **`drawTags`** (`QPainter *p`, `const QRect &r`, `const QString &tagsString`, `bool isSelected`) `const`
- `QRect drawThumbnail` (`QPainter *p`, `const QRect &thumbRect`, `const QPixmap &background`, `const QPixmap &thumbnail`, `bool isGrouped`) `const`
- void **`drawTitle`** (`QPainter *p`, `const QRect &titleRect`, `const QString &title`) `const`
- `ItemViewDelegate` (`ItemViewDelegatePrivate` &dd, `QWidget *const parent`)
- void **`prepareBackground`** ()
- void **`prepareFonts`** ()
- void **`prepareMetrics`** (`int maxWidth`)
- void **`prepareRatingPixmap`** (`bool composeOverBackground=true`)
- `QPixmap ratingPixmap` (`int rating`, `bool selected`) `const`

## Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- `QString squeezedTextCached` (`QPainter *const p`, `int width`, `const QString &text`) `const`
- `QPixmap thumbnailBorderPixmap` (`const QSize &pixSize`, `bool isGrouped=false`) `const`

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **`drawOverlays`** (`QPainter *p`, `const QStyleOptionViewItem &option`, `const QModelIndex &index`) `const`
- virtual void **`overlayDestroyed`** (`QObject *o`)  
*Declare as slot in the derived class calling this method.*

## Additional Inherited Members

## Signals inherited from [Digikam::ItemViewDelegate](#)

- void **`hideNotification`** ()
- void **`requestNotification`** (`const QModelIndex &index`, `const QString &message`)

## Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Protected Slots inherited from [Digikam::ItemDelegate](#)

- void **modelChanged** ()
- void **modelContentsChanged** ()

## Protected Slots inherited from [Digikam::ItemViewDelegate](#)

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

## Protected Attributes inherited from [Digikam::ItemViewDelegate](#)

- [ItemViewDelegatePrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > **m\_overlays**

## 6.894.1 Member Function Documentation

### 6.894.1.1 thumbnailPixmap()

```
QPixmap Digikam::ItemFaceDelegate::thumbnailPixmap (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemDelegate](#).

### 6.894.1.2 updateRects()

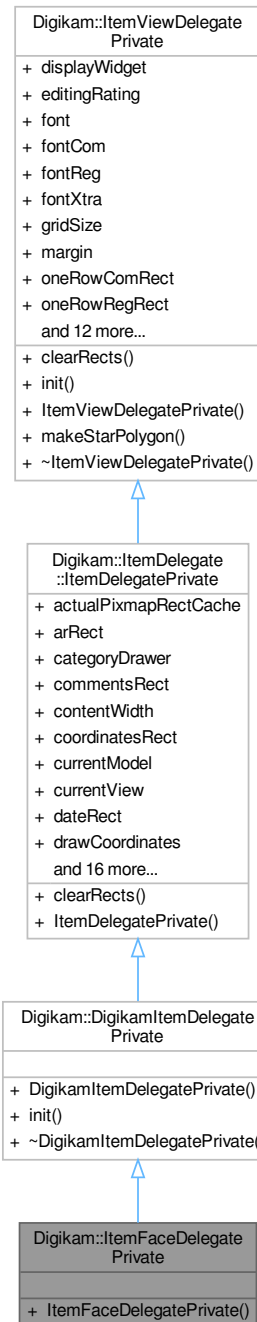
```
void Digikam::ItemFaceDelegate::updateRects ( ) [override], [protected], [virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The paint() method operates depending on these rects.

Reimplemented from [Digikam::DigikamItemDelegate](#).

## 6.895 Digikam::ItemFaceDelegatePrivate Class Reference

Inheritance diagram for Digikam::ItemFaceDelegatePrivate:



### Additional Inherited Members

### Public Member Functions inherited from [Digikam::DigikamItemDelegatePrivate](#)

- void `init` ([DigikamItemDelegate](#) \*const q, [ItemCategorizedView](#) \*const parent)

## Public Member Functions inherited from Digikam::ItemDelegate::ItemDelegatePrivate

- void `clearRects` () override

## Public Member Functions inherited from Digikam::ItemViewDelegatePrivate

- void `init` (ItemViewDelegate \*const \_q, QWidget \*const \_widget)
- void `makeStarPolygon` ()

## Public Attributes inherited from Digikam::ItemDelegate::ItemDelegatePrivate

- QCache< int, QRect > `actualPixmapRectCache`
- QRect `arRect`
- ItemCategoryDrawer \* `categoryDrawer`
- QRect `commentsRect`
- int `contentWidth`
- QRect `coordinatesRect`
- QAbstractItemModel \* `currentModel`
- ItemCategorizedView \* `currentView`
- QRect `dateRect`
- bool `drawCoordinates`
- bool `drawFocusFrame`
- bool `drawImageFormat`
- bool `drawImageFormatTop`
- bool `drawMouseOverFrame`
- QRect `groupRect`
- QRect `imageInformationRect`
- QRect `modDateRect`
- QRect `nameRect`
- QRect `pickLabelRect`
- QRect `pixmapRect`
- bool `ratingOverThumbnail`
- QRect `resolutionRect`
- QRect `sizeRect`
- QRect `specialInfoRect`
- QRect `tagRect`
- QRect `titleRect`

## Public Attributes inherited from Digikam::ItemViewDelegatePrivate

- QWidget \* `displayWidget` = nullptr
- QPersistentModelIndex `editingRating`
- QFont `font`
- QFont `fontCom`
- QFont `fontReg`
- QFont `fontXtra`
- QSize `gridSize`
- int `margin` = 5
- QRect `oneRowComRect`
- QRect `oneRowRegRect`
- QRect `oneRowXtraRect`
- ItemViewDelegate \* `q` = nullptr

- int **radius** = 3
  - constant values for drawing*
- QVector< QPixmap > **ratingPxmmaps** = QVector< QPixmap >(10)
- QRect **ratingRect**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- QPolygon **starPolygon**
- QSize **starPolygonSize**
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

## 6.896 Digikam::ItemFilterModel Class Reference

Inheritance diagram for Digikam::ItemFilterModel:



### Public Types

- enum `ItemFilterModelRoles` {
  - `CategorizationModeRole` = `ItemModel::FilterModelRoles + 1` , `SortOrderRole` = `ItemModel::FilterModelRoles`

```
+ 2 , CategoryAlbumIdRole = ItemModel::FilterModelRoles + 3 , CategoryFormatRole = ItemModel::FilterModelRoles + 4 ,
CategoryDateRole = ItemModel::FilterModelRoles + 5 , CategoryFaceRole = ItemModel::FilterModelRoles + 6 ,
GroupsOpenRole = ItemModel::FilterModelRoles + 7 , ItemFilterModelPointerRole = ItemModel::FilterModelRoles + 50 }
```

## Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## Public Slots

- void [packageDiscarded](#) (const [ItemFilterModelTodoPackage](#) &package)
- void [packageFinished](#) (const [ItemFilterModelTodoPackage](#) &package)
- void [preprocessInfos](#) (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void [processAddedInfos](#) (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void [setAllGroupsOpen](#) (bool open)
- void [setCategorizationMode](#) ([ItemSortSettings::CategorizationMode](#) mode)
- void [setCategorizationSortOrder](#) ([ItemSortSettings::SortOrder](#) order)
- void [setDayFilter](#) (const QList< QDateTime > &days)
- void [setExceptionList](#) (const QList< qlonglong > &idlist, const QString &id)
- void [setGeolocationFilter](#) (const [ItemFilterSettings::GeolocationCondition](#) &condition)
- void [setGroupItemFilterSettings](#) (const [GroupItemFilterSettings](#) &settings)
- void [setGroupOpen](#) (qlonglong group, bool open)
- void [setIdWhitelist](#) (const QList< qlonglong > &idList, const QString &id)
- virtual void [setItemFilterSettings](#) (const [ItemFilterSettings](#) &settings)
- virtual void [setItemSortSettings](#) (const [ItemSortSettings](#) &settings)
- void [setMimeTypeFilter](#) (int mimeTypeFilter)
- void [setRatingFilter](#) (int rating, [ItemFilterSettings::RatingCondition](#) ratingCond, bool isUnratedExcluded)
- void [setSortOrder](#) ([ItemSortSettings::SortOrder](#) order)
- void [setSortRole](#) ([ItemSortSettings::SortRole](#) role)
- void [setStringTypeNatural](#) (bool natural)
- void [setTagFilter](#) (const QList< int > &includedTags, const QList< int > &excludedTags, [ItemFilterSettings::MatchingCondition](#) matchingCond, bool showUnTagged, const QList< int > &clTagIds, const QList< int > &plTagIds)
- void [setTextFilter](#) (const [SearchTextFilterSettings](#) &settings)
- void [setUrlWhitelist](#) (const QList< QUrl > &urlList, const QString &id)
- void [setVersionItemFilterSettings](#) (const [VersionItemFilterSettings](#) &settings)
- void [setVersionManagerSettings](#) (const [VersionManagerSettings](#) &settings)
- void [toggleGroupOpen](#) (qlonglong group)

## Signals

- void [filterMatches](#) (bool matches)
- void [filterMatchesForText](#) (bool matchesByText)
- void [filterSettingsChanged](#) (const [ItemFilterSettings](#) &settings)
- void [imageInfosAboutToBeRemoved](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosAdded](#) (const QList< [ItemInfo](#) > &infos)
- void [packageToFilter](#) (const [ItemFilterModelTodoPackage](#) &package)
- void [packageToPrepare](#) (const [ItemFilterModelTodoPackage](#) &package)
- void [reAddingFinished](#) ()
- void [reAddItemInfos](#) (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)



**Public Member Functions**

- void [addPrepareHook](#) ([ItemFilterModelPrepareHook](#) \*const hook)
- QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const override
- [GroupItemFilterSettings](#) **groupItemFilterSettings** () const
- [ItemFilterModel](#) \* [imageFilterModel](#) () const override
- [ItemFilterSettings](#) **imageFilterSettings** () const
- [ItemSortSettings](#) **imageSortSettings** () const
- void **infosToProcess** (const QList< [ItemInfo](#) > &infos)
- void **infosToProcess** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues, bool forReAdd=true)
- void **init** ([ItemFilterModel](#) \*qq)
- bool **isAllGroupsOpen** () const
- bool **isGroupOpen** (qulonglong group) const
- [ItemFilterModel](#) (QObject \*const parent=nullptr)
- [ItemFilterModelPrivate](#) ()
- void **removePrepareHook** ([ItemFilterModelPrepareHook](#) \*const hook)
- void **setSendItemInfoSignals** (bool sendSignals)
- void **setupWorkers** ()
- [DatabaseFields::Set suggestedWatchFlags](#) () const
- [VersionItemFilterSettings](#) **versionItemFilterSettings** () const

**Public Member Functions inherited from [Digikam::ImageSortFilterModel](#)**

- qulonglong **imageId** (const QModelIndex &index) const
- QList< qulonglong > **imageIds** (const QList< QModelIndex > &indexes) const
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- QList< [ItemInfo](#) > **imageInfos** (const QList< QModelIndex > &indexes) const
- QList< [ItemInfo](#) > **imageInfosSorted** () const
- [ImageSortFilterModel](#) (QObject \*const parent=nullptr)
- QModelIndex **indexForImageId** (qulonglong id) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info) const
- QModelIndex **indexForPath** (const QString &filePath) const
- QModelIndex **mapFromDirectSourceToSourceItemModel** (const QModelIndex &sourceModel\_index) const
- QModelIndex **mapFromSourceItemModel** (const QModelIndex &imagemodel\_index) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceItemModel** (const QModelIndex &index) const
- void **setSourceFilterModel** ([ImageSortFilterModel](#) \*const model)
- void **setSourceItemModel** ([ItemModel](#) \*const model)
- [ImageSortFilterModel](#) \* **sourceFilterModel** () const
- [ItemModel](#) \* **sourceItemModel** () const

**Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)**

- [DCategorizedSortFilterProxyModel](#) (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void **setCategorizedModel** (bool categorizedModel)
- void **setSortCategoriesByNaturalComparison** (bool [sortCategoriesByNaturalComparison](#))
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool **sortCategoriesByNaturalComparison** () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

## Public Attributes

- [ItemFilterSettings](#) **filter**
- [ItemFilterSettings](#) **filterCopy**
- [ItemFilterModelFilterer](#) \* **filterer** = nullptr
- [QHash](#)< [qlonglong](#), [bool](#) > **filterResults**
- [GroupItemFilterSettings](#) **groupFilter**
- [GroupItemFilterSettings](#) **groupFilterCopy**
- [bool](#) **hasOneMatch** = false
- [bool](#) **hasOneMatchForText** = false
- [ItemModel](#) \* **imageModel** = nullptr
- [unsigned int](#) **lastDiscardVersion** = 0
- [unsigned int](#) **lastFilteredVersion** = 0
- [QMutex](#) **mutex**
- [bool](#) **needPrepare** = false
- [bool](#) **needPrepareComments** = false
- [bool](#) **needPrepareGroups** = false
- [bool](#) **needPrepareTags** = false
- [QList](#)< [ItemFilterModelPrepareHook](#) \* > **prepareHooks**
- [ItemFilterModelPreparer](#) \* **preparer** = nullptr
- [ItemFilterModel](#) \* **q** = nullptr
- [int](#) **sentOut** = 0
- [int](#) **sentOutForReAdd** = 0
- [ItemSortSettings](#) **sorter**
- [QTimer](#) \* **updateFilterTimer** = nullptr
- [volatile unsigned int](#) **version** = 0
- [VersionItemFilterSettings](#) **versionFilter**
- [VersionItemFilterSettings](#) **versionFilterCopy**

## Protected Slots

- [void](#) **slotImageChange** (const [ImageChangeset](#) &changeset)
- [void](#) **slotImageTagChange** (const [ImageTagChangeset](#) &changeset)
- [void](#) **slotModelReset** ()
- [void](#) **slotRowsAboutToBeRemoved** (const [QModelIndex](#) &parent, [int](#) start, [int](#) end)
- [void](#) **slotRowsInserted** (const [QModelIndex](#) &parent, [int](#) start, [int](#) end)
- [void](#) **slotUpdateFilter** ()

## Protected Member Functions

- [virtual QString](#) **categoryIdentifier** (const [ItemInfo](#) &info, const [FaceTagsIface](#) &face) const
- [int](#) **compareCategories** (const [QModelIndex](#) &left, const [QModelIndex](#) &right) const override
- [virtual int](#) **compareInfosCategories** (const [ItemInfo](#) &left, const [ItemInfo](#) &right) const
- [virtual int](#) **compareInfosCategories** (const [ItemInfo](#) &left, const [ItemInfo](#) &right, const [FaceTagsIface](#) &leftFace, const [FaceTagsIface](#) &rightFace) const
- [bool](#) **filterAcceptsRow** ([int](#) source\_row, const [QModelIndex](#) &source\_parent) const override
- [virtual bool](#) **infosLessThan** (const [ItemInfo](#) &left, const [ItemInfo](#) &right) const
- **ItemFilterModel** ([ItemFilterModelPrivate](#) &dd, [QObject](#) \*const parent)
- [void](#) **setDirectSourceItemModel** ([ItemModel](#) \*const model) override
- [bool](#) **subSortLessThan** (const [QModelIndex](#) &left, const [QModelIndex](#) &right) const override

## Protected Member Functions inherited from [Digikam::ImageSortFilterModel](#)

- void **setSourceModel** (QAbstractItemModel \*const model) override  
*NOTE: made protected.*

## Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override

## Protected Attributes

- ItemFilterModelPrivate \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ImageSortFilterModel](#)

- [ImageSortFilterModel](#) \* **m\_chainedModel** = nullptr

## 6.896.1 Member Enumeration Documentation

### 6.896.1.1 ItemFilterModelRoles

```
enum Digikam::ItemFilterModel::ItemFilterModelRoles
```

#### Enumerator

CategorizationModeRole	Returns the current categorization mode.
SortOrderRole	Returns the current sort order.
CategoryAlbumIdRole	Returns the number of items in the index category. Returns the id of the <a href="#">PAlbum</a> of the index which is used for category
CategoryFormatRole	Returns the format of the index which is used for category.
CategoryDateRole	Returns the date of the index which is used for category.
CategoryFaceRole	Returns the suggested name for the face in this index.
GroupsOpenRole	Returns true if the given image is a group leader, and the group is opened.

## 6.896.2 Member Function Documentation

### 6.896.2.1 addPrepareHook()

```
void Digikam::ItemFilterModel::addPrepareHook (
    ItemFilterModelPrepareHook *const hook )
```

Add a hook to get added images for preparation tasks before they are added in the model

### 6.896.2.2 categoryIdentifier()

```
QString Digikam::ItemFilterModel::categoryIdentifier (
    const ItemInfo & info,
    const FaceTagsIface & face ) const [protected], [virtual]
```

Returns a unique identifier for the category if info. The string need not be for user display.

### 6.896.2.3 compareCategories()

```
int Digikam::ItemFilterModel::compareCategories (
    const QModelIndex & left,
    const QModelIndex & right ) const [override], [protected], [virtual]
```

This method compares the category of the `left` index with the category of the `right` index.

Internally and if not reimplemented, this method will ask for `left` and `right` models for role `CategorySortRole`. In order to correctly sort categories, the `data()` method of the model should return a `qlonglong` (or numeric) value, or a `QString` object. `QString` objects will be sorted with `QString::localeAwareCompare` if `sortCategoriesByNaturalComparison()` is true.

#### Note

Please have present that: `QString(QChar(QChar::ObjectReplacementCharacter)) > QString(QChar(QChar::ReplacementCharacter)) > [ all possible strings ] > QString();`

This means that `QString()` will be sorted the first one, while `QString(QChar(QChar::ObjectReplacementCharacter))` and `QString(QChar(QChar::ReplacementCharacter))` will be sorted in last position.

#### Warning

Please note that `data()` method of the model should return always information of the same type. If you return a `QString` for an index, you should return always `QStrings` for all indexes for role `CategorySortRole` in order to correctly sort categories. You can't mix by returning a `QString` for one index, and a `qlonglong` for other.

#### Note

If you need a more complex layout, you will have to reimplement this method.

#### Returns

A negative value if the category of `left` should be placed before the category of `right`. 0 if `left` and `right` are on the same category, and a positive value if the category of `left` should be placed after the category of `right`.

Reimplemented from [Digikam::DCategorizedSortFilterProxyModel](#).

#### 6.896.2.4 compareInfosCategories() [1/2]

```
int Digikam::ItemFilterModel::compareInfosCategories (
    const ItemInfo & left,
    const ItemInfo & right ) const [protected], [virtual]
```

Reimplement to customize category sorting. Return negative if category of left < category right, Return 0 if left and right are in the same category, else return positive.

Reimplemented in [Digikam::ItemAlbumFilterModel](#).

#### 6.896.2.5 compareInfosCategories() [2/2]

```
int Digikam::ItemFilterModel::compareInfosCategories (
    const ItemInfo & left,
    const ItemInfo & right,
    const FaceTagsIface & leftFace,
    const FaceTagsIface & rightFace ) const [protected], [virtual]
```

In order to be able to Categorize by Faces, it's necessary to pass in the face as well. One image may have multiple Faces in it, hence just the [ItemInfo](#) isn't sufficient.

Reimplemented in [Digikam::ItemAlbumFilterModel](#).

#### 6.896.2.6 data()

```
QVariant Digikam::ItemFilterModel::data (
    const QModelIndex & index,
    int role = Qt::DisplayRole ) const [override]
```

Keeping track of the Face (if any) associated with this Model Index is important to allow categorization by Face.

#### 6.896.2.7 filterMatches

```
void Digikam::ItemFilterModel::filterMatches (
    bool matches ) [signal]
```

Signals that the set filter matches at least one index

#### 6.896.2.8 filterMatchesForText

```
void Digikam::ItemFilterModel::filterMatchesForText (
    bool matchesByText ) [signal]
```

Signals that the set text filter matches at least one entry. If no text filter is set, this signal is emitted with 'false' when [filterMatches\(\)](#) is emitted.

### 6.896.2.9 filterSettingsChanged

```
void Digikam::ItemFilterModel::filterSettingsChanged (
    const ItemFilterSettings & settings ) [signal]
```

Emitted when the filter settings have been changed (the model may not yet have been updated)

### 6.896.2.10 imageFilterModel()

```
ItemFilterModel * Digikam::ItemFilterModel::imageFilterModel ( ) const [override], [virtual]
```

Returns this, any chained [ItemFilterModel](#), or 0.

Reimplemented from [Digikam::ImageSortFilterModel](#).

### 6.896.2.11 imageInfosAdded

```
void Digikam::ItemFilterModel::imageInfosAdded (
    const QList< ItemInfo > & infos ) [signal]
```

These signals need to be explicitly enabled with [setSendItemInfoSignals\(\)](#)

### 6.896.2.12 infosLessThan()

```
bool Digikam::ItemFilterModel::infosLessThan (
    const ItemInfo & left,
    const ItemInfo & right ) const [protected], [virtual]
```

Reimplement to customize sorting. Do not take categories into account here.

### 6.896.2.13 isGroupOpen()

```
bool Digikam::ItemFilterModel::isGroupOpen (
    qlonglong group ) const
```

group is identified by the id of its group leader

### 6.896.2.14 setDayFilter

```
void Digikam::ItemFilterModel::setDayFilter (
    const QList< QDateTime > & days ) [slot]
```

Adjust the current [ItemFilterSettings](#). Equivalent to retrieving the current filter settings, adjusting the parameter and calling [setItemFilterSettings](#). Provided for convenience. It is encouraged to use [setItemFilterSettings](#) if you change more than one parameter at a time.

### 6.896.2.15 setDirectSourceItemModel()

```
void Digikam::ItemFilterModel::setDirectSourceItemModel (
    ItemModel *const model ) [override], [protected], [virtual]
```

Reimplement if needed. Called only when model shall be set as (direct) sourceModel.

Reimplemented from [Digikam::ImageSortFilterModel](#).

### 6.896.2.16 setGroupItemFilterSettings

```
void Digikam::ItemFilterModel::setGroupItemFilterSettings (
    const GroupItemFilterSettings & settings ) [slot]
```

Changes the current version image filter settings and refilters.

### 6.896.2.17 setItemFilterSettings

```
void Digikam::ItemFilterModel::setItemFilterSettings (
    const ItemFilterSettings & settings ) [virtual], [slot]
```

Changes the current image filter settings and refilters.

Reimplemented in [Digikam::ItemAlbumFilterModel](#).

### 6.896.2.18 setItemSortSettings

```
void Digikam::ItemFilterModel::setItemSortSettings (
    const ItemSortSettings & settings ) [virtual], [slot]
```

Changes the current image sort settings and resorts.

### 6.896.2.19 setSendItemInfoSignals()

```
void Digikam::ItemFilterModel::setSendItemInfoSignals (
    bool sendSignals )
```

Enables sending imageInfosAdded and imageInfosAboutToBeRemoved

### 6.896.2.20 setVersionItemFilterSettings

```
void Digikam::ItemFilterModel::setVersionItemFilterSettings (
    const VersionItemFilterSettings & settings ) [slot]
```

Changes the current version image filter settings and refilters.

### 6.896.2.21 subSortLessThan()

```
bool Digikam::ItemFilterModel::subSortLessThan (
    const QModelIndex & left,
    const QModelIndex & right ) const [override], [protected], [virtual]
```

This method has a similar purpose as [lessThan\(\)](#) has on [QSortFilterProxyModel](#). It is used for sorting items that are in the same category.

#### Returns

Returns true if the item `left` is less than the item `right` when sorting.

Reimplemented from [Digikam::DCategorizedSortFilterProxyModel](#).

### 6.896.2.22 suggestedWatchFlags()

```
DatabaseFields::Set Digikam::ItemFilterModel::suggestedWatchFlags ( ) const
```

Returns a set of [DatabaseFields](#) suggested to set as watch flags on the source [ItemModel](#). The contained flags will be those that this model can sort or filter by.



## 6.897 Digikam::ItemFilterModelFilterer Class Reference

Inheritance diagram for Digikam::ItemFilterModelFilterer:



### Public Member Functions

- **ItemFilterModelFilterer** (ItemFilterModel::ItemFilterModelPrivate \*const d)
- void [process](#) (ItemFilterModelTodoPackage package) override

## Public Member Functions inherited from [Digikam::ItemFilterModelWorker](#)

- bool **checkVersion** (const [ItemFilterModelTodoPackage](#) &package)
- **ItemFilterModelWorker** (ItemFilterModel::ItemFilterModelPrivate \*const dd)

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const char \*method, Qt::ConnectionType type=Qt::AutoConnection) const
- QThread::Priority **priority** () const
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum **DeactivatingMode** { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::ItemFilterModelWorker](#)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void **deactivate** ([DeactivatingMode](#) mode=[FlushSignals](#))
- void **schedule** ()

## Signals inherited from [Digikam::ItemFilterModelWorker](#)

- void **discarded** (const [ItemFilterModelTodoPackage](#) &package)
- void **processed** (const [ItemFilterModelTodoPackage](#) &package)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## Protected Attributes inherited from [Digikam::ItemFilterModelWorker](#)

- ItemFilterModel::ItemFilterModelPrivate \* **d** = nullptr

### 6.897.1 Member Function Documentation

#### 6.897.1.1 process()

```
void Digikam::ItemFilterModelFilterer::process (  
    ItemFilterModelTodoPackage package ) [override], [virtual]
```

Implements [Digikam::ItemFilterModelWorker](#).

## 6.898 Digikam::ItemFilterModelPrepareHook Class Reference

### Public Member Functions

- virtual void **prepare** (const QVector< [ItemInfo](#) > &infos)=0

## 6.899 Digikam::ItemFilterModelPreparer Class Reference

Inheritance diagram for Digikam::ItemFilterModelPreparer:



### Public Member Functions

- **ItemFilterModelPreparer** (`ItemFilterModel::ItemFilterModelPrivate *const d`)
- void `process` (`ItemFilterModelTodoPackage` package) override

## Public Member Functions inherited from [Digikam::ItemFilterModelWorker](#)

- bool **checkVersion** (const [ItemFilterModelTodoPackage](#) &package)
- **ItemFilterModelWorker** (ItemFilterModel::ItemFilterModelPrivate \*const dd)

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const char \*method, Qt::ConnectionType type=Qt::AutoConnection) const
- QThread::Priority **priority** () const
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum **DeactivatingMode** { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::ItemFilterModelWorker](#)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void **deactivate** ([DeactivatingMode](#) mode=[FlushSignals](#))
- void **schedule** ()

## Signals inherited from [Digikam::ItemFilterModelWorker](#)

- void **discarded** (const [ItemFilterModelTodoPackage](#) &package)
- void **processed** (const [ItemFilterModelTodoPackage](#) &package)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## Protected Attributes inherited from [Digikam::ItemFilterModelWorker](#)

- ItemFilterModel::ItemFilterModelPrivate \* **d** = nullptr

## 6.899.1 Member Function Documentation

### 6.899.1.1 process()

```
void Digikam::ItemFilterModelPreparer::process (
    ItemFilterModelTodoPackage package ) [override], [virtual]
```

Implements [Digikam::ItemFilterModelWorker](#).

## 6.900 Digikam::ItemFilterModelTodoPackage Class Reference

### Public Member Functions

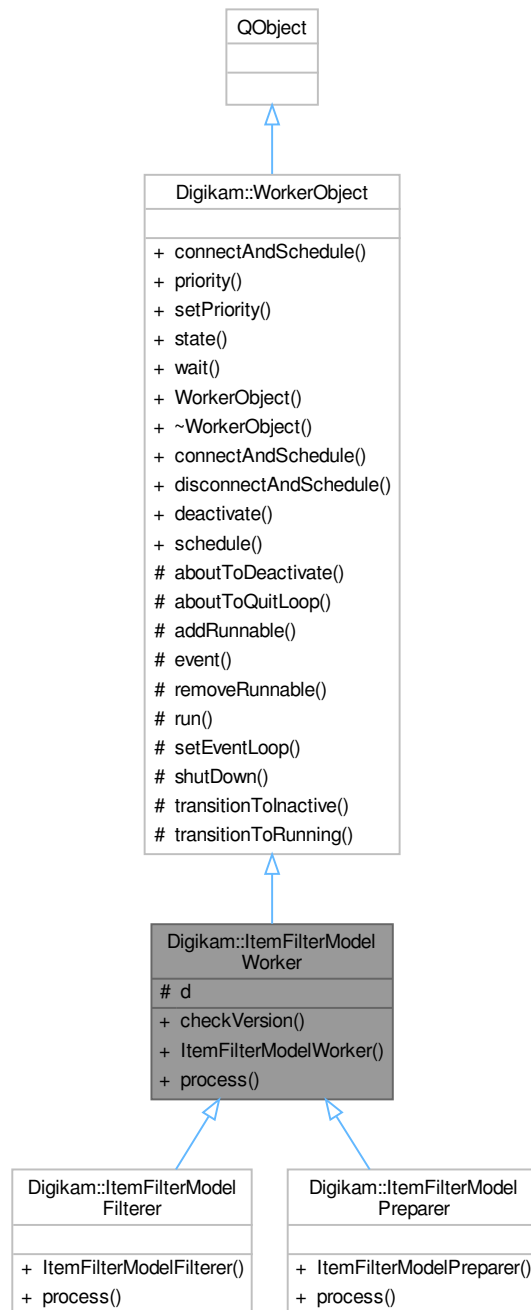
- **ItemFilterModelTodoPackage** (const QVector< [ItemInfo](#) > &infos, const QVector< QVariant > &extra↵  
Values, int version, bool isForReAdd)

### Public Attributes

- QVector< QVariant > **extraValues**
- QHash< qlonglong, bool > **filterResults**
- QVector< [ItemInfo](#) > **infos**
- bool **isForReAdd** = false
- unsigned int **version** = 0

## 6.901 Digikam::ItemFilterModelWorker Class Reference

Inheritance diagram for Digikam::ItemFilterModelWorker:



### Public Slots

- virtual void **process** ([ItemFilterModelTodoPackage](#) package)=0

## Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

## Signals

- void **discarded** (const [ItemFilterModelTodoPackage](#) &package)
- void **processed** (const [ItemFilterModelTodoPackage](#) &package)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Public Member Functions

- bool **checkVersion** (const [ItemFilterModelTodoPackage](#) &package)
- **ItemFilterModelWorker** ([ItemFilterModel::ItemFilterModelPrivate](#) \*const dd)

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const char \*method, [Qt::](#)↔[ConnectionType](#) type=[Qt::AutoConnection](#)) const
- [QThread::Priority](#) **priority** () const
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Protected Attributes

- [ItemFilterModel::ItemFilterModelPrivate](#) \* **d** = nullptr

## Additional Inherited Members

## Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, [Qt::](#)[ConnectionType](#) type=[Qt::AutoConnection](#))
- static bool **disconnectAndSchedule** (const [QObject](#) \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)



## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## 6.902 Digikam::ItemFilterSettings Class Reference

### Public Types

- enum [GeolocationCondition](#) { **GeolocationNoFilter** = 0 , **GeolocationNoCoordinates** = 1 << 1 , **GeolocationHasCoordinates** = 1 << 2 }
- Possible logical matching condition used to sort geolocation.*
- enum [MatchingCondition](#) { **OrCondition** , **AndCondition** }
- Possible logical matching condition used to sort tags id.*
- enum [RatingCondition](#) { **GreaterEqualCondition** , **EqualCondition** , **LessEqualCondition** }
- Possible conditions used to filter rating: >=, =, <=.*

### Public Member Functions

- bool **isFiltering** () const
- Returns if images will be filtered by these criteria at all.*
- bool **isFilteringByColorLabels** () const
- Returns if the color labels is a filter criteria.*
- bool **isFilteringByDay** () const
- Returns if the day is a filter criteria.*
- bool **isFilteringByGeolocation** () const
- Returns whether geolocation is a filter criteria.*
- bool **isFilteringByPickLabels** () const
- Returns if the pick labels is a filter criteria.*
- bool **isFilteringByRating** () const
- Returns if the rating is a filter criteria.*
- bool **isFilteringByTags** () const
- Returns if the tag is a filter criteria.*
- bool **isFilteringByText** () const
- Returns if the text (including comment) is a filter criteria.*
- bool **isFilteringByTypeMime** () const
- Returns if the type mime is a filter criteria.*
- bool **matches** (const [ItemInfo](#) &info, bool \*const foundText=nullptr) const
- void **setAlbumNames** (const QMap< int, QString > &albumNameHash)
- void **setDayFilter** (const QList< QDateTime > &days)
- Date filter —
- void **setGeolocationFilter** (const [GeolocationCondition](#) &condition)

- *Geolocation filter*
- void **setIdWhitelist** (const QList< qlonglong > &idList, const QString &id)
  - *ID whitelist filter*
- void **setMimeTypeFilter** (int mimeTypeFilter)
  - *Mime filter* —
- void **setRatingFilter** (int rating, [RatingCondition](#) ratingCond, bool isUnratedExcluded)
  - *Rating filter* —
- void **setTagFilter** (const QList< int > &includedTags, const QList< int > &excludedTags, [MatchingCondition](#) matchingCond, bool showUnTagged, const QList< int > &clTagIds, const QList< int > &plTagIds)
  - *Tags filter* —
- void **setTagNames** (const QHash< int, QString > &tagNameHash)
- void **setTextFilter** (const [SearchTextFilterSettings](#) &settings)
  - *Text filter* —
- void **setUrlWhitelist** (const QList< QUrl > &urlList, const QString &id)
  - *URL whitelist filter*
- [DatabaseFields::Set watchFlags](#) () const
  - *Change notification* —

## 6.902.1 Member Function Documentation

### 6.902.1.1 matches()

```
bool Digikam::ItemFilterSettings::matches (
    const ItemInfo & info,
    bool *const foundText = nullptr ) const
```

Returns true if the given [ItemInfo](#) matches the filter criteria. Optionally, foundText is set to true if it matched by text search.

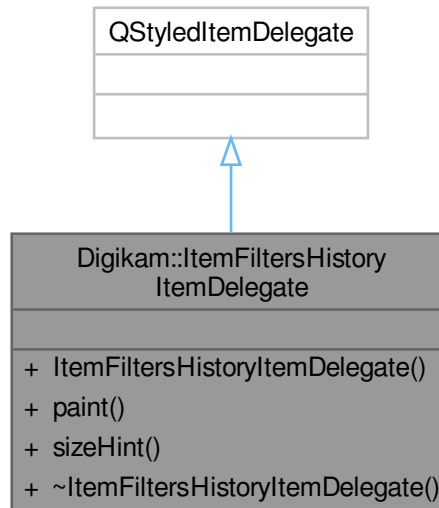
### 6.902.1.2 watchFlags()

```
DatabaseFields::Set Digikam::ItemFilterSettings::watchFlags ( ) const
```

Returns database fields a change in which would affect the current filtering. To find out if an image tag change affects filtering, test [isFilteringByTags\(\)](#). The text filter will also be affected by changes in tags and album names.

## 6.903 Digikam::ItemFiltersHistoryItemDelegate Class Reference

Inheritance diagram for Digikam::ItemFiltersHistoryItemDelegate:

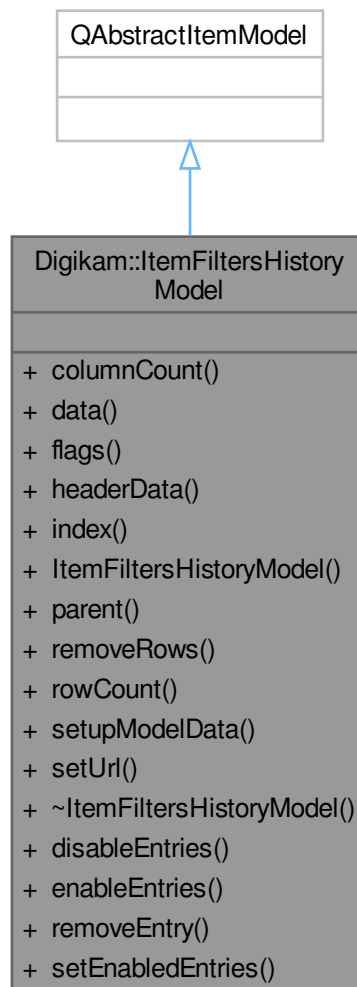


### Public Member Functions

- **ItemFiltersHistoryItemDelegate** (`QObject *const parent=nullptr`)
- void **paint** (`QPainter *painter, const QStyleOptionViewItem &option, const QModelIndex &index`) const override
- `QSize` **sizeHint** (`const QStyleOptionViewItem &option, const QModelIndex &index`) const override

## 6.904 Digikam::ItemFiltersHistoryModel Class Reference

Inheritance diagram for Digikam::ItemFiltersHistoryModel:



### Public Slots

- void **disableEntries** (int count)
- void **enableEntries** (int count)
- void **removeEntry** (const QModelIndex &index)
- void **setEnabledEntries** (int count)

### Public Member Functions

- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override

- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- **ItemFiltersHistoryModel** (QObject \*const parent=nullptr, const QUrl &url=QUrl())
- QModelIndex **parent** (const QModelIndex &index) const override
- bool **removeRows** (int row, int count, const QModelIndex &parent) override
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setupModelData** (const QList< [DImageHistory::Entry](#) > &entries, [ItemFiltersHistoryTreeltem](#) \*parent=nullptr)
- void **setUrl** (const QUrl &url)

## 6.905 Digikam::ItemFiltersHistoryTreeltem Class Reference

### Public Member Functions

- void **appendChild** ([ItemFiltersHistoryTreeltem](#) \*const child)
- [ItemFiltersHistoryTreeltem](#) \* **child** (int row) const
- int **childCount** () const
- int **columnCount** () const
- QVariant **data** (int column) const
- bool **isDisabled** () const
- [ItemFiltersHistoryTreeltem](#) (const QList< QVariant > &data, [ItemFiltersHistoryTreeltem](#) \*const parent=nullptr)
- [ItemFiltersHistoryTreeltem](#) (const QString &data, [ItemFiltersHistoryTreeltem](#) \*const parent=nullptr)
- [ItemFiltersHistoryTreeltem](#) \* **parent** () const
- void **removeChild** (int row)
- int **row** () const
- void **setDisabled** (bool disabled) const

## 6.906 Digikam::ItemFullScreenOverlay Class Reference

Inheritance diagram for Digikam::ItemFullScreenOverlay:



### Signals

- void **signalFullscreen** (const QList< QModelIndex > &indexes)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- **ItemFullScreenOverlay** (QObject \*const parent)
- void **setActive** (bool active) override

## Public Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- **ItemViewHoverButton** \* **button** () const
- **HoverButtonDelegateOverlay** (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- **AbstractWidgetDelegateOverlay** (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Static Public Member Functions

- static **ItemFullScreenOverlay** \* **instance** (QObject \*const parent)

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- **ItemViewHoverButton** \* **createButton** () override
- void **updateButton** (const QModelIndex &index) override
- void **widgetEnterEvent** () override
- void **widgetLeaveEvent** () override

## Protected Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- QWidget \* **createWidget** () override
- void **visualChange** () override

### Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- void [widgetLeaveNotifyMultiple](#) ()

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

### Additional Inherited Members

### Protected Slots inherited from [Digikam::HoverButtonDelegateOverlay](#)

- void [slotEntered](#) (const QModelIndex &index) override
- void [slotReset](#) () override

### Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void [slotEntered](#) (const QModelIndex &index)
- virtual void [slotLayoutChanged](#) ()
- virtual void [slotReset](#) ()
- virtual void [slotRowsRemoved](#) (const QModelIndex &parent, int start, int end)
- virtual void [slotViewportEntered](#) ()

### Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

### Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr



## 6.906.1 Member Function Documentation

### 6.906.1.1 checkIndex()

```
bool Digikam::ItemFullScreenOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.906.1.2 createButton()

```
ItemViewHoverButton * Digikam::ItemFullScreenOverlay::createButton ( ) [override], [protected],
[virtual]
```

Create your widget here. Pass view() as parent.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.906.1.3 setActive()

```
void Digikam::ItemFullScreenOverlay::setActive (
    bool active ) [override], [virtual]
```

Will call [createButton\(\)](#).

Reimplemented from [Digikam::HoverButtonDelegateOverlay](#).

### 6.906.1.4 updateButton()

```
void Digikam::ItemFullScreenOverlay::updateButton (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.906.1.5 widgetEnterEvent()

```
void Digikam::ItemFullScreenOverlay::widgetEnterEvent ( ) [override], [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.906.1.6 widgetLeaveEvent()

```
void Digikam::ItemFullScreenOverlay::widgetLeaveEvent ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.907 Digikam::ItemFullScreenOverlayButton Class Reference

Inheritance diagram for Digikam::ItemFullScreenOverlayButton:



## Public Member Functions

- **ItemFullScreenOverlayButton** (QAbstractItemView \*const parentView)
- QSize **sizeHint** () const override

## Public Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- QModelIndex **index** () const
- void **initIcon** ()
- **ItemViewHoverButton** (QAbstractItemView \*const parentView)
- void **reset** ()
- void **setIndex** (const QModelIndex &index)
- void **setVisible** (bool visible) override

## Protected Member Functions

- QIcon **icon** () override
- void **updateToolTip** () override

## Protected Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- void **enterEvent** (QEnterEvent \*event)
- void **leaveEvent** (QEvent \*event)
- void **paintEvent** (QPaintEvent \*event)
- void **setup** ()

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ItemViewHoverButton](#)

- void **refreshIcon** ()
- void **setFadingValue** (int value)
- void **startFading** ()
- void **stopFading** ()

## Protected Attributes inherited from [Digikam::ItemViewHoverButton](#)

- QTimerLine \* **m\_fadingTimeLine** = nullptr
- int **m\_fadingValue** = 0
- QIcon **m\_icon**
- QPersistentModelIndex **m\_index**
- bool **m\_isHovered** = false

## 6.907.1 Member Function Documentation

### 6.907.1.1 icon()

```
QIcon Digikam::ItemFullScreenOverlayButton::icon ( ) [override], [protected], [virtual]
```

Return your icon here. Will be queried again on toggle.

Implements [Digikam::ItemViewHoverButton](#).

### 6.907.1.2 sizeHint()

```
QSize Digikam::ItemFullScreenOverlayButton::sizeHint ( ) const [override], [virtual]
```

Reimplement to match the size of your icon

Implements [Digikam::ItemViewHoverButton](#).

### 6.907.1.3 updateToolTip()

```
void Digikam::ItemFullScreenOverlayButton::updateToolTip ( ) [override], [protected], [virtual]
```

Optionally update tooltip here. Will be called again on state change.

Reimplemented from [Digikam::ItemViewHoverButton](#).

## 6.908 Digikam::ItemGPS Class Reference

Inheritance diagram for Digikam::ItemGPS:



### Public Member Functions

- `ItemGPS` (const `ItemInfo` &info)
- bool `loadImageData` () override
- QString `saveChanges` () override

## Public Member Functions inherited from [Digikam::GPSItemContainer](#)

- **GPSItemContainer** (const [QUrl](#) &url)
- bool **isDirty** () const
- [QUrl](#) **url** () const
- [QDateTime](#) **dateTime** () const
- void **setCoordinates** (const [GeoCoordinates](#) &newCoordinates)
- [GeoCoordinates](#) **coordinates** () const
- [GPSDataContainer](#) **gpsData** () const
- void **setGPSData** (const [GPSDataContainer](#) &container)
- void **restoreGPSData** (const [GPSDataContainer](#) &container)
 

*Restore the gps data to `container`. Sets `m_dirty` to false if container equals savedState.*
- void **setTagList** (const [QList](#)< [QList](#)< [TagData](#) > > &externalTagList)
- bool **isTagListDirty** () const
- [QList](#)< [QList](#)< [TagData](#) > > **getTagList** () const
- void **restoreRGTagList** (const [QList](#)< [QList](#)< [TagData](#) > > &tagList)
- void **writeTagsToXmp** (const bool writeXmpTags)
- void **writeLocations** (const bool writeMetaLoc)
- bool **lessThan** (const [GPSItemContainer](#) \*const otherItem, const int column) const

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::GPSItemContainer](#)

- static void **setHeaderData** ([GPSItemModel](#) \*const model)

## Static Public Attributes inherited from [Digikam::GPSItemContainer](#)

- static const int **ColumnAccuracy** = 6
- static const int **ColumnAltitude** = 5
- static const int **ColumnDateTime** = 2
- static const int **ColumnDOP** = 9
- static const int **ColumnFilename** = 1
- static const int **ColumnFixType** = 10
- static const int **ColumnGPSItemContainerCount** = 13
- static const int **ColumnLatitude** = 3
- static const int **ColumnLongitude** = 4
- static const int **ColumnNSatellites** = 11
- static const int **ColumnSpeed** = 12
- static const int **ColumnStatus** = 8
- static const int **ColumnTags** = 7
- static const int **ColumnThumbnail** = 0
- static const int **RoleCoordinates** = [Qt::UserRole](#) + 1

## Protected Member Functions inherited from [Digikam::GPSItemContainer](#)

- void **setLocationInfo** (const [TagData](#) &tagData, [IptcCoreLocationInfo](#) &locationInfo)
- QVariant **data** (const int column, const int role) const  
*these are only to be called by the [GPSItemModel](#)*
- void **setModel** ([GPSItemModel](#) \*const model)
- void **emitDataChanged** ()
- [DMetadata](#) \* **getMetadataForFile** () const
- [SaveProperties](#) **saveProperties** () const

## Protected Attributes inherited from [Digikam::GPSItemContainer](#)

- [GPSItemModel](#) \* **m\_model** = nullptr
- [QUrl](#) **m\_url**
- [QDateTime](#) **m\_dateTime**
- bool **m\_dirty** = false
- [GPSDataContainer](#) **m\_gpsData**
- [GPSDataContainer](#) **m\_savedState**
- bool **m\_tagListDirty** = false
- [QList](#)< [QList](#)< [TagData](#) > > **m\_tagList**
- [QList](#)< [QList](#)< [TagData](#) > > **m\_savedTagList**
- bool **m\_writeXmpTags** = true
- bool **m\_writeMetaLoc** = true

### 6.908.1 Member Function Documentation

#### 6.908.1.1 loadImageData()

```
bool Digikam::ItemGPS::loadImageData ( ) [override], [virtual]
```

Reimplemented from [Digikam::GPSItemContainer](#).

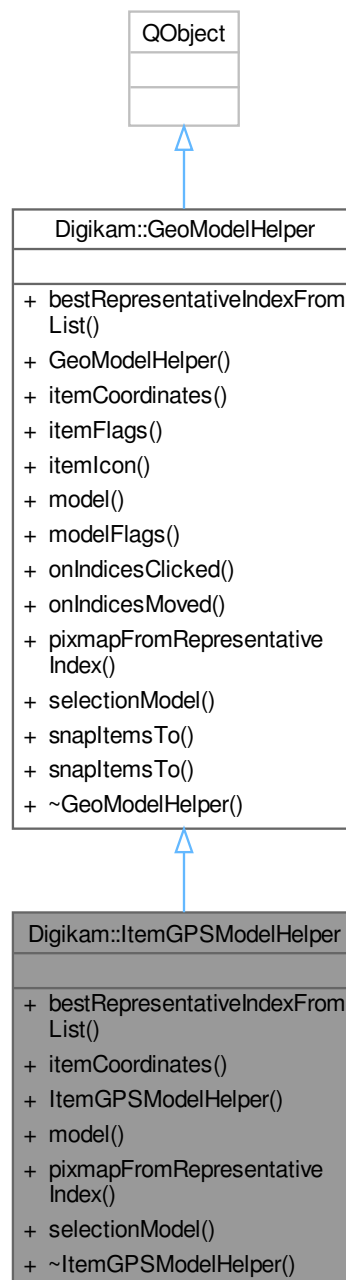
### 6.908.1.2 saveChanges()

QString Digikam::ItemGPS::saveChanges ( ) [override], [virtual]

Reimplemented from [Digikam::GPSItemContainer](#).

## 6.909 Digikam::ItemGPSModelHelper Class Reference

Inheritance diagram for Digikam::ItemGPSModelHelper:





## Public Member Functions

- QPersistentModelIndex [bestRepresentativeIndexFromList](#) (const QList< QPersistentModelIndex > &list, const int sortKey) override
- bool [itemCoordinates](#) (const QModelIndex &index, [GeoCoordinates](#) \*const coordinates) const override
- **ItemGPSModelHelper** (QStandardItemModel \*const itemModel, QObject \*const parent=nullptr)
- QAbstractItemModel \* [model](#) () const override  
*these are necessary for grouped and ungrouped models*
- QPixmap [pixmapFromRepresentativeIndex](#) (const QPersistentModelIndex &index, const QSize &size) override  
*these are used by MarkerModel for grouped models*
- QItemSelectionModel \* [selectionModel](#) () const override

## Public Member Functions inherited from [Digikam::GeoModelHelper](#)

- **GeoModelHelper** (QObject \*const parent=nullptr)
- virtual PropertyFlags **itemFlags** (const QModelIndex &index) const
- virtual bool [itemIcon](#) (const QModelIndex &index, QPoint \*const offset, QSize \*const size, QPixmap \*const pixmap, QUrl \*const url) const  
*these are necessary for ungrouped models*
- virtual PropertyFlags **modelFlags** () const
- virtual void [onIndicesClicked](#) (const QList< QPersistentModelIndex > &clickedIndices)
- virtual void **onIndicesMoved** (const QList< QPersistentModelIndex > &movedIndices, const [GeoCoordinates](#) &targetCoordinates, const QPersistentModelIndex &targetSnapIndex)
- virtual void **snapItemsTo** (const QModelIndex &targetIndex, const QList< QModelIndex > &snappedIndices)
- void **snapItemsTo** (const QModelIndex &targetIndex, const QList< QPersistentModelIndex > &snappedIndices)

## Additional Inherited Members

## Public Types inherited from [Digikam::GeoModelHelper](#)

- enum **PropertyFlag** { **FlagNull** = 0 , **FlagVisible** = 1 , **FlagMovable** = 2 , **FlagSnaps** = 4 }

## Signals inherited from [Digikam::GeoModelHelper](#)

- void **signalModelChangedDrastically** ()
- void **signalThumbnailAvailableForIndex** (const QPersistentModelIndex &index, const QPixmap &pixmap)
- void **signalVisibilityChanged** ()

## 6.909.1 Member Function Documentation

### 6.909.1.1 [bestRepresentativeIndexFromList\(\)](#)

```
QPersistentModelIndex Digikam::ItemGPSModelHelper::bestRepresentativeIndexFromList (
    const QList< QPersistentModelIndex > & list,
    const int sortKey ) [override], [virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.909.1.2 itemCoordinates()

```
bool Digikam::ItemGPSModelHelper::itemCoordinates (
    const QModelIndex & index,
    GeoCoordinates *const coordinates ) const [override], [virtual]
```

Implements [Digikam::GeoModelHelper](#).

### 6.909.1.3 model()

```
QAbstractItemModel * Digikam::ItemGPSModelHelper::model ( ) const [override], [virtual]
```

Implements [Digikam::GeoModelHelper](#).

### 6.909.1.4 pixmapFromRepresentativeIndex()

```
QPixmap Digikam::ItemGPSModelHelper::pixmapFromRepresentativeIndex (
    const QPersistentModelIndex & index,
    const QSize & size ) [override], [virtual]
```

Reimplemented from [Digikam::GeoModelHelper](#).

### 6.909.1.5 selectionModel()

```
QItemSelectionModel * Digikam::ItemGPSModelHelper::selectionModel ( ) const [override], [virtual]
```

Implements [Digikam::GeoModelHelper](#).

## 6.910 Digikam::ItemHistoryGraph Class Reference

### Public Types

- enum [HistoryLoadingFlag](#) { [LoadRelationCloud](#) = 1 << 0 , [LoadSubjectHistory](#) = 1 << 1 , [LoadLeavesHistory](#) = 1 << 2 , [LoadAll](#) = LoadRelationCloud | LoadSubjectHistory | LoadLeavesHistory }
- enum [ProcessingMode](#) { [NoProcessing](#) , [PrepareForDisplay](#) }

## Public Member Functions

- void **addHistory** (const [DImageHistory](#) &history, const [HistoryImageld](#) &historySubject=[HistoryImageld](#)())
- void [addHistory](#) (const [DImageHistory](#) &history, const [ItemInfo](#) &historySubject=[ItemInfo](#)())
- void [addRelations](#) (const [QList](#)< [QPair](#)< [qlonglong](#), [qlonglong](#) > > &pairs)
- void [addScannedHistory](#) (const [DImageHistory](#) &history, [qlonglong](#) historySubjectId)
- [QList](#)< [qlonglong](#) > **allImagelds** () const
- [QList](#)< [ItemInfo](#) > [allImages](#) () const
- [QHash](#)< [ItemInfo](#), [HistoryImageld::Types](#) > [categorize](#) () const
- void [clear](#) ()
- [ItemHistoryGraphData](#) & **data** ()
- const [ItemHistoryGraphData](#) & **data** () const
- void [dropUnresolvedEntries](#) ()
- bool [hasEdges](#) () const
- bool [hasUnresolvedEntries](#) () const
- bool **isEmpty** () const
- bool **isNull** () const
- bool **isSingleVertex** () const
- [ItemHistoryGraph](#) (const [ItemHistoryGraph](#) &other)
- [QList](#)< [ItemInfo](#) > [leafImages](#) () const
- [ItemHistoryGraph](#) & **operator=** (const [ItemHistoryGraph](#) &other)
- void [prepareForDisplay](#) (const [ItemInfo](#) &subject)
- void [reduceEdges](#) ()
- [QList](#)< [QPair](#)< [qlonglong](#), [qlonglong](#) > > [relationCloud](#) () const
- [QPair](#)< [QList](#)< [qlonglong](#) >, [QList](#)< [qlonglong](#) > > **relationCloudParallel** () const
- [QList](#)< [ItemInfo](#) > [rootImages](#) () const
- void [sortForInfo](#) (const [ItemInfo](#) &subject)

## Static Public Member Functions

- static [ItemHistoryGraph](#) [fromInfo](#) (const [ItemInfo](#) &info, [HistoryLoadingMode](#) loadingMode=[LoadAll](#), [ProcessingMode](#) processingMode=[PrepareForDisplay](#))

## 6.910.1 Member Enumeration Documentation

### 6.910.1.1 HistoryLoadingFlag

```
enum Digikam::ItemHistoryGraph::HistoryLoadingFlag
```

#### Enumerator

LoadRelationCloud	Load the relation cloud to the graph. Will give all edges, but no further info.
LoadSubjectHistory	Will load the <a href="#">DImageHistory</a> of the given subject.
LoadLeavesHistory	Will load the <a href="#">DImageHistory</a> of all leave vertices of the graph.

## 6.910.2 Member Function Documentation

### 6.910.2.1 addHistory()

```
void Digikam::ItemHistoryGraph::addHistory (
```

```
const DImageHistory & history,
const ItemInfo & historySubject = ItemInfo() )
```

Add the given history. The optionally given info or id is used as the "current" image of the history. If you read a history from a file's metadata or the database, you shall give the relevant subject.

#### 6.910.2.2 addRelations()

```
void Digikam::ItemHistoryGraph::addRelations (
    const QList< QPair< qlonglong, qlonglong > > & pairs )
```

Add images and their relations from the given pairs. Each pair (a,b) means "a is derived from b".

#### 6.910.2.3 addScannedHistory()

```
void Digikam::ItemHistoryGraph::addScannedHistory (
    const DImageHistory & history,
    qlonglong historySubjectId )
```

This is very similar to addHistory. The only difference is that no attempt is made to retrieve an [ItemInfo](#) for the historySubjectId. Can be useful in the context of scanning

#### 6.910.2.4 allImages()

```
QList< ItemInfo > Digikam::ItemHistoryGraph::allImages ( ) const
```

Returns image infos / ids from all vertices in this graph.

#### 6.910.2.5 categorize()

```
QHash< ItemInfo, HistoryImageId::Types > Digikam::ItemHistoryGraph::categorize ( ) const
```

Attempts at a categorization of all images in the graph into the types defined by [HistoryImageId](#). The type will be invalid if no decision can be made due to conflicting data.

#### 6.910.2.6 clear()

```
void Digikam::ItemHistoryGraph::clear ( )
```

Clears this graph.

#### 6.910.2.7 dropUnresolvedEntries()

```
void Digikam::ItemHistoryGraph::dropUnresolvedEntries ( )
```

Remove all vertices from the graph for which no existing [ItemInfo](#) could be found in the database.

**6.910.2.8 fromInfo()**

```
ItemHistoryGraph Digikam::ItemHistoryGraph::fromInfo (
    const ItemInfo & info,
    HistoryLoadingMode loadingMode = LoadAll,
    ProcessingMode processingMode = PrepareForDisplay ) [static]
```

Convenience: Reads all available history for the given info from the database and returns the created graph. Depending on mode, the graph will be preparedForDisplay(). If no history is recorded and no relations found, a single-vertex graph is returned.

**6.910.2.9 hasEdges()**

```
bool Digikam::ItemHistoryGraph::hasEdges ( ) const
```

Returns if the graph contains any edges. Because loops are not allowed, this also means (!isEmpty() && !isSingleVertex()).

**6.910.2.10 hasUnresolvedEntries()**

```
bool Digikam::ItemHistoryGraph::hasUnresolvedEntries ( ) const
```

Returns true if for any entry no [ItemInfo](#) could be located.

**6.910.2.11 leafImages()**

```
QList< ItemInfo > Digikam::ItemHistoryGraph::leafImages ( ) const
```

Returns image infos / ids from all leaf vertices in this graph, i.e. vertices with no subsequent history.

**6.910.2.12 prepareForDisplay()**

```
void Digikam::ItemHistoryGraph::prepareForDisplay (
    const ItemInfo & subject )
```

Combines [reduceEdges\(\)](#), [dropOrphans\(\)](#) and [sortForInfo\(\)](#).

**6.910.2.13 reduceEdges()**

```
void Digikam::ItemHistoryGraph::reduceEdges ( )
```

Remove edges which provide only duplicate information (performs a transitive reduction). Especially call this when [addRelations\(\)](#) was used.

**6.910.2.14 relationCloud()**

```
QList< QPair< qlonglong, qlonglong > > Digikam::ItemHistoryGraph::relationCloud ( ) const
```

Returns all possible relations between images in this graph, the edges of the transitive closure. The first variant returns (1,2),(3,4),(6,8), the second (1,3,6)(2,4,8).

**6.910.2.15 rootImages()**

```
QList< ItemInfo > Digikam::ItemHistoryGraph::rootImages ( ) const
```

Returns image infos / ids from all root vertices in this graph, i.e. vertices with no precedent history.

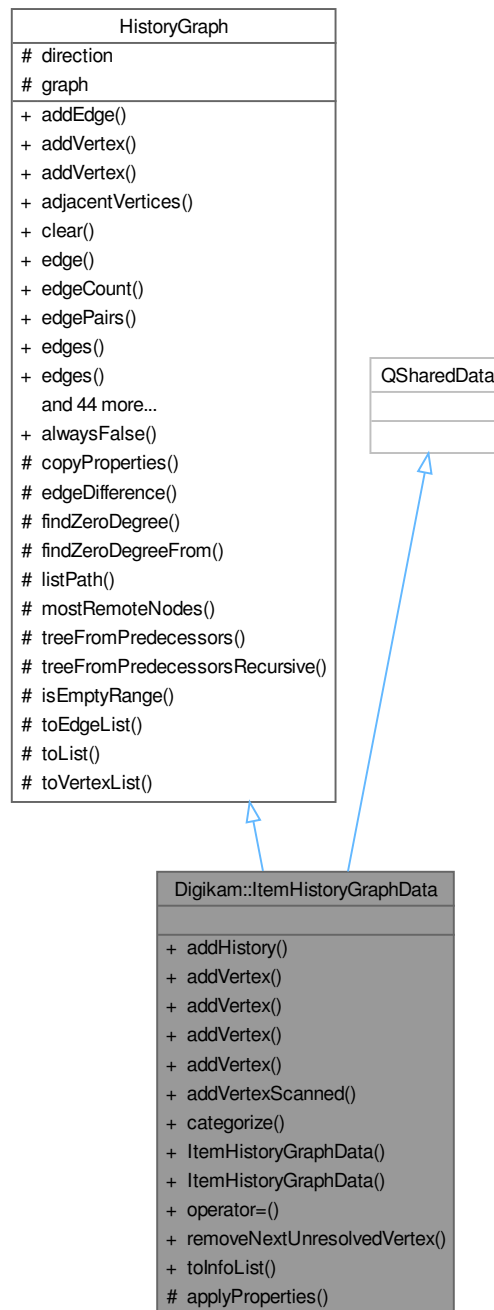
**6.910.2.16 sortForInfo()**

```
void Digikam::ItemHistoryGraph::sortForInfo (
    const ItemInfo & subject )
```

Sort vertex information prioritizing for the given vertex.

## 6.911 Digikam::ItemHistoryGraphData Class Reference

Inheritance diagram for Digikam::ItemHistoryGraphData:



### Public Member Functions

- void **addHistory** (const [DImageHistory](#) &givenHistory, qlonglong extraCurrent=0)
- [Vertex](#) **addVertex** (const [HistoryImageId](#) &id)

- [Vertex](#) **addVertex** (const [ItemInfo](#) &info)
- [Vertex](#) **addVertex** (const QList< [HistoryImageld](#) > &imagelds)
- [Vertex](#) **addVertex** (qulonglong id)
- [Vertex](#) **addVertexScanned** (qulonglong id)
- QHash< [Vertex](#), HistoryImageld::Types > **categorize** () const
- [ItemHistoryGraphData](#) (const [HistoryGraph](#) &g)
- [ItemHistoryGraphData](#) & **operator=** (const [HistoryGraph](#) &g)
- int **removeNextUnresolvedVertex** (int begin)
- QList< [ItemInfo](#) > **toInfoList** (const QList< [Vertex](#) > &vertices) const

## Public Member Functions inherited from [Digikam::Graph](#)< [VertexProperties](#), [EdgeProperties](#) >

- [Edge](#) **addEdge** (const [Vertex](#) &v1, const [Vertex](#) &v2)
- [Vertex](#) **addVertex** ()
- [Vertex](#) **addVertex** (const [VertexProperties](#) &properties)
- QList< [Vertex](#) > **adjacentVertices** (const [Vertex](#) &v, [AdjacencyFlags](#) flags=AllEdges) const
- void **clear** ()
- [Edge](#) **edge** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- int **edgeCount** () const
- QList< [VertexPair](#) > **edgePairs** () const
- QList< [Edge](#) > **edges** () const
- QList< [Edge](#) > **edges** (const [Vertex](#) &v, [AdjacencyFlags](#) flags=AllEdges) const
- template<class T >  
[Vertex](#) **findVertexByProperties** (const T &value) const
- const [GraphContainer](#) & **getGraph** () const
- [Graph](#) (const [Graph](#) &g)
- [Graph](#) ([MeaningOfDirection](#) dir=ParentToChild)
- bool **hasEdge** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- bool **hasEdges** () const
- bool **hasEdges** (const [Vertex](#) &v, [AdjacencyFlags](#) flags=AllEdges) const
- int **inDegree** (const [Vertex](#) &v) const
- bool **isConnected** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- bool **isEmpty** () const
- bool **isLeaf** (const [Vertex](#) &v) const
- bool **isRoot** (const [Vertex](#) &v) const
- QList< [Vertex](#) > **leaves** () const
- QList< [Vertex](#) > **leavesFrom** (const [Vertex](#) &v) const
- QList< [Vertex](#) > **longestPathTouching** (const [Vertex](#) &v) const
- template<typename LessThan >  
QList< [Vertex](#) > **longestPathTouching** (const [Vertex](#) &v, LessThan lessThan) const
- [MeaningOfDirection](#) **meaningOfDirection** () const
- [Graph](#) & **operator=** (const [Graph](#) &other)
- int **outDegree** (const [Vertex](#) &v) const
- [EdgeProperties](#) & **properties** (const [Edge](#) &e)
- const [EdgeProperties](#) & **properties** (const [Edge](#) &e) const
- [VertexProperties](#) & **properties** (const [Vertex](#) &v)
- const [VertexProperties](#) & **properties** (const [Vertex](#) &v) const
- [EdgeProperties](#) **properties** (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- void **remove** (const [Vertex](#) &v)
- QList< [Vertex](#) > **roots** () const
- QList< [Vertex](#) > **rootsOf** (const [Vertex](#) &v) const
- void **setProperties** (const [Edge](#) &e, const [EdgeProperties](#) &props)
- void **setProperties** (const [Vertex](#) &v, const [VertexProperties](#) &props)



- QMap< [Vertex](#), int > [shortestDistancesFrom](#) (const [Vertex](#) &v) const
- QList< [Vertex](#) > [shortestPath](#) (const [Vertex](#) &v1, const [Vertex](#) &v2) const
- [Vertex source](#) (const [Edge](#) &e) const
- [Vertex target](#) (const [Edge](#) &e) const
- QList< [Vertex](#) > [topologicalSort](#) () const
- [Graph transitiveClosure](#) (GraphCopyFlags flags=CopyAllProperties) const
- [Graph transitiveReduction](#) (QList< [Edge](#) > \*removedEdges=0, GraphCopyFlags flags=CopyAllProperties) const
- int [vertexCount](#) () const
  - NOTE: for "hasAdjacentVertices", simply use `hasEdges(v, flags)`.
- QList< [Vertex](#) > [vertices](#) () const
- QList< [Vertex](#) > [verticesBreadthFirst](#) (const [Vertex](#) &givenRef=[Vertex](#)()) const
- template<typename LessThan >
  - QList< [Vertex](#) > [verticesDepthFirstSorted](#) (const [Vertex](#) &givenRef, LessThan lessThan) const
- QList< [Vertex](#) > [verticesDominatedBy](#) (const [Vertex](#) &v, const [Vertex](#) &root, const QList< [Vertex](#) > &presortedVertices) const
- QList< [Vertex](#) > [verticesDominatedBy](#) (const [Vertex](#) &v, const [Vertex](#) &root, ReturnOrder order=Breadth↔FirstOrder) const
- template<typename LessThan >
  - QList< [Vertex](#) > [verticesDominatedByDepthFirstSorted](#) (const [Vertex](#) &v, const [Vertex](#) &root, LessThan lessThan) const

### Protected Member Functions

- void [applyProperties](#) ([Vertex](#) &v, const QList< [ItemInfo](#) > &infos, const QList< [HistoryImageId](#) > &ids)

### Protected Member Functions inherited from [Digikam::Graph](#)< [VertexProperties](#), [EdgeProperties](#) >

- void [copyProperties](#) ([Graph](#) &other, GraphCopyFlags flags, const std::vector< [vertex\\_t](#) > &copiedVertices) const
- QList< [Edge](#) > [edgeDifference](#) (const [Graph](#) &other, const std::vector< [vertex\\_t](#) > &copiedVertices) const
- QList< [Vertex](#) > [findZeroDegree](#) (bool inOrOut) const
- QList< [Vertex](#) > [findZeroDegreeFrom](#) (const [Vertex](#) &v, bool inOrOut) const
- QList< [Vertex](#) > [listPath](#) (const [Vertex](#) &root, const [Vertex](#) &target, const [VertexVertexMap](#) &predecessors, [MeaningOfDirection](#) dir=ParentToChild) const
- QList< [Vertex](#) > [mostRemoteNodes](#) (const [VertexIntMap](#) &distances) const
- QList< [Vertex](#) > [treeFromPredecessors](#) (const [Vertex](#) &v, const [VertexVertexMap](#) &predecessors) const
- void [treeFromPredecessorsRecursive](#) (const [Vertex](#) &v, QList< [Vertex](#) > &vertices, const [VertexVertexMap](#) &predecessors) const

### Additional Inherited Members

### Public Types inherited from [Digikam::Graph](#)< [VertexProperties](#), [EdgeProperties](#) >

- typedef graph\_traits::adjacency\_iterator [adjacency\\_iter](#)
- typedef std::pair< [adjacency\\_iter](#), [adjacency\\_iter](#) > [adjacency\\_vertex\\_range\\_t](#)
- enum [AdjacencyFlags](#) {
  - [OutboundEdges](#) = 1 << 0 , [InboundEdges](#) = 1 << 1 , [EdgesToLeaf](#) = 1 << 2 , [EdgesToRoot](#) = 1 << 3 ,
  - [AllEdges](#) = InboundEdges | OutboundEdges }
- typedef boost::property\_map< [GraphContainer](#), [edge\\_properties\\_t](#) >::const\_type [const\\_edge\\_property](#)↔  
[\\_map\\_t](#)

- typedef boost::property\_map< GraphContainer, boost::vertex\_index\_t >::const\_type **const\_vertex\_index\_map\_t**
- typedef boost::property\_map< GraphContainer, vertex\_properties\_t >::const\_type **const\_vertex\_property\_map\_t**
- typedef graph\_traits::degree\_size\_type **degree\_t**
- typedef graph\_traits::edge\_iterator **edge\_iter**
- typedef boost::property\_map< GraphContainer, edge\_properties\_t >::type **edge\_property\_map\_t**
- typedef std::pair< edge\_iter, edge\_iter > **edge\_range\_t**
- typedef graph\_traits::edge\_descriptor **edge\_t**
- typedef QPair< [Edge](#), [Edge](#) > **EdgePair**
- typedef boost::graph\_traits< GraphContainer > [graph\\_traits](#)
- typedef boost::adjacency\_list< boost::vecS, boost::vecS, boost::bidirectionalS, boost::property< boost::vertex\_index\_t, int, boost::property< vertex\_properties\_t, VertexProperties > >, boost::property< edge\_properties\_t, EdgeProperties > > **GraphContainer**
- enum **GraphCopyFlags** { **CopyVertexProperties** = 1 << 0 , **CopyEdgeProperties** = 1 << 1 , **CopyAllProperties** = CopyVertexProperties | CopyEdgeProperties }
- typedef graph\_traits::in\_edge\_iterator **in\_edge\_iter**
- typedef boost::inv\_adjacency\_iterator\_generator< GraphContainer, vertex\_t, in\_edge\_iter >::type **inv\_adjacency\_iter**
- typedef std::pair< inv\_adjacency\_iter, inv\_adjacency\_iter > **inv\_adjacency\_vertex\_range\_t**
- typedef graph\_traits::out\_edge\_iterator **out\_edge\_iter**
- typedef std::pair< out\_edge\_iter, out\_edge\_iter > **out\_edge\_range\_t**
- enum **ReturnOrder** { **BreadthFirstOrder** , **DepthFirstOrder** }
- typedef boost::property\_map< GraphContainer, boost::vertex\_index\_t >::type **vertex\_index\_map\_t**
- typedef graph\_traits::vertex\_iterator **vertex\_iter**
- typedef boost::property\_map< GraphContainer, vertex\_properties\_t >::type **vertex\_property\_map\_t**
- typedef std::pair< vertex\_iter, vertex\_iter > **vertex\_range\_t**
- typedef graph\_traits::vertex\_descriptor **vertex\_t**
- typedef [QMapForAdaptors](#)< [Vertex](#), int > **VertexIntMap**
- typedef boost::associative\_property\_map< [VertexIntMap](#) > **VertexIntMapAdaptor**
- typedef QPair< [Vertex](#), [Vertex](#) > **VertexPair**
- typedef [QMapForAdaptors](#)< [Vertex](#), [Vertex](#) > **VertexVertexMap**
- typedef boost::associative\_property\_map< [VertexVertexMap](#) > **VertexVertexMapAdaptor**

### Static Public Member Functions inherited from [Digikam::Graph](#)< [VertexProperties](#), [EdgeProperties](#) >

- template<typename T >  
static bool **alwaysFalse** (const T &, const T &)

### Static Protected Member Functions inherited from [Digikam::Graph](#)< [VertexProperties](#), [EdgeProperties](#) >

- template<typename range\_t >  
static bool **isEmptyRange** (const range\_t &range)
- template<typename range\_t >  
static QList< [Edge](#) > **toEdgeList** (const range\_t &range)
- template<typename Value , typename range\_t >  
static QList< Value > **toList** (const range\_t &range)
- template<typename range\_t >  
static QList< [Vertex](#) > **toVertexList** (const range\_t &range)

**Protected Attributes inherited from****Digikam::Graph** < **VertexProperties**, **EdgeProperties** >

- [MeaningOfDirection](#) **direction** = [ParentToChild](#)
- [GraphContainer](#) **graph**

**6.912 Digikam::ItemHistoryGraphModel Class Reference**

Inheritance diagram for Digikam::ItemHistoryGraphModel:



## Public Types

- enum **ExtraRoles** {  
**IsImageItemRole** = Qt::UserRole + 1000 , **IsFilterActionItemRole** = Qt::UserRole + 1001 , **IsHeaderItemRole** = Qt::UserRole + 1002 , **IsCategoryItemRole** = Qt::UserRole + 1003 ,  
**IsSeparatorItemRole** = Qt::UserRole + 1004 , **IsSubjectImageRole** = Qt::UserRole + 1010 , **FilterActionRole** = Qt::UserRole + 1020 }
- enum **Mode** { **ImagesListMode** , **ImagesTreeMode** , **CombinedTreeMode** }

## Public Member Functions

- [FilterAction](#) **filterAction** (const QModelIndex &index) const
  - bool **hasImage** (const [ItemInfo](#) &info)
  - [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
  - DECLARE\_MODEL\_DRAG\_DROP\_METHODS [ItemListModel](#) \* [imageModel](#) () const
  - QModelIndex **imageModelIndex** (const QModelIndex &index) const
  - QModelIndex **indexForInfo** (const [ItemInfo](#) &info) const
  - bool **isFilterAction** (const QModelIndex &index) const
  - bool **isImage** (const QModelIndex &index) const
  - [ItemHistoryGraphModel](#) (QWidget \*const parent)
  - Mode **mode** () const
  - void **setHistory** (const [ItemInfo](#) &subject, const [ItemHistoryGraph](#) &graph=[ItemHistoryGraph](#)())
  - void **setMode** (Mode mode)
  - [ItemInfo](#) **subject** () const
- 
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override  
*QAbstractItemModel implementation.*
  - int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
  - int **columnCount** (const QModelIndex &parent=QModelIndex()) const override
  - Qt::ItemFlags **flags** (const QModelIndex &index) const override
  - bool **hasChildren** (const QModelIndex &parent=QModelIndex()) const override
  - QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
  - QModelIndex **parent** (const QModelIndex &index) const override
  - QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
  - bool **setData** (const QModelIndex &index, const QVariant &value, int role) override

## Public Member Functions inherited from [Digikam::DragDropModelImplementation](#)

- virtual Qt::ItemFlags [dragDropFlags](#) (const QModelIndex &index) const
- Qt::ItemFlags [dragDropFlagsV2](#) (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const
- [DragDropModelImplementation](#) ()=default
- bool **dropMimeData** (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool **isDragEnabled** (const QModelIndex &index) const
- virtual bool **isDropEnabled** (const QModelIndex &index) const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const
- QStringList **mimeTypes** () const
- void **setDragDropHandler** ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions [supportedDropActions](#) () const

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* `m_dragDropHandler` = nullptr

## 6.912.1 Member Function Documentation

### 6.912.1.1 `imageModel()`

```
ItemListModel * Digikam::ItemHistoryGraphModel::imageModel ( ) const
```

Returns an internal image model used for entries representing images. Note: Set a thumbnail thread on this model if you need thumbnails.

### 6.912.1.2 `imageModelIndex()`

```
QModelIndex Digikam::ItemHistoryGraphModel::imageModelIndex (
    const QModelIndex & index ) const
```

If the given index is represented by the internal image model, return the image model's index. Otherwise an invalid index is returned.

### 6.912.1.3 `indexForInfo()`

```
QModelIndex Digikam::ItemHistoryGraphModel::indexForInfo (
    const ItemInfo & info ) const
```

Note: There may be multiple indexes for an info. The index found first is returned.

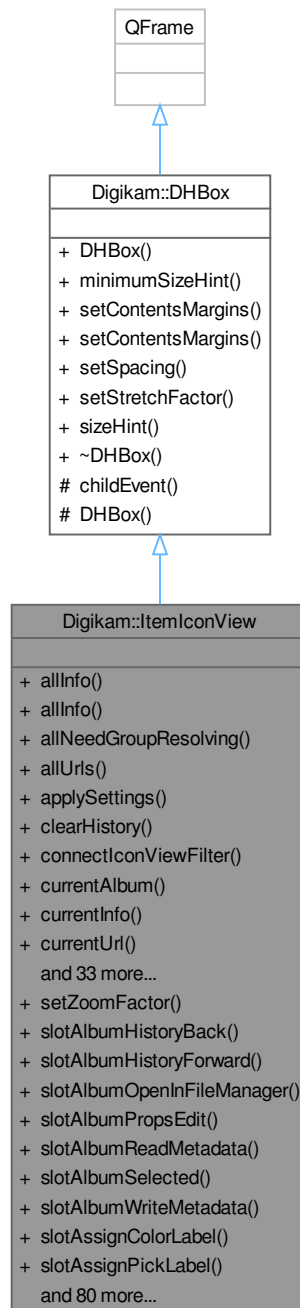
### 6.912.1.4 `setHistory()`

```
void Digikam::ItemHistoryGraphModel::setHistory (
    const ItemInfo & subject,
    const ItemHistoryGraph & graph = ItemHistoryGraph() )
```

Set the history subject and the history graph. Per default, the subject's history graph is read.

## 6.913 Digikam::ItemIconView Class Reference

Inheritance diagram for Digikam::ItemIconView:



### Classes

- class [Private](#)

## Public Slots

- void **setZoomFactor** (double zoom)
- void **slotAlbumHistoryBack** (int steps=1)
- void **slotAlbumHistoryForward** (int steps=1)
- void **slotAlbumOpenInFileManager** ()
- void **slotAlbumPropsEdit** ()
- void **slotAlbumReadMetadata** ()
- void **slotAlbumSelected** (const QList< Album \* > &albums)
- void **slotAlbumWriteMetadata** ()
- void **slotAssignColorLabel** (int colorId)
- void **slotAssignPickLabel** (int pickId)
- void **slotAssignRating** (int rating, bool toggle=true)
- void **slotAssignTag** ()
- void **slotAssignTag** (int tagID)
- void **slotCopySelectionTo** ()
- void **slotCreateGroupByFilenameFromSelection** ()
- void **slotCreateGroupByTimeFromSelection** ()
- void **slotCreateGroupByTimelapseFromSelection** ()
- void **slotCreateGroupFromSelection** ()
- void **slotDeleteAlbum** ()
- void **slotDeleteTag** ()
- void **slotEditor** ()
- void **slotEditTag** ()
- void **slotFileWithDefaultApplication** ()
- void **slotFitToWindow** ()
- void **slotFocusAndNextImage** ()
- void **slotGotoAlbumAndItem** (const ItemInfo &imageInfo)
- void **slotGotoDateAndItem** (const ItemInfo &imageInfo)
- void **slotGotoTagAndItem** (int tagID)
- void **slotIconView** ()
- void **slotImageAddToCurrentQueue** ()
- void **slotImageAddToExistingQueue** (int)
- void **slotImageAddToLightTable** ()
- void **slotImageAddToNewQueue** ()
- void **slotImageDelete** ()
- void **slotImageDeletePermanently** ()
- void **slotImageDeletePermanentlyDirectly** ()
- void **slotImageEdit** ()
  - Tools methods (Editor, BQM, Light Table) - itemiconview\_tools.cpp.*
- void **slotImageExifOrientation** (int orientation)
- void **slotImageFindSimilar** ()
- void **slotImageLightTable** ()
- void **slotImagePaste** ()
- void **slotImagePreview** ()
- void **slotImageQualitySorter** ()
  - Side-bars handling methods - itemiconview\_sidebars.cpp.*
- void **slotImageReadMetadata** ()
- void **slotImageRecognizeFaces** ()
- void **slotImageRemoveAllFaces** ()
- void **slotImageRename** ()
- void **slotImageScanForFaces** ()
- void **slotImageSeparationSortOrder** (int order)
- void **slotImageTrashDirectly** ()

- void **slotImageWriteMetadata** ()
- void **slotLeftSideBarActivate** (QWidget \*widget)
- void **slotLeftSideBarActivate** (SidebarWidget \*widget)
- void **slotLeftSideBarActivateAlbums** ()
- void **slotLeftSideBarActivateTags** ()
- void **slotLightTable** ()
- void **slotMapWidgetView** ()
- void **slotMoveSelectionToAlbum** ()
- void **slotNewAdvancedSearch** ()
- void **slotNewAlbum** ()
- void **slotNewDuplicatesSearch** (const QList< PAlbum \* > &albums={})
- void **slotNewDuplicatesSearch** (const QList< TAlbum \* > &albums)
- void **slotNewKeywordSearch** ()

*Search management methods - itemiconview\_search.cpp.*

- void **slotNewTag** ()
- void **slotNotificationError** (const QString &message, int type)
- void **slotOpenTagsManager** ()
- void **slotQueueMgr** ()
- void **slotRefresh** ()
- void **slotRemoveSelectedFromGroup** ()
- void **slotRemoveTag** (int tagID)
- void **slotRenameAlbum** ()
- void **slotRightSideBarActivateAssignedTags** ()
- void **slotRightSideBarActivateComments** ()
- void **slotRightSideBarActivateTitles** ()
- void **slotSelectAlbum** (const QUrl &url)
- void **slotSelectAll** ()
- void **slotSelectInvert** ()
- void **slotSelectNone** ()
- void **slotSeparateImages** (int mode)
- void **slotSetAsAlbumThumbnail** (const ItemInfo &info)
- void **slotSetCurrentUriWhenAvailable** (const QUrl &url)
- void **slotSetCurrentWhenAvailable** (const qlonglong id)
- void **slotSortAlbums** (int role)
- void **slotSortImages** (int order)
- void **slotSortImagesOrder** (int order)
- void **slotTableView** ()
- void **slotUngroupSelected** ()
- void **slotZoomIn** ()
- void **slotZoomOut** ()
- void **slotZoomTo100Percents** ()

## Signals

- void **signalAlbumSelected** (Album \*)
- void **signalChangedTab** (QWidget \*)
- void **signalFuzzySidebarActive** (bool active)
- void **signalGotoAlbumAndItem** (const ItemInfo &)
- void **signalGotoDateAndItem** (AlbumIconItem \*)
- void **signalGotoTagAndItem** (int tagID)
- void **signalImageSelected** (const ItemInfoList &selectedImage, const ItemInfoList &allImages)
- void **signalNoCurrentItem** ()
- void **signalSelectionChanged** (int numberOfSelectedItems)
- void **signalSeparationModeChanged** (int category)



- void **signalSwitchedToIconView** ()
- void **signalSwitchedToMapView** ()
- void **signalSwitchedToPreview** ()
- void **signalSwitchedToTableView** ()
- void **signalSwitchedToTrashView** ()
- void **signalThumbSizeChanged** (int)
- void **signalTrashSelectionChanged** (const QString &text)
- void **signalZoomChanged** (double)

## Public Member Functions

- [ItemInfoList](#) **allInfo** (const bool grouping=false) const
- [ItemInfoList](#) **allInfo** (const [OperationType](#) type) const
- bool **allNeedGroupResolving** (const [OperationType](#) type) const  
*Item Group methods - itemiconview\_groups.cpp.*
- QList< QUrl > **allUrls** (bool grouping=false) const
- void **applySettings** ()
- void **clearHistory** ()
- void **connectIconViewFilter** ([FilterStatusBar](#) \*const filter)
- [Album](#) \* **currentAlbum** () const  
*Album management methods - itemiconview\_album.cpp.*
- [ItemInfo](#) **currentInfo** () const
- QUrl **currentUrl** () const
- void **getBackwardHistory** (QStringList &titles)
- void **getForwardHistory** (QStringList &titles)
- bool **hasCurrentItem** () const
- void **hideSideBars** ()
- void **imageTransform** ([MetaEngineRotation::TransformationAction](#) transform)
- int **itemCount** () const  
*Items management methods - itemiconview\_items.cpp.*
- [ItemIconView](#) (QWidget \*const parent, [DModelFactory](#) \*const modelCollection)
- QList< [SidebarWidget](#) \* > **leftSidebarWidgets** () const
- void **nextLeftSideBarTab** ()
- void **nextRightSideBarTab** ()
- void **previousLeftSideBarTab** ()
- void **previousRightSideBarTab** ()
- void **refreshView** ()
- [ItemInfoList](#) **selectedInfoList** (const bool currentFirst=false, const bool grouping=false) const
- [ItemInfoList](#) **selectedInfoList** (const [OperationType](#) type, const bool currentFirst=false) const
- bool **selectedNeedGroupResolving** (const [OperationType](#) type) const
- QList< QUrl > **selectedUrls** (bool grouping=false) const
- QList< QUrl > **selectedUrls** (const [OperationType](#) type) const
- void **setAllGroupsOpen** (bool open)
- void **setRecurseAlbums** (bool recursive)
- void **setRecurseTags** (bool recursive)  
*Tags management methods - itemiconview\_tags.cpp.*
- void **setThumbSize** (int size)
- void **setToolsIconView** ([DCategorizedView](#) \*const view)  
*Views management methods - itemiconview\_views.cpp.*
- void **showSideBars** ()
- void **toggleFullScreen** (bool set)
- void **toggleLeftSidebar** ()
- void **toggleRightSidebar** ()

- void **toggleShowBar** (bool)
- void **toggleTag** (int tagID)
- StackedView::StackedViewMode **viewMode** () const
- double **zoomMax** () const
- double **zoomMin** () const

*Zoom management methods - itemiconview\_zoom.cpp.*

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.913.1 Member Function Documentation

### 6.913.1.1 allNeedGroupResolving()

```
bool Digikam::ItemIconView::allNeedGroupResolving (
    const OperationType type ) const
```

Query whether the operation to be performed on currently selected or all all items in the currently active view should be performed on all grouped items or just the first.

### 6.913.1.2 allUrls()

```
QList< QUrl > Digikam::ItemIconView::allUrls (
    bool grouping = false ) const
```

Get all items in the current view. Whether only the first or all grouped items are returned is determined as described above.

### 6.913.1.3 selectedUrls()

```
QList< QUrl > Digikam::ItemIconView::selectedUrls (
    bool grouping = false ) const
```

Get currently selected items. By default only the first images in groups are given, while all can be obtained by setting the grouping parameter to true. Given an operation, it will be determined from settings/user query whether only the first or all items in a group are returned. Ideally only the latter (giving an operation) is used.

### 6.913.1.4 slotFitToWindow

```
void Digikam::ItemIconView::slotFitToWindow ( ) [slot]
```

### 6.913.1.5 slotImageQualitySorter

```
void Digikam::ItemIconView::slotImageQualitySorter ( ) [slot]
```

Tools methods (Editor, BQM, Light Table) - itemiconview\_tools.cpp.

### 6.913.1.6 slotRemoveTag

```
void Digikam::ItemIconView::slotRemoveTag (
    int tagID ) [slot]
```

Implementation for Automatic Icon Removal of Confirmed Tags. QTimer to ensure TagRemoval is complete.

If the face just removed was the final face associated with that Tag, reset Tag Icon.

## 6.914 Digikam::ItemIconView::Private Class Reference

### Public Member Functions

- void **addPageUpDownActions** (const [ItemIconView](#) \*const q, QWidget \*const w)
- QString **userPresentableAlbumTitle** (const QString &title) const

### Public Attributes

- [AlbumFolderViewSideBarWidget](#) \* **albumFolderSideBar** = nullptr
- [AlbumHistory](#) \* **albumHistory** = nullptr
- [AlbumManager](#) \* **albumManager** = nullptr
- [AlbumModificationHelper](#) \* **albumModificationHelper** = nullptr
- [DateFolderViewSideBarWidget](#) \* **dateViewSideBar** = nullptr
- QMainWindow \* **dockArea** = nullptr
- [DNotificationWidget](#) \* **errorWidget** = nullptr
- [FilterSideBarWidget](#) \* **filterWidget** = nullptr
- [FuzzySearchSideBarWidget](#) \* **fuzzySearchSideBar** = nullptr
- [DigikamItemView](#) \* **iconView** = nullptr
- int **initialAlbumID** = 0
- [AlbumLabelsSearchHandler](#) \* **labelsSearchHandler** = nullptr
- [LabelsSideBarWidget](#) \* **labelsSideBar** = nullptr
- StackedView::StackedViewMode **lastViewMode** = StackedView::IconViewMode
- [Sidebar](#) \* **leftSideBar** = nullptr
- QList< [SidebarWidget](#) \* > **leftSideBarWidgets**
- [DModelFactory](#) \* **modelCollection** = nullptr
- QTimer \* **msgNotifyTimer** = nullptr
- [DigikamApp](#) \* **parent** = nullptr
- [PeopleSideBarWidget](#) \* **peopleSideBar** = nullptr
- [ItemPropertiesSideBarDB](#) \* **rightSideBar** = nullptr

- [SearchModificationHelper](#) \* **searchModificationHelper** = nullptr
- [SearchSideBarWidget](#) \* **searchSideBar** = nullptr
- QTimer \* **selectionTimer** = nullptr
- [SidebarSplitter](#) \* **splitter** = nullptr
- [StackedView](#) \* **stackedView** = nullptr
- [TableView](#) \* **tableView** = nullptr
- [TagModificationHelper](#) \* **tagModificationHelper** = nullptr
- [TagViewSideBarWidget](#) \* **tagViewSideBar** = nullptr
- int **thumbSize** = ThumbnailSize::Medium
- QTimer \* **thumbSizeTimer** = nullptr
- [TimelineSideBarWidget](#) \* **timelineSideBar** = nullptr
- [TrashView](#) \* **trashView** = nullptr
- bool **useAlbumHistory** = false
- [ItemViewUtilities](#) \* **utilities** = nullptr

## 6.915 Digikam::ItemInfo Class Reference

The [ItemInfo](#) class contains provides access to the database for a single image. The properties can be read and written. Information will be cached.

### Public Types

- typedef [DatabaseFields::Hash](#)< QVariant > **DatabaseFieldsHashRaw**

### Public Member Functions

- [ItemInfo](#) *copyItem* (int dstAlbumID, const QString &dstFileName)
- bool *isLocationAvailable* () const
- [ItemInfo](#) ()
- [ItemInfo](#) (const [ItemInfo](#) &info)
- [ItemInfo](#) (const [ItemListerRecord](#) &record)
- [ItemInfo](#) (qulonglong ID)
- bool **operator!=** (const [ItemInfo](#) &info) const
- bool **operator<** (const [ItemInfo](#) &info) const
- [ItemInfo](#) & **operator=** (const [ItemInfo](#) &info)
- bool **operator==** (const [ItemInfo](#) &info) const
- [~ItemInfo](#) ()

### Operations with Properties

- bool *isNull* () const
- QString *name* () const
- QDateTime *dateTime* () const
- QDateTime *modDateTime* () const
- qulonglong *fileSize* () const
- QSize *dimensions* () const
- QUrl *fileUrl* () const
- QString *filePath* () const
- QString *relativePath* () const
- qulonglong *id* () const
- int *albumId* () const
- int *albumRootId* () const

- double [aspectRatio](#) () const
- qlonglong [manualOrder](#) () const
- DatabaseItem::Category [category](#) () const
- QString [format](#) () const
- bool [isVisible](#) () const
- bool [isRemoved](#) () const
- int [orientation](#) () const
- QString [title](#) () const
- QString [comment](#) () const
- int [faceCount](#) () const
- int [unconfirmedFaceCount](#) () const
- QMap< QString, QString > [getSuggestedNames](#) () const
- void [setName](#) (const QString &newName)
- void [setDateTime](#) (const QDateTime &dateTime)
- void [setModDateTime](#) (const QDateTime &dateTime)
- void [setManualOrder](#) (qlonglong value)
- void [setOrientation](#) (int value)
- void [setVisible](#) (bool isVisible)
- DatabaseFieldsHashRaw [getDatabaseFieldsRaw](#) (const DatabaseFields::Set &requestedSet) const
- QVariant [getDatabaseFieldRaw](#) (const DatabaseFields::Set &requestedField) const

### Operations with Geolocation

- [ItemPosition](#) [imagePosition](#) () const
- double [longitudeNumber](#) () const
- double [latitudeNumber](#) () const
- double [altitudeNumber](#) () const
- bool [hasCoordinates](#) () const
- bool [hasAltitude](#) () const

### Operations with History

- [DImageHistory](#) [imageHistory](#) () const
- void [setItemHistory](#) (const [DImageHistory](#) &history)
- bool [hasImageHistory](#) () const
- QString [uuid](#) () const
- void [setUuid](#) (const QString &uuid)
- [HistoryImageId](#) [historyImageId](#) () const
- bool [hasDerivedImages](#) () const
- bool [hasAncestorImages](#) () const
- QList< [ItemInfo](#) > [derivedImages](#) () const
- QList< [ItemInfo](#) > [ancestorImages](#) () const
- QList< QPair< qlonglong, qlonglong > > [relationCloud](#) () const
- void [markDerivedFrom](#) (const [ItemInfo](#) &ancestorImage)

### Operations with Groups

- bool [isGrouped](#) () const
- bool [hasGroupedImages](#) () const
- int [numberOfGroupedImages](#) () const
- [ItemInfo](#) [groupImage](#) () const
- qlonglong [groupImageId](#) () const
- QList< [ItemInfo](#) > [groupedImages](#) () const
- void [addToGroup](#) (const [ItemInfo](#) &info)
- void [removeFromGroup](#) ()
- void [clearGroup](#) ()

### Operations with Containers

- [ImageCommonContainer](#) [imageCommonContainer](#) () const
- [ImageMetadataContainer](#) [imageMetadataContainer](#) () const

- [VideoMetadataContainer](#) **videoMetadataContainer** () const
- [PhotoInfoContainer](#) **photoInfoContainer** () const
- [VideoInfoContainer](#) **videoInfoContainer** () const
- [Template](#) **metadataTemplate** () const
- void [setMetadataTemplate](#) (const [Template](#) &t)
- void [removeMetadataTemplate](#) ()
- [ItemComments](#) **imageComments** (const [CoreDbAccess](#) &access) const
- [ItemCopyright](#) **imageCopyright** () const
- [ItemExtendedProperties](#) **imageExtendedProperties** () const

### Operations with Labels

- int [pickLabel](#) () const
- int [colorLabel](#) () const
- int [rating](#) () const
- void [setPickLabel](#) (int value)
- void [setColorLabel](#) (int value)
- void [setRating](#) (int value)

### Static Public Member Functions

- static [ItemInfo](#) **fromLocalFile** (const QString &path)
- static [ItemInfo](#) **fromLocationAlbumAndName** (int locationId, const QString &album, const QString &name)
- static [ItemInfo](#) **fromUrl** (const QUrl &url)

### Operations with Similarity

- class **ItemInfoCache**
- class **ItemInfoList**
- double **similarityTo** (const qlonglong imageId) const
- double **currentSimilarity** () const
- qlonglong **currentReferenceImage** () const
- size\_t **hash** () const
- QList< [ItemInfo](#) > **fromUniqueHash** (const QString &uniqueHash, qlonglong fileSize)
- QString **uniqueHash** () const

### Operations with Tags

- void [setTag](#) (int tagId)
- void [addTagPaths](#) (const QStringList &tagPaths)
- void [removeTag](#) (int tagId)
- void [removeAllTags](#) ()
- [ItemTagPair](#) **imageTagPair** (int tagId) const
- QList< [ItemTagPair](#) > **availableItemTagPairs** () const
- QList< int > **tagIds** () const

### Operations with Thumbnails

- [ThumbnailIdentifier](#) **thumbnailIdentifier** () const
- [ThumbnailInfo](#) **thumbnailInfo** () const
- static [ThumbnailIdentifier](#) **thumbnailIdentifier** (qlonglong id)

## 6.915.1 Detailed Description

### Note

access rules for all methods in this class: [ItemInfoData](#) members shall be accessed only under [CoreDbAccess](#) lock. The id and albumId are the exception to this rule, as they are primitive and will never change during the lifetime of an object.

## 6.915.2 Constructor & Destructor Documentation

### 6.915.2.1 ItemInfo() [1/4]

```
Digikam::ItemInfo::ItemInfo ( )
```

Constructor Creates a null image info

### 6.915.2.2 ItemInfo() [2/4]

```
Digikam::ItemInfo::ItemInfo (
    qlonglong ID ) [explicit]
```

Constructor. Creates an [ItemInfo](#) object without any cached data initially.

#### Parameters

<i>ID</i>	the unique ID for this image
-----------	------------------------------

### 6.915.2.3 ItemInfo() [3/4]

```
Digikam::ItemInfo::ItemInfo (
    const ItemListerRecord & record ) [explicit]
```

Constructor. Creates an [ItemInfo](#) object where the provided information will initially be available cached, without database access.

### 6.915.2.4 ItemInfo() [4/4]

```
Digikam::ItemInfo::ItemInfo (
    const ItemInfo & info )
```

Copy constructor.

### 6.915.2.5 ~ItemInfo()

```
Digikam::ItemInfo::~ItemInfo ( )
```

Destructor

## 6.915.3 Member Function Documentation

### 6.915.3.1 addTagPaths()

```
void Digikam::ItemInfo::addTagPaths (
    const QStringList & tagPaths )
```

Adds tags in the list to the item. Tags are created if they do not yet exist

### 6.915.3.2 addToGroup()

```
void Digikam::ItemInfo::addToGroup (
    const ItemInfo & info )
```

Group this image behind the given image

### 6.915.3.3 albumId()

```
int Digikam::ItemInfo::albumId ( ) const
```

#### Returns

the id of the [PAlbum](#) to which this item belongs

### 6.915.3.4 albumRootId()

```
int Digikam::ItemInfo::albumRootId ( ) const
```

The album root id

### 6.915.3.5 aspectRatio()

```
double Digikam::ItemInfo::aspectRatio ( ) const
```

#### Returns

the id of the Aspect Ratio for this item

### 6.915.3.6 category()

```
DatabaseItem::Category Digikam::ItemInfo::category ( ) const
```

Returns the category of the item: Image, Audio, Video



### 6.915.3.7 clearGroup()

```
void Digikam::ItemInfo::clearGroup ( )
```

This image [hasGroupedImages\(\)](#): Split up the group, remove all [groupedImages\(\)](#) from this image's group.

### 6.915.3.8 colorLabel()

```
int Digikam::ItemInfo::colorLabel ( ) const
```

Returns the Color Label Id (see ColorLabel values in globals.h)

### 6.915.3.9 comment()

```
QString Digikam::ItemInfo::comment ( ) const
```

#### Returns

the default comment for this item

### 6.915.3.10 copyItem()

```
ItemInfo Digikam::ItemInfo::copyItem (
    int dstAlbumID,
    const QString & dstFileName )
```

Copy database information of this item to a newly created item

#### Parameters

<i>dstAlbumID</i>	destination album id
<i>dstFileName</i>	new filename

#### Returns

an [ItemInfo](#) object of the new item

### 6.915.3.11 currentReferenceImage()

```
qulonglong Digikam::ItemInfo::currentReferenceImage ( ) const
```

Returns the id of the current fuzzy search reference image.

### 6.915.3.12 dateTime()

```
QDateTime Digikam::ItemInfo::dateTime ( ) const
```

#### Returns

the datetime of the image

### 6.915.3.13 dimensions()

```
QSize Digikam::ItemInfo::dimensions ( ) const
```

#### Returns

the dimensions of the image (valid only if dimensions have been requested)

### 6.915.3.14 faceCount()

```
int Digikam::ItemInfo::faceCount ( ) const
```

#### Returns

the number of Faces in this item.

### 6.915.3.15 filePath()

```
QString Digikam::ItemInfo::filePath ( ) const
```

Returns the file path to the image

### 6.915.3.16 fileSize()

```
qulonglong Digikam::ItemInfo::fileSize ( ) const
```

#### Returns

the filesize of the image

### 6.915.3.17 fileUrl()

```
QUrl Digikam::ItemInfo::fileUrl ( ) const
```

Returns the [file://](#) url. This is equivalent to `QUrl::fromLocalFile(filePath())`

### 6.915.3.18 format()

```
QString Digikam::ItemInfo::format ( ) const
```

Returns the image format / mimetype as a standardized string (see project/documents/DBSCHEMA.ODS).

### 6.915.3.19 fromLocalFile()

```
ItemInfo Digikam::ItemInfo::fromLocalFile (
    const QString & path ) [static]
```

Creates an [ItemInfo](#) object from a file url.

### 6.915.3.20 fromLocationAlbumAndName()

```
ItemInfo Digikam::ItemInfo::fromLocationAlbumAndName (
    int locationId,
    const QString & album,
    const QString & name ) [static]
```

Create an [ItemInfo](#) object from the given combination, which must be cleaned and corresponding to the values in the database

### 6.915.3.21 fromUniqueHash()

```
QList< ItemInfo > Digikam::ItemInfo::fromUniqueHash (
    const QString & uniqueHash,
    qlonglong fileSize )
```

Scans the database for items with the given signature.

### 6.915.3.22 getDatabaseFieldsRaw()

```
ItemInfo::DatabaseFieldsHashRaw Digikam::ItemInfo::getDatabaseFieldsRaw (
    const DatabaseFields::Set & requestedSet ) const
```

### 6.915.3.23 getSuggestedNames()

```
QMap< QString, QString > Digikam::ItemInfo::getSuggestedNames ( ) const
```

#### Returns

the map of Tag Region (in XML form) to Suggested Names for all Faces in the Image. Used to categorize images based on Face Suggestions.

#### 6.915.3.24 groupedImages()

```
QList< ItemInfo > Digikam::ItemInfo::groupedImages ( ) const
```

Returns the list of images grouped behind this image (not including this image itself) and an empty list if there is none.

#### 6.915.3.25 groupImage()

```
ItemInfo Digikam::ItemInfo::groupImage ( ) const
```

Returns the leading image of the group. Returns a null image if this image is not grouped ([isGrouped\(\)](#))

#### 6.915.3.26 hasDerivedImages()

```
bool Digikam::ItemInfo::hasDerivedImages ( ) const
```

Retrieve information about images from which this image is derived ([ancestorImages](#)) and images that have been derived from this images ([derivedImages](#)).

#### 6.915.3.27 hasGroupedImages()

```
bool Digikam::ItemInfo::hasGroupedImages ( ) const
```

The image is the leading image of a group, there are other images grouped behind this one.

#### 6.915.3.28 hash()

```
size_t Digikam::ItemInfo::hash ( ) const
```

Return a signature for the item.

#### 6.915.3.29 historyImageId()

```
HistoryImageId Digikam::ItemInfo::historyImageId ( ) const
```

Constructs a [HistoryImageId](#) with all available information for this image.

#### 6.915.3.30 id()

```
qulonglong Digikam::ItemInfo::id ( ) const
```

##### Returns

the unique image id for this item

### 6.915.3.31 imageComments()

```
ItemComments Digikam::ItemInfo::imageComments (
    const CoreDbAccess & access ) const
```

Retrieve the [ItemComments](#) object for this item. This object allows full read and write access to all comments and their properties. You need to hold [CoreDbAccess](#) to ensure the validity. For simple, cached read access see [comment\(\)](#).

### 6.915.3.32 imageCommonContainer()

```
ImageCommonContainer Digikam::ItemInfo::imageCommonContainer ( ) const
```

Retrieve information about the image, in form of numbers and user presentable strings, for certain defined fields of information (see [databaseinfocontainers.h](#))

### 6.915.3.33 imageCopyright()

```
ItemCopyright Digikam::ItemInfo::imageCopyright ( ) const
```

Retrieve the [ItemCopyright](#) object for this item. This object allows full read and write access to all copyright values.

### 6.915.3.34 imageExtendedProperties()

```
ItemExtendedProperties Digikam::ItemInfo::imageExtendedProperties ( ) const
```

Retrieve the [ItemExtendedProperties](#) object for this item. This object allows full read and write access to all extended properties values.

### 6.915.3.35 imageHistory()

```
DImageHistory Digikam::ItemInfo::imageHistory ( ) const
```

Retrieves and sets the image history from the database. Note: The image history retrieved here does typically include all steps from the original to this image, but does not reference this image itself.

### 6.915.3.36 imagePosition()

```
ItemPosition Digikam::ItemInfo::imagePosition ( ) const
```

Retrieve the [ItemPosition](#) object for this item.

### 6.915.3.37 imageTagPair()

```
ItemTagPair Digikam::ItemInfo::imageTagPair (
    int tagId ) const
```

Retrieve an [ItemTagPair](#) object for a single tag, or for all image/tag pairs for which properties are available (not necessarily the assigned tags)

**6.915.3.38 isGrouped()**

```
bool Digikam::ItemInfo::isGrouped ( ) const
```

The image is grouped in the group of another (leading) image.

**6.915.3.39 isLocationAvailable()**

```
bool Digikam::ItemInfo::isLocationAvailable ( ) const
```

Returns true if this is a valid [ItemInfo](#), and the location of the image is currently available (information freshly obtained from [CollectionManager](#))

**6.915.3.40 isNull()**

```
bool Digikam::ItemInfo::isNull ( ) const
```

Returns if this objects contains valid data

**6.915.3.41 isRemoved()**

```
bool Digikam::ItemInfo::isRemoved ( ) const
```

Returns true if the corresponding file was not deleted.

**6.915.3.42 isVisible()**

```
bool Digikam::ItemInfo::isVisible ( ) const
```

Returns true if the image is marked as visible in the database.

**6.915.3.43 longitudeNumber()**

```
double Digikam::ItemInfo::longitudeNumber ( ) const
```

Retrieves the coordinates and the altitude. Returns 0 if [hasCoordinates\(\)](#), or [hasAltitude](#) resp, is false.

**6.915.3.44 manualOrder()**

```
qlonglong Digikam::ItemInfo::manualOrder ( ) const
```

Returns the manual sort order

### 6.915.3.45 markDerivedFrom()

```
void Digikam::ItemInfo::markDerivedFrom (
    const ItemInfo & ancestorImage )
```

Add a relation to the database: This image is derived from the ancestorImage.

### 6.915.3.46 metadataTemplate()

```
Template Digikam::ItemInfo::metadataTemplate ( ) const
```

Retrieve metadata template information about the image.

### 6.915.3.47 modDateTime()

```
QDateTime Digikam::ItemInfo::modDateTime ( ) const
```

#### Returns

the modification datetime of the image

### 6.915.3.48 name()

```
QString Digikam::ItemInfo::name ( ) const
```

#### Returns

the name of the image

### 6.915.3.49 orientation()

```
int Digikam::ItemInfo::orientation ( ) const
```

Returns the orientation of the image, ([MetaEngine::ImageOrientation](#), EXIF standard)

### 6.915.3.50 pickLabel()

```
int Digikam::ItemInfo::pickLabel ( ) const
```

Returns the Pick Label Id (see PickLabel values in globals.h)

### 6.915.3.51 rating()

```
int Digikam::ItemInfo::rating ( ) const
```

Returns the rating

**6.915.3.52 relationCloud()**

```
QList< QPair< qlonglong, qlonglong > > Digikam::ItemInfo::relationCloud ( ) const
```

Returns the cloud of all directly or indirectly related images, derived images or ancestors, in from of "a derived from b" pairs.

**6.915.3.53 relativePath()**

```
QString Digikam::ItemInfo::relativePath ( ) const
```

Returns the relative path part to the image

**6.915.3.54 removeAllTags()**

```
void Digikam::ItemInfo::removeAllTags ( )
```

Remove all tags from the item (removes it from database)

**6.915.3.55 removeFromGroup()**

```
void Digikam::ItemInfo::removeFromGroup ( )
```

This image is grouped behind another image: Remove this image from its group

**6.915.3.56 removeMetadataTemplate()**

```
void Digikam::ItemInfo::removeMetadataTemplate ( )
```

Remove all template info about the image from database.

**6.915.3.57 removeTag()**

```
void Digikam::ItemInfo::removeTag (
    int tagID )
```

Remove a tag from the item (removes it from database)

**Parameters**

<i>tagID</i>	the ID of the tag to remove
--------------	-----------------------------

**6.915.3.58 setColorLabel()**

```
void Digikam::ItemInfo::setColorLabel (
```



```
int value )
```

Set the color Label Id for the item (see ColorLabel values from globals.h)

### 6.915.3.59 setDateTime()

```
void Digikam::ItemInfo::setDateTime (
    const QDateTime & dateTime )
```

Set the date and time (write it to database)

#### Parameters

<i>dateTime</i>	the new date and time.
-----------------	------------------------

### 6.915.3.60 setManualOrder()

```
void Digikam::ItemInfo::setManualOrder (
    qulonglong value )
```

Set the manual sorting order for the item

### 6.915.3.61 setMetadataTemplate()

```
void Digikam::ItemInfo::setMetadataTemplate (
    const Template & t )
```

Set metadata template information (write it to database)

#### Parameters

<i>t</i>	the new template data.
----------	------------------------

### 6.915.3.62 setModDateTime()

```
void Digikam::ItemInfo::setModDateTime (
    const QDateTime & dateTime )
```

Set the modification date and time (write it to database)

#### Parameters

<i>dateTime</i>	the new modification date and time.
-----------------	-------------------------------------

**6.915.3.63 setName()**

```
void Digikam::ItemInfo::setName (
    const QString & newName )
```

Set the name (write it to database)

**Parameters**

<i>newName</i>	the new name.
----------------	---------------

**6.915.3.64 setOrientation()**

```
void Digikam::ItemInfo::setOrientation (
    int value )
```

Set the orientation for the item

**6.915.3.65 setPickLabel()**

```
void Digikam::ItemInfo::setPickLabel (
    int value )
```

Set the pick Label Id for the item (see PickLabel values from globals.h)

**6.915.3.66 setRating()**

```
void Digikam::ItemInfo::setRating (
    int value )
```

Set the rating for the item

**6.915.3.67 setTag()**

```
void Digikam::ItemInfo::setTag (
    int tagID )
```

Adds a tag to the item (writes it to database)

**Parameters**

<i>tagID</i>	the ID of the tag to add
--------------	--------------------------

**6.915.3.68 setVisible()**

```
void Digikam::ItemInfo::setVisible (
```

```
bool isVisible )
```

Set the visibility flag - triggers between Visible and Hidden

### 6.915.3.69 tagIds()

```
QList< int > Digikam::ItemInfo::tagIds ( ) const
```

#### Returns

a list of IDs of tags assigned to this item

#### See also

[tagNames](#)

[tagPaths](#)

[Album::id\(\)](#)

### 6.915.3.70 thumbnailIdentifier()

```
ThumbnailIdentifier Digikam::ItemInfo::thumbnailIdentifier ( ) const
```

Fills a [ThumbnailIdentifier](#) / [ThumbnailInfo](#) from this [ItemInfo](#)

### 6.915.3.71 title()

```
QString Digikam::ItemInfo::title ( ) const
```

#### Returns

the default title for this item

### 6.915.3.72 unconfirmedFaceCount()

```
int Digikam::ItemInfo::unconfirmedFaceCount ( ) const
```

#### Returns

the number of Unconfirmed Faces in this item.

### 6.915.3.73 uniqueHash()

```
QString Digikam::ItemInfo::uniqueHash ( ) const
```

#### Returns

the unique hash signature as string of the image.

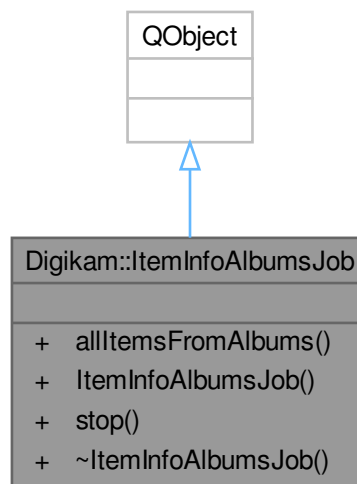
### 6.915.3.74 uuid()

```
QString Digikam::ItemInfo::uuid ( ) const
```

Retrieves and sets this' images UUID

## 6.916 Digikam::ItemInfoAlbumsJob Class Reference

Inheritance diagram for Digikam::ItemInfoAlbumsJob:



### Signals

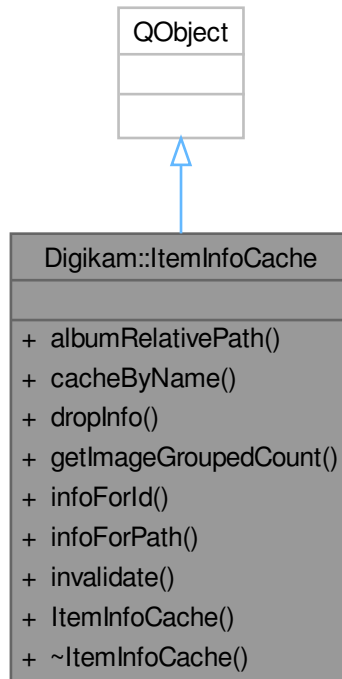
- void **signalCompleted** (const [ItemInfoList](#) &items)

### Public Member Functions

- void **allItemsFromAlbums** (const AlbumList &albumsList)
- **ItemInfoAlbumsJob** (QObject \*const parent=nullptr)
- void **stop** ()

## 6.917 Digikam::ItemInfoCache Class Reference

Inheritance diagram for Digikam::ItemInfoCache:



### Public Member Functions

- `QString albumRelativePath (int albumId)`
- `void cacheByName (const QExplicitlySharedDataPointer< ItemInfoData > &infoPtr)`
- `void dropInfo (const QExplicitlySharedDataPointer< ItemInfoData > &infoPtr)`
- `int getImageGroupedCount (qulonglong id)`
- `QExplicitlySharedDataPointer< ItemInfoData > infoForId (qulonglong id)`
- `QExplicitlySharedDataPointer< ItemInfoData > infoForPath (int albumRootId, const QString &relativePath, const QString &name)`
- `void invalidate ()`

### 6.917.1 Member Function Documentation

#### 6.917.1.1 albumRelativePath()

```

QString Digikam::ItemInfoCache::albumRelativePath (
    int albumId )
  
```

Returns the cached relativePath for the given album id.

### 6.917.1.2 cacheByName()

```
void Digikam::ItemInfoCache::cacheByName (
    const QExplicitlySharedDataPointer< ItemInfoData > & infoPtr )
```

Call this to put data in the hash by file name if you have newly created data and the name is filled. Call under write lock.

### 6.917.1.3 dropInfo()

```
void Digikam::ItemInfoCache::dropInfo (
    const QExplicitlySharedDataPointer< ItemInfoData > & infoPtr )
```

Call this when the data has been dereferenced, before deletion.

### 6.917.1.4 getImageGroupedCount()

```
int Digikam::ItemInfoCache::getImageGroupedCount (
    qlonglong id )
```

Returns the cached grouped count for the given image id.

### 6.917.1.5 infoForId()

```
QExplicitlySharedDataPointer< ItemInfoData > Digikam::ItemInfoCache::infoForId (
    qlonglong id )
```

Return an [ItemInfoData](#) object for the given image id. A new object is created, or an existing object is returned. If a new object is created, the id field will be initialized.

### 6.917.1.6 infoForPath()

```
QExplicitlySharedDataPointer< ItemInfoData > Digikam::ItemInfoCache::infoForPath (
    int albumRootId,
    const QString & relativePath,
    const QString & name )
```

Return an [ItemInfoData](#) object for the given album root, relativePath and file name triple. Works if previously cached with [cacheByName](#). Returns 0 if not found.

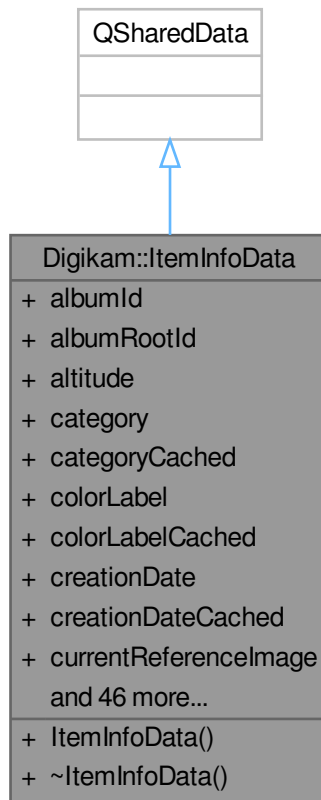
### 6.917.1.7 invalidate()

```
void Digikam::ItemInfoCache::invalidate ( )
```

Invalidate the cache and all its cached data

## 6.918 Digikam::ItemInfoData Class Reference

Inheritance diagram for Digikam::ItemInfoData:



### Public Types

- typedef `DatabaseFields::Hash` < QVariant > `DatabaseFieldsHashRaw`

### Public Attributes

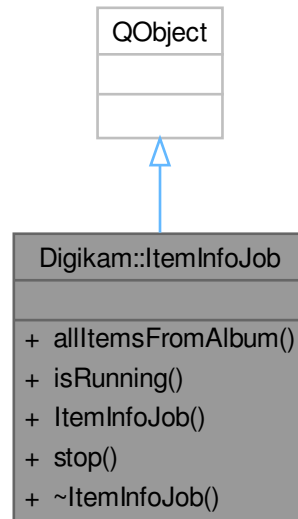
- int `albumId` = -1
- int `albumRootId` = -1
- double `altitude` = 0
- `DatabaseItem::Category` `category` = `DatabaseItem::UndefinedCategory`
- bool `categoryCached` = false
- quint8 `colorLabel` = `NoColorLabel`
- bool `colorLabelCached` = false
- `QDateTime` `creationDate`
- bool `creationDateCached` = false
- qlonglong `currentReferenceImage` = -1
- double `currentSimilarity` = 0.0

- [DatabaseFieldsHashRaw](#) **databaseFieldsHashRaw**
- QString **defaultComment**
- bool **defaultCommentCached** = false
- QString **defaultTitle**
- bool **defaultTitleCached** = false
- int **faceCount** = 0
- bool **faceCountCached** = false
- QMap< QString, QString > **faceSuggestions**
- bool **faceSuggestionsCached** = false
- qlonglong **fileSize** = 0
- bool **fileSizeCached** = false
- QString **format**
- bool **formatCached** = false
- qlonglong **groupImage** = -1
  - group leader, if the image is grouped*
- bool **groupImageCached** = false
- bool **hasAltitude** = false
- bool **hasCoordinates** = false
- bool **hasImageMetadata** = true
- bool **hasVideoMetadata** = true
- qlonglong **id** = -1
- DatabaseFields::ImageMetadataMinSizeType **imageMetadataCached** = DatabaseFields::ImageMetadata↔None
- QSize **imageSize**
- bool **imageSizeCached** = false
- bool **invalid** = false
- double **latitude** = 0
- double **longitude** = 0
- qlonglong **manualOrder** = 0
- bool **manualOrderCached** = false
- QDateTime **modificationDate**
- bool **modificationDateCached** = false
- QString **name**
- int **orientation** = 0
- bool **orientationCached** = false
- quint8 **pickLabel** = NoPickLabel
- bool **pickLabelCached** = false
- bool **positionsCached** = false
- quint8 **rating** = -1
- bool **ratingCached** = false
- QList< int > **tagIds**
- bool **tagIdsCached** = false
- int **unconfirmedFaceCount** = 0
- bool **unconfirmedFaceCountCached** = false
- QString **uniqueHash**
- bool **uniqueHashCached** = false
- DatabaseFields::VideoMetadataMinSizeType **videoMetadataCached** = DatabaseFields::VideoMetadata↔None



## 6.919 Digikam::ItemInfoJob Class Reference

Inheritance diagram for Digikam::ItemInfoJob:



### Signals

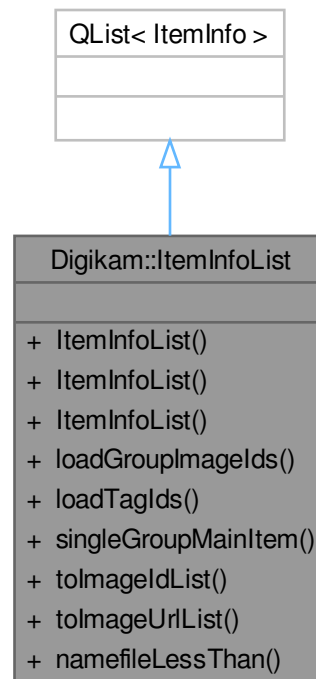
- void **signalCompleted** ()
- void **signalItemsInfo** (const [ItemInfoList](#) &items)

### Public Member Functions

- void **allItemsFromAlbum** ([Album](#) \*const album)
- bool **isRunning** () const
- void **stop** ()

## 6.920 Digikam::ItemInfoList Class Reference

Inheritance diagram for Digikam::ItemInfoList:



### Public Member Functions

- **ItemInfoList** (const QList< [ItemInfo](#) > &list)
- **ItemInfoList** (const QList< qlonglong > &idList)
- void **loadGroupImagelds** () const
- void **loadTagIds** () const
- [ItemInfo](#) **singleGroupMainItem** () const  
*singleGroupMainItem*
- QList< qlonglong > **tolmageldList** () const
- QList< QUrl > **tolmageUrlList** () const

### Static Public Member Functions

- static bool **namefileLessThan** (const [ItemInfo](#) &d1, const [ItemInfo](#) &d2)

## 6.920.1 Member Function Documentation

### 6.920.1.1 singleGroupMainItem()

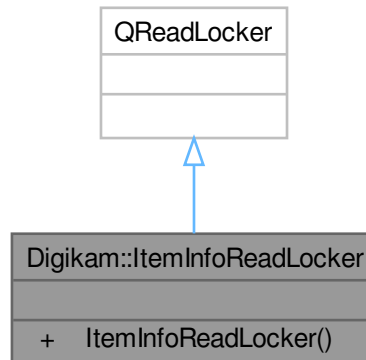
[ItemInfo](#) Digikam::ItemInfoList::singleGroupMainItem ( ) const

#### Returns

If the list contains of items of only one group including the main item, this main item is returned, otherwise a null [ItemInfo](#).

## 6.921 Digikam::ItemInfoReadLocker Class Reference

Inheritance diagram for Digikam::ItemInfoReadLocker:



## 6.922 Digikam::ItemInfoSet Class Reference

### Public Member Functions

- `ItemInfoSet` (int id, const [ItemInfo](#) &inf)

### Public Attributes

- `ItemInfo info`
- int `queueId` = 0

### 6.922.1 Detailed Description

A container of associated [ItemInfo](#) and queue id.

## 6.923 Digikam::ItemInfoStatic Class Reference

### Static Public Member Functions

- static `ItemInfoCache * cache` ()
- static void `create` ()
- static void `destroy` ()

**Public Attributes**

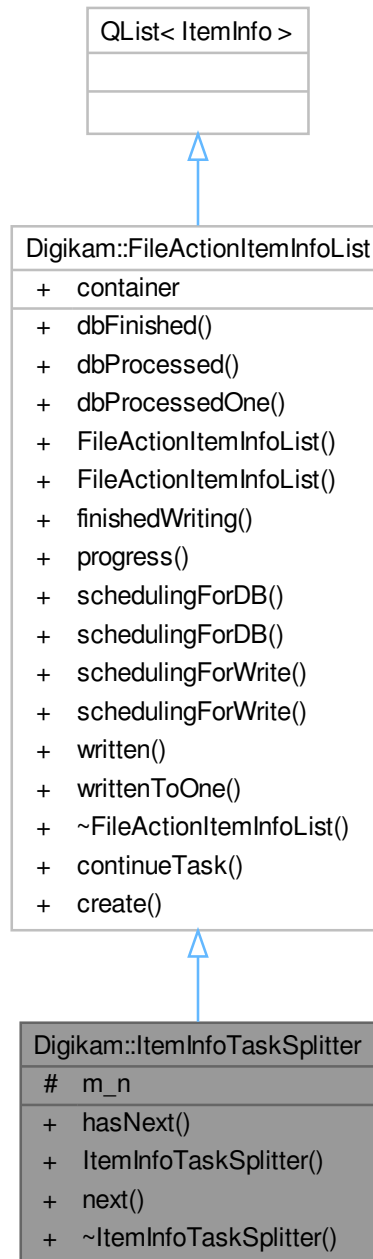
- [ItemInfoCache](#) **m\_cache**
- [QReadWriteLock](#) **m\_lock**

**Static Public Attributes**

- static [ItemInfoStatic](#) \* **m\_instance** = nullptr

## 6.924 Digikam::ItemInfoTaskSplitter Class Reference

Inheritance diagram for Digikam::ItemInfoTaskSplitter:



### Public Member Functions

- bool **hasNext** () const
- **ItemInfoTaskSplitter** (const [FileActionItemInfoList](#) &list)
- [FileActionItemInfoList](#) **next** ()

## Public Member Functions inherited from [Digikam::FileActionItemInfoList](#)

- void **dbFinished** () const
- void **dbProcessed** (int numberOfInfos) const
- void **dbProcessedOne** () const
  - db worker progress info*
- **FileActionItemInfoList** (const [FileActionItemInfoList](#) &copy)
- void **finishedWriting** () const
- [FileActionProgressItemContainer](#) \* **progress** () const
- void **schedulingForDB** (const QString &action, [FileActionProgressItemCreator](#) \*const creator)
- void **schedulingForDB** (int numberOfInfos, const QString &action, [FileActionProgressItemCreator](#) \*const creator)
  - before sending to db worker*
- void **schedulingForWrite** (const QString &action, [FileActionProgressItemCreator](#) \*const creator) const
- void **schedulingForWrite** (int numberOfInfos, const QString &action, [FileActionProgressItemCreator](#) \*const creator) const
  - db worker calls this before sending to file worker*
- void **written** (int numberOfInfos) const
- void **writtenToOne** () const
  - file worker calls this when finished*

## Protected Attributes

- int **m\_n** = 1

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::FileActionItemInfoList](#)

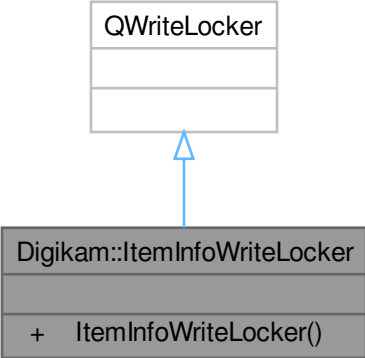
- static [FileActionItemInfoList](#) **continueTask** (const QList< [ItemInfo](#) > &list, [FileActionProgressItemContainer](#) \*const container)
- static [FileActionItemInfoList](#) **create** (const QList< [ItemInfo](#) > &list)

## Public Attributes inherited from [Digikam::FileActionItemInfoList](#)

- QExplicitlySharedDataPointer< [FileActionProgressItemContainer](#) > **container**

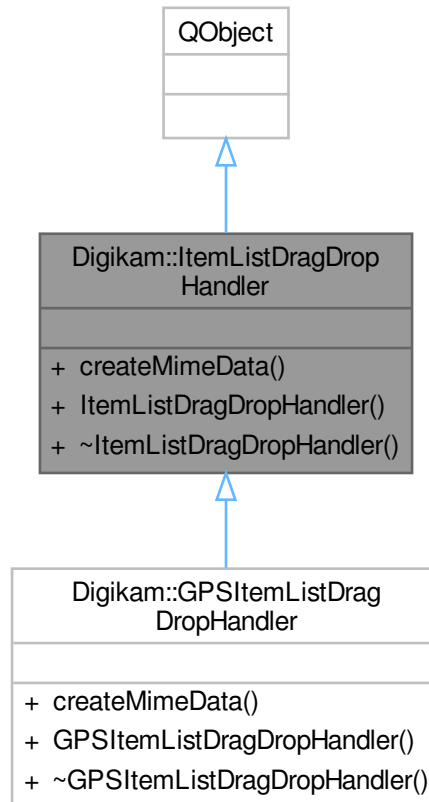
## 6.925 Digikam::ItemInfoWriteLocker Class Reference

Inheritance diagram for Digikam::ItemInfoWriteLocker:



## 6.926 Digikam::ItemListDragDropHandler Class Reference

Inheritance diagram for Digikam::ItemListDragDropHandler:



### Public Member Functions

- virtual `QMimeData *` **createMimeData** (`const QList< QPersistentModelIndex > &modelIndices`)=0
- **ItemListDragDropHandler** (`QObject *const parent=nullptr`)

## 6.927 Digikam::ItemLister Class Reference

### Classes

- class [Private](#)



## Public Member Functions

- void [list](#) ([ItemListerReceiver](#) \*const receiver, const [CoreDbUrl](#) &url)
- void [listDateRange](#) ([ItemListerReceiver](#) \*const receiver, const [QDate](#) &startDate, const [QDate](#) &endDate)
- void [setListOnlyAvailable](#) (bool listOnlyAvailable)
- void [setRecursive](#) (bool recursive)

## Operations with TAlbum

- void [listTag](#) ([ItemListerReceiver](#) \*const receiver, const [QList](#)< int > &tagIds)
- void [listImageTagPropertySearch](#) ([ItemListerReceiver](#) \*const receiver, const [QString](#) &xml)
- [QString](#) [tagSearchXml](#) (int tagId, const [QString](#) &type, bool includeChildTags) const

## Operations with SAlbum

- void [listSearch](#) ([ItemListerReceiver](#) \*const receiver, const [QString](#) &xml, int limit=0, qlonglong reference←  
ImageId=-1)
- void [listHaarSearch](#) ([ItemListerReceiver](#) \*const receiver, const [QString](#) &xml)
- void [listAreaRange](#) ([ItemListerReceiver](#) \*const receiver, double lat1, double lat2, double lon1, double lon2)

## Operations with PAlbum

- void [listPAlbum](#) ([ItemListerReceiver](#) \*const receiver, int albumRootId, const [QString](#) &album)

## 6.927.1 Member Function Documentation

### 6.927.1.1 list()

```
void Digikam::ItemLister::list (
    ItemListerReceiver *const receiver,
    const CoreDbUrl & url )
```

Convenience method for [Album](#), [Tag](#) and [Date](#) URLs, *not* for [Search](#) URLs.

### 6.927.1.2 listAreaRange()

```
void Digikam::ItemLister::listAreaRange (
    ItemListerReceiver *const receiver,
    double lat1,
    double lat2,
    double lon1,
    double lon2 )
```

List the images whose coordinates are between coordinates contained in [areaCoordinates](#)(lat1, lat2, lng1, lng2).

### 6.927.1.3 listDateRange()

```
void Digikam::ItemLister::listDateRange (
    ItemListerReceiver *const receiver,
    const QDate & startDate,
    const QDate & endDate )
```

List those images whose date lies in the range beginning with startDate (inclusive) and ending before endDate (exclusive).

### 6.927.1.4 listHaarSearch()

```
void Digikam::ItemLister::listHaarSearch (
    ItemListerReceiver *const receiver,
    const QString & xml )
```

Execute the search specified by search XML describing a Haar search

#### Parameters

<i>receiver</i>	the receiver for the searches
<i>xml</i>	SearchXml describing the query

### 6.927.1.5 listImageTagPropertySearch()

```
void Digikam::ItemLister::listImageTagPropertySearch (
    ItemListerReceiver *const receiver,
    const QString & xml )
```

Execute the search specified by search XML describing a Tag Properties search. Two special add-ons: Non-unique by image id; if enabled, uses the extended ImageRecord protocol to pass the property value in the record's extra↔ Value.

#### Parameters

<i>receiver</i>	the receiver for the searches
<i>xml</i>	SearchXml describing the query

### 6.927.1.6 listPAlbum()

```
void Digikam::ItemLister::listPAlbum (
    ItemListerReceiver *const receiver,
    int albumRootId,
    const QString & album )
```

List images in the [Album](#) (physical album) specified by albumRoot, album. The results will be fed to the specified receiver.

### 6.927.1.7 listSearch()

```
void Digikam::ItemLister::listSearch (
    ItemListerReceiver *const receiver,
    const QString & xml,
    int limit = 0,
    qlonglong referenceImageId = -1 )
```

Execute the search specified by search XML

#### Parameters

<i>receiver</i>	the receiver for the searches
<i>xml</i>	SearchXml describing the query
<i>limit</i>	the limit the count of the result set. If limit = 0, then no limit is set.
<i>referenceImageId</i>	the id of a reference image in the search query.

### 6.927.1.8 listTag()

```
void Digikam::ItemLister::listTag (
    ItemListerReceiver *const receiver,
    const QList< int > & tagIds )
```

List the images which have assigned the tags specified by tagIds Updated to support multiple tags

### 6.927.1.9 setListOnlyAvailable()

```
void Digikam::ItemLister::setListOnlyAvailable (
    bool listOnlyAvailable )
```

Adjust the setting if images from collections that are currently not in the state "available" will be included in the listing. Default: true.

### 6.927.1.10 setRecursive()

```
void Digikam::ItemLister::setRecursive (
    bool recursive )
```

Adjust the setting if album or tags will be listed recursively (i.e. including subalbums / subtags)

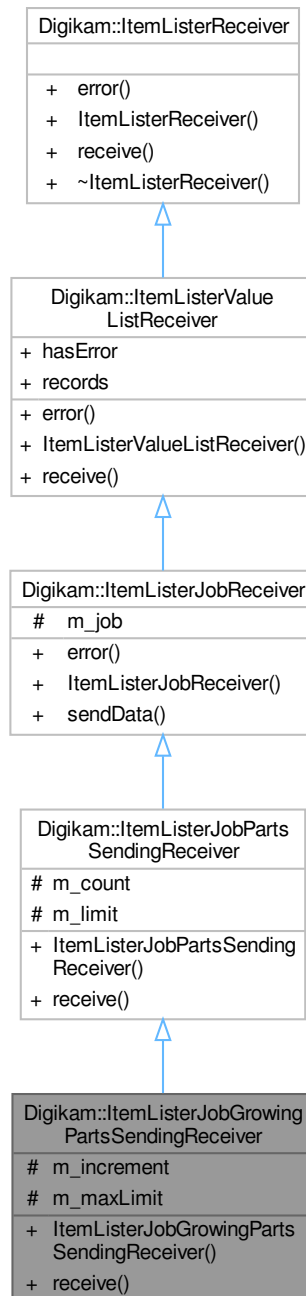
## 6.928 Digikam::ItemLister::Private Class Reference

### Public Attributes

- bool **listOnlyAvailableImages** = true
- bool **recursive** = true

## 6.929 Digikam::ItemListerJobGrowingPartsSendingReceiver Class Reference

Inheritance diagram for Digikam::ItemListerJobGrowingPartsSendingReceiver:



### Public Member Functions

- **ItemListerJobGrowingPartsSendingReceiver** (`DBJob *const job, int start, int end, int increment`)
- void `receive` (`const ItemListerRecord &record`) override

## Public Member Functions inherited from [Digikam::ItemListerJobPartsSendingReceiver](#)

- **ItemListerJobPartsSendingReceiver** ([DBJob](#) \*const job, int limit)
- void **receive** (const [ItemListerRecord](#) &record) override

## Public Member Functions inherited from [Digikam::ItemListerJobReceiver](#)

- void **error** (const QString &errMsg) override
- **ItemListerJobReceiver** ([DBJob](#) \*const job)
- void **sendData** ()

## Public Member Functions inherited from [Digikam::ItemListerValueListReceiver](#)

- void **error** (const QString &errMsg) override
- void **receive** (const [ItemListerRecord](#) &record) override

## Protected Attributes

- int **m\_increment** = 0
- int **m\_maxLimit** = 0

## Protected Attributes inherited from [Digikam::ItemListerJobPartsSendingReceiver](#)

- int **m\_count** = 0
- int **m\_limit** = 0

## Protected Attributes inherited from [Digikam::ItemListerJobReceiver](#)

- [DBJob](#) \*const **m\_job** = nullptr

## Additional Inherited Members

## Public Attributes inherited from [Digikam::ItemListerValueListReceiver](#)

- bool **hasError** = false
- QList< [ItemListerRecord](#) > **records**

## 6.929.1 Member Function Documentation

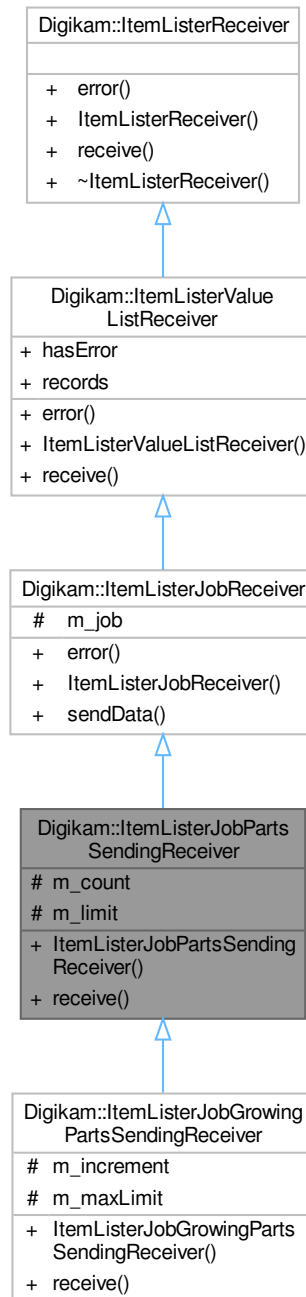
### 6.929.1.1 receive()

```
void Digikam::ItemListerJobGrowingPartsSendingReceiver::receive (
    const ItemListerRecord & record ) [override], [virtual]
```

Implements [Digikam::ItemListerReceiver](#).

## 6.930 Digikam::ItemListerJobPartsSendingReceiver Class Reference

Inheritance diagram for Digikam::ItemListerJobPartsSendingReceiver:



### Public Member Functions

- `ItemListerJobPartsSendingReceiver` (`DBJob *const job, int limit`)
- void `receive` (`const ItemListerRecord &record`) override

## Public Member Functions inherited from [Digikam::ItemListerJobReceiver](#)

- void [error](#) (const QString &errMsg) override
- [ItemListerJobReceiver](#) ([DBJob](#) \*const job)
- void [sendData](#) ()

## Public Member Functions inherited from [Digikam::ItemListerValueListReceiver](#)

- void [error](#) (const QString &errMsg) override
- void [receive](#) (const [ItemListerRecord](#) &record) override

## Protected Attributes

- int [m\\_count](#) = 0
- int [m\\_limit](#) = 0

## Protected Attributes inherited from [Digikam::ItemListerJobReceiver](#)

- [DBJob](#) \*const [m\\_job](#) = nullptr

## Additional Inherited Members

## Public Attributes inherited from [Digikam::ItemListerValueListReceiver](#)

- bool [hasError](#) = false
- QList< [ItemListerRecord](#) > [records](#)

## 6.930.1 Member Function Documentation

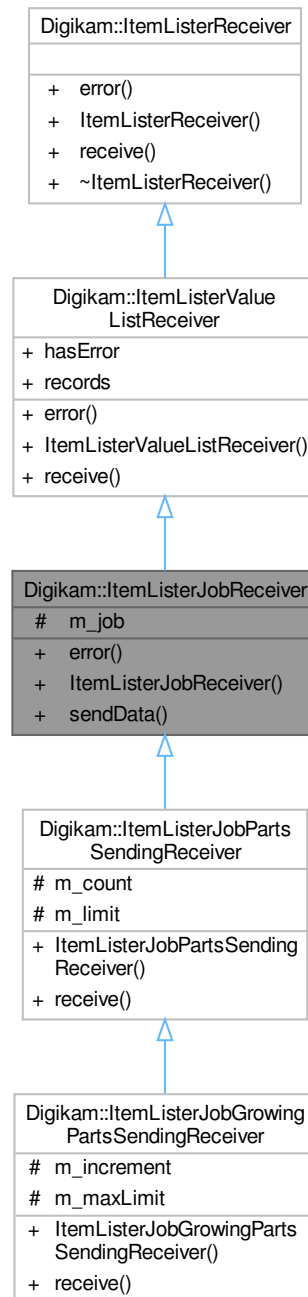
### 6.930.1.1 [receive\(\)](#)

```
void Digikam::ItemListerJobPartsSendingReceiver::receive (  
    const ItemListerRecord & record ) [override], [virtual]
```

Implements [Digikam::ItemListerReceiver](#).

## 6.931 Digikam::ItemListerJobReceiver Class Reference

Inheritance diagram for Digikam::ItemListerJobReceiver:



### Public Member Functions

- void `error` (const QString &errMsg) override
- `ItemListerJobReceiver` (DBJob \*const job)
- void `sendData` ()



## Public Member Functions inherited from [Digikam::ItemListerValueListReceiver](#)

- void [error](#) (const QString &errMsg) override
- void [receive](#) (const [ItemListerRecord](#) &record) override

## Protected Attributes

- [DBJob](#) \*const [m\\_job](#) = nullptr

## Additional Inherited Members

## Public Attributes inherited from [Digikam::ItemListerValueListReceiver](#)

- bool [hasError](#) = false
- QList< [ItemListerRecord](#) > [records](#)

## 6.931.1 Member Function Documentation

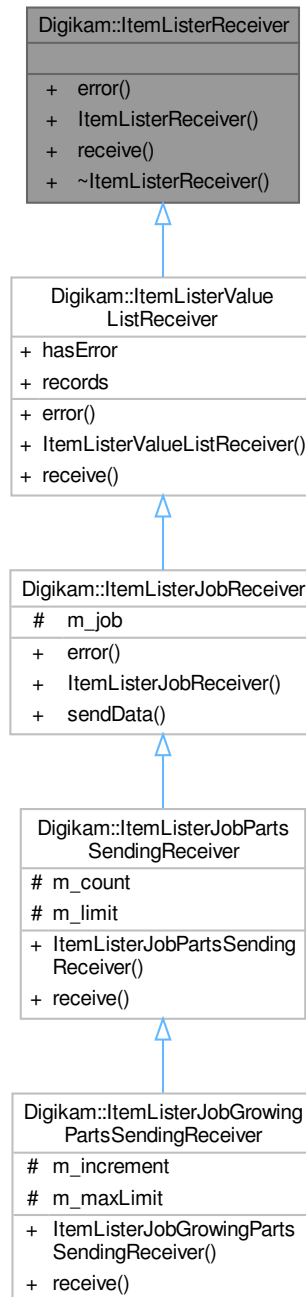
### 6.931.1.1 [error\(\)](#)

```
void Digikam::ItemListerJobReceiver::error (
    const QString & errMsg ) [override], [virtual]
```

Reimplemented from [Digikam::ItemListerReceiver](#).

## 6.932 Digikam::ItemListerReceiver Class Reference

Inheritance diagram for Digikam::ItemListerReceiver:



### Public Member Functions

- virtual void **error** (const QString &)
- virtual void **receive** (const [ItemListerRecord](#) &record)=0

## 6.933 Digikam::ItemLISTERRecord Class Reference

### Public Member Functions

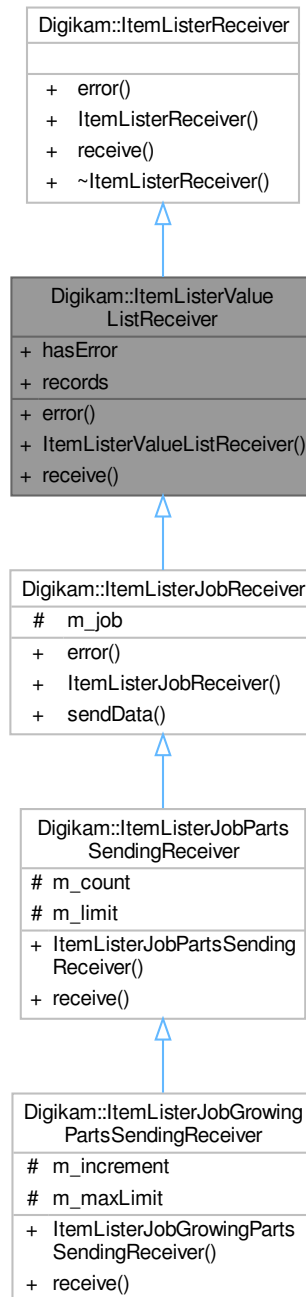
- bool **operator==** (const [ItemLISTERRecord](#) &record) const

### Public Attributes

- int **albumID** = -1
- int **albumRootID** = -1
- DatabaseItem::Category **category** = DatabaseItem::UndefinedCategory
- QDateTime **creationDate**
- qlonglong **currentReferenceImage** = -1
- double **currentSimilarity** = 0.0
- QList< QVariant > **extraValues**
- qlonglong **fileSize** = -1
- QString **format**
- qlonglong **imageID** = -1
- QSize **imageSize**
- QDateTime **modificationDate**
- QString **name**
- int **rating** = -1

## 6.934 Digikam::ItemListerValueListReceiver Class Reference

Inheritance diagram for Digikam::ItemListerValueListReceiver:



### Public Member Functions

- void `error` (const QString &errMsg) override
- void `receive` (const `ItemListerRecord` &record) override

## Public Attributes

- bool **hasError** = false
- QList< [ItemListerRecord](#) > **records**

## 6.934.1 Member Function Documentation

### 6.934.1.1 error()

```
void Digikam::ItemListerValueListReceiver::error (
    const QString & errMsg ) [override], [virtual]
```

Reimplemented from [Digikam::ItemListerReceiver](#).

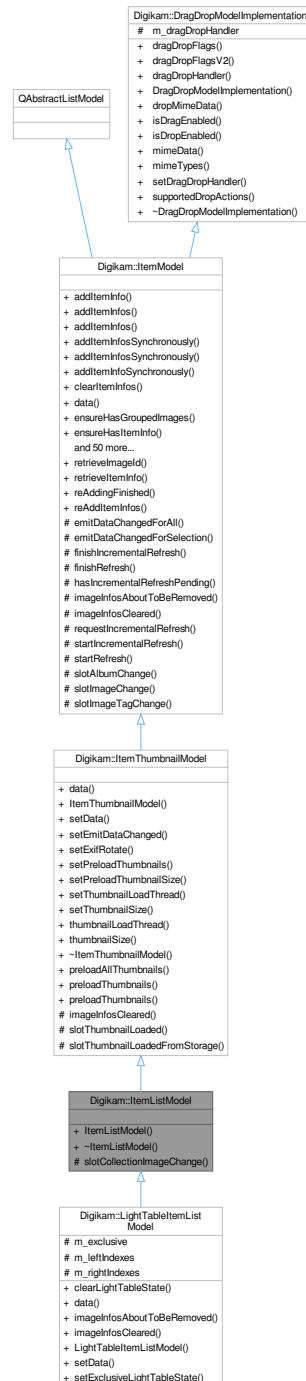
### 6.934.1.2 receive()

```
void Digikam::ItemListerValueListReceiver::receive (
    const ItemListerRecord & record ) [override], [virtual]
```

Implements [Digikam::ItemListerReceiver](#).

## 6.935 Digikam::ItemListModel Class Reference

Inheritance diagram for Digikam::ItemListModel:



### Signals

- void `imageInfosRemoved` (const QList< `ItemInfo` > &infos)

## Signals inherited from [Digikam::ItemThumbnailModel](#)

- void **thumbnailAvailable** (const QModelIndex &index, int requestedSize)
- void **thumbnailFailed** (const QModelIndex &index, int requestedSize)

## Signals inherited from [Digikam::ItemModel](#)

- void **allRefreshingFinished** ()
- void **imageChange** (const ImageChangeset &, const QItemSelection &)
- void **imageInfosAboutToBeAdded** (const QList< ItemInfo > &infos)
- void **imageInfosAboutToBeRemoved** (const QList< ItemInfo > &infos)
- void **imageInfosAdded** (const QList< ItemInfo > &infos)
- void **imageInfosRemoved** (const QList< ItemInfo > &infos)
- void **imageTagChange** (const ImageTagChangeset &, const QItemSelection &)
- void **preprocess** (const QList< ItemInfo > &infos, const QList< QVariant > &)
- void **processAdded** (const QList< ItemInfo > &infos, const QList< QVariant > &)
- void **readyForIncrementalRefresh** ()

## Public Member Functions

- **ItemListModel** (QWidget \*const parent)

## Public Member Functions inherited from [Digikam::ItemThumbnailModel](#)

- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- **ItemThumbnailModel** (QWidget \*const parent)
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::DisplayRole) override
- void **setEmitDataChanged** (bool emitSignal)
- void **setExifRotate** (bool rotate)
- void **setPreloadThumbnails** (bool preload)
- void **setPreloadThumbnailSize** (const ThumbnailSize &thumbSize)  
*If you want to fix a size for preloading, do it here.*
- void **setThumbnailLoadThread** (ThumbnailLoadThread \*const thread)
- void **setThumbnailSize** (const ThumbnailSize &thumbSize)  
*Set the thumbnail size to use.*
- **ThumbnailLoadThread** \* **thumbnailLoadThread** () const
- **ThumbnailSize** **thumbnailSize** () const

## Public Member Functions inherited from [Digikam::ItemModel](#)

- void **addItemInfo** (const ItemInfo &info)
- void **addItemInfos** (const QList< ItemInfo > &infos)
- void **addItemInfos** (const QList< ItemInfo > &infos, const QList< QVariant > &extraValues)
- void **addItemInfosSynchronously** (const QList< ItemInfo > &infos)
- void **addItemInfosSynchronously** (const QList< ItemInfo > &infos, const QList< QVariant > &extraValues)
- void **addItemInfoSynchronously** (const ItemInfo &info)
- void **clearItemInfos** ()
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- void **ensureHasGroupedImages** (const ItemInfo &groupLeader)
- void **ensureHasItemInfo** (const ItemInfo &info)
- void **ensureHasItemInfos** (const QList< ItemInfo > &infos)

- void **ensureHasItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **hasImage** (const [ItemInfo](#) &info) const
- bool **hasImage** (const [ItemInfo](#) &info, const QVariant &extraValue) const
- bool **hasImage** (qulonglong id) const
- bool **hasImage** (qulonglong id, const QVariant &extraValue) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- qulonglong **imageId** (const QModelIndex &index) const
- qulonglong **imageId** (int row) const
- QList< qulonglong > **imageIds** () const
- QList< qulonglong > **imageIds** (const QList< QModelIndex > &indexes) const
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfo](#) **imageInfo** (const QString &filePath) const
- [ItemInfo](#) **imageInfo** (int row) const
- [ItemInfo](#) & **imageInfoRef** (const QModelIndex &index) const
- [ItemInfo](#) & **imageInfoRef** (int row) const
- QList< [ItemInfo](#) > **imageInfos** () const
- QList< [ItemInfo](#) > **imageInfos** (const QList< QModelIndex > &indexes) const
- QList< [ItemInfo](#) > **imageInfos** (const QString &filePath) const
- QModelIndex **index** (int row, int column=0, const QModelIndex &parent=QModelIndex()) const override
- QList< QModelIndex > **indexesForImageId** (qulonglong id) const
- QList< QModelIndex > **indexesForItemInfo** (const [ItemInfo](#) &info) const
- QList< QModelIndex > **indexesForPath** (const QString &filePath) const
- QModelIndex **indexForImageId** (qulonglong id) const
- QModelIndex **indexForImageId** (qulonglong id, const QVariant &extraValue) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info, const QVariant &extraValue) const
- QModelIndex **indexForPath** (const QString &filePath) const
- bool **isEmpty** () const
- bool **isRefreshing** () const
- int **itemCount** () const
- **ItemModel** (QObject \*const parent=nullptr)
- bool **keepsFilePathCache** () const
- int **numberOfIndexesForImageId** (qulonglong id) const
- int **numberOfIndexesForItemInfo** (const [ItemInfo](#) &info) const
- void **removeIndex** (const QModelIndex &indexes)
- void **removeIndexes** (const QList< QModelIndex > &indexes)
- void **removeItemInfo** (const [ItemInfo](#) &info)
- void **removeItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **removeItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **setKeepsFilePathCache** (bool keepCache)
- DECLARE\_MODEL\_DRAG\_DROP\_METHODS void **setPreprocessor** (QObject \*const processor)
- void **setSendRemovalSignals** (bool send)
- void **setWatchFlags** (const [DatabaseFields::Set](#) &set)
- QList< [ItemInfo](#) > **uniqueItemInfos** () const
- void **unsetPreprocessor** (QObject \*const processor)



## Public Member Functions inherited from Digikam::DragDropModelImplementation

- virtual Qt::ItemFlags [dragDropFlags](#) (const QModelIndex &index) const
- Qt::ItemFlags [dragDropFlagsV2](#) (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* [dragDropHandler](#) () const
- [DragDropModelImplementation](#) ()=default
- bool [dropMimeData](#) (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool [isDragEnabled](#) (const QModelIndex &index) const
- virtual bool [isDropEnabled](#) (const QModelIndex &index) const
- QMimeData \* [mimeData](#) (const QModelIndexList &indexes) const
- QStringList [mimeTypes](#) () const
- void [setDragDropHandler](#) ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions [supportedDropActions](#) () const

## Protected Slots

- void [slotCollectionImageChange](#) (const [CollectionImageChangeset](#) &changeset)

## Protected Slots inherited from Digikam::ItemThumbnailModel

- void [slotThumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const QPixmap &thumb)
- void [slotThumbnailLoadedFromStorage](#) (const [LoadingDescription](#) &loadingDescription, const QPixmap &thumb)

## Protected Slots inherited from Digikam::ItemModel

- virtual void [slotAlbumChange](#) (const [AlbumChangeset](#) &changeset)
- virtual void [slotImageChange](#) (const [ImageChangeset](#) &changeset)
- virtual void [slotImageTagChange](#) (const [ImageTagChangeset](#) &changeset)

## Additional Inherited Members

## Public Types inherited from Digikam::ItemModel

- enum [ItemModelRoles](#) {  
[ItemModelPointerRole](#) = Qt::UserRole , [ItemModelInternalId](#) = Qt::UserRole + 1 , [ThumbnailRole](#) = Qt::UserRole + 2 , [CreationDateRole](#) = Qt::UserRole + 3 ,  
[ExtraDataRole](#) = Qt::UserRole + 5 , [ExtraDataDuplicateCount](#) = Qt::UserRole + 6 , [LTRightPanelRole](#) = Qt::UserRole + 50 , [LTRightPanelRole](#) = Qt::UserRole + 51 ,  
[SubclassRoles](#) = Qt::UserRole + 100 , [FilterModelRoles](#) = Qt::UserRole + 500 }

## Public Slots inherited from Digikam::ItemThumbnailModel

- void [preloadAllThumbnails](#) ()
- void [preloadThumbnails](#) (const QList< [ItemInfo](#) > &)
- void [preloadThumbnails](#) (const QList< QModelIndex > &)

## Public Slots inherited from [Digikam::ItemModel](#)

- void **reAddingFinished** ()
- void **reAddItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)

## Static Public Member Functions inherited from [Digikam::ItemModel](#)

- static qlonglong **retrievelmageld** (const QModelIndex &index)
- static [ItemInfo](#) **retrievelItemInfo** (const QModelIndex &index)

## Protected Member Functions inherited from [Digikam::ItemThumbnailModel](#)

- void [imageInfosCleared](#) () override

## Protected Member Functions inherited from [Digikam::ItemModel](#)

- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelection** (const QItemSelection &selection)
- void **finishIncrementalRefresh** ()
- void **finishRefresh** ()
- bool **hasIncrementalRefreshPending** () const
- virtual void [imageInfosAboutToBeRemoved](#) (int, int)
- void [requestIncrementalRefresh](#) ()
- void [startIncrementalRefresh](#) ()
- void [startRefresh](#) ()

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr

## 6.935.1 Member Function Documentation

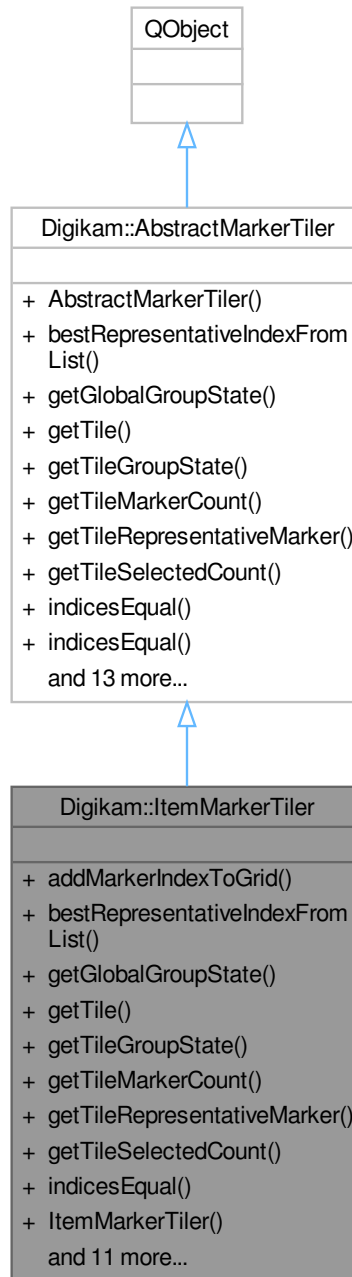
### 6.935.1.1 [imageInfosRemoved](#)

```
void Digikam::ItemListModel::imageInfosRemoved (  
    const QList< ItemInfo > & infos ) [signal]
```

Emitted when images are removed from the model because they are removed in the database

## 6.936 Digikam::ItemMarkerTiler Class Reference

Inheritance diagram for Digikam::ItemMarkerTiler:



### Public Member Functions

- void **addMarkerIndexToGrid** (const QPersistentModelIndex &markerIndex)
- QVariant **bestRepresentativeIndexFromList** (const QList< QVariant > &indices, const int sortKey) override

- GeoGroupState [getGlobalGroupState](#) () override
- [Tile](#) \* [getTile](#) (const [TileIndex](#) &tileIndex, const bool stopIfEmpty) override
- GeoGroupState [getTileGroupState](#) (const [TileIndex](#) &tileIndex) override
- int [getTileMarkerCount](#) (const [TileIndex](#) &tileIndex) override
- QVariant [getTileRepresentativeMarker](#) (const [TileIndex](#) &tileIndex, const int sortKey) override
- *These should be implemented for thumbnail handling.*
- int [getTileSelectedCount](#) (const [TileIndex](#) &tileIndex) override
- bool [indicesEqual](#) (const QVariant &a, const QVariant &b) const override
- **ItemMarkerTiler** ([GeoModelHelper](#) \*const modelHelper, QObject \*const parent=nullptr)
- void [onIndicesClicked](#) (const [ClickInfo](#) &clickInfo) override
- *These can be implemented if you want to react to actions in geolocation interface.*
- void [onIndicesMoved](#) (const [TileIndex::List](#) &tileIndicesList, const [GeoCoordinates](#) &targetCoordinates, const [QPersistentModelIndex](#) &targetSnapIndex) override
- QPixmap [pixmapFromRepresentativeIndex](#) (const QVariant &index, const QSize &size) override
- void [prepareTiles](#) (const [GeoCoordinates](#) &upperLeft, const [GeoCoordinates](#) &lowerRight, int level) override
- void [regenerateTiles](#) () override
- void [removeMarkerIndexFromGrid](#) (const [QModelIndex](#) &markerIndex, const bool ignoreSelection=false)
- *Remove a marker from the grid.*
- void [setActive](#) (const bool state) override
- void [setMarkerGeoModelHelper](#) ([GeoModelHelper](#) \*const modelHelper)
- [Tile](#) \* [tileNew](#) () override
- TilerFlags [tilerFlags](#) () const override
- *These have to be implemented.*

## Public Member Functions inherited from [Digikam::AbstractMarkerTiler](#)

- **AbstractMarkerTiler** (QObject \*const parent=nullptr)
- bool [indicesEqual](#) (const QList &a, const QList &b, const int upToLevel) const
- bool [isDirty](#) () const
- void [resetRootTile](#) ()
- [Tile](#) \* [rootTile](#) ()
- void [setDirty](#) (const bool state=true)

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractMarkerTiler](#)

- enum [TilerFlag](#) { [FlagNull](#) = 0 , [FlagMovable](#) = 1 }

## Signals inherited from [Digikam::AbstractMarkerTiler](#)

- void [signalThumbnailAvailableForIndex](#) (const QVariant &index, const QPixmap &pixmap)
- void [signalTilesOrSelectionChanged](#) ()

## 6.936.1 Member Function Documentation

### 6.936.1.1 [bestRepresentativeIndexFromList\(\)](#)

```
QVariant Digikam::ItemMarkerTiler::bestRepresentativeIndexFromList (
    const QList< QVariant > & indices,
    const int sortKey ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.936.1.2 getGlobalGroupState()

```
GeoGroupState Digikam::ItemMarkerTiler::getGlobalGroupState ( ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.936.1.3 getTile()

```
AbstractMarkerTiler::Tile * Digikam::ItemMarkerTiler::getTile (
    const TileIndex & tileIndex,
    const bool stopIfEmpty ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.936.1.4 getTileGroupState()

```
GeoGroupState Digikam::ItemMarkerTiler::getTileGroupState (
    const TileIndex & tileIndex ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.936.1.5 getTileMarkerCount()

```
int Digikam::ItemMarkerTiler::getTileMarkerCount (
    const TileIndex & tileIndex ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.936.1.6 getTileRepresentativeMarker()

```
QVariant Digikam::ItemMarkerTiler::getTileRepresentativeMarker (
    const TileIndex & tileIndex,
    const int sortKey ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.936.1.7 getTileSelectedCount()

```
int Digikam::ItemMarkerTiler::getTileSelectedCount (
    const TileIndex & tileIndex ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

### 6.936.1.8 indicesEqual()

```
bool Digikam::ItemMarkerTiler::indicesEqual (
    const QVariant & a,
    const QVariant & b ) const [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

**6.936.1.9 onIndicesClicked()**

```
void Digikam::ItemMarkerTiler::onIndicesClicked (
    const ClickInfo & clickInfo ) [override], [virtual]
```

Reimplemented from [Digikam::AbstractMarkerTiler](#).

**6.936.1.10 onIndicesMoved()**

```
void Digikam::ItemMarkerTiler::onIndicesMoved (
    const TileIndex::List & tileIndicesList,
    const GeoCoordinates & targetCoordinates,
    const QPersistentModelIndex & targetSnapIndex ) [override], [virtual]
```

Reimplemented from [Digikam::AbstractMarkerTiler](#).

**6.936.1.11 pixmapFromRepresentativeIndex()**

```
QPixmap Digikam::ItemMarkerTiler::pixmapFromRepresentativeIndex (
    const QVariant & index,
    const QSize & size ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

**6.936.1.12 prepareTiles()**

```
void Digikam::ItemMarkerTiler::prepareTiles (
    const GeoCoordinates & upperLeft,
    const GeoCoordinates & lowerRight,
    int level ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

**6.936.1.13 regenerateTiles()**

```
void Digikam::ItemMarkerTiler::regenerateTiles ( ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

**6.936.1.14 removeMarkerIndexFromGrid()**

```
void Digikam::ItemMarkerTiler::removeMarkerIndexFromGrid (
    const QModelIndex & markerIndex,
    const bool ignoreSelection = false )
```

**Parameters**

<i>markerIndex</i>	The marker index to remove
<i>ignoreSelection</i>	Do not remove the marker from the count of selected items. This is only used by slotSourceModelRowsAboutToBeRemoved internally, because the selection model sends us an extra signal about the deselection.

**6.936.1.15 setActive()**

```
void Digikam::ItemMarkerTiler::setActive (
    const bool state ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

**6.936.1.16 tileNew()**

```
AbstractMarkerTiler::Tile * Digikam::ItemMarkerTiler::tileNew ( ) [override], [virtual]
```

Implements [Digikam::AbstractMarkerTiler](#).

**6.936.1.17 tilerFlags()**

```
AbstractMarkerTiler::TilerFlags Digikam::ItemMarkerTiler::tilerFlags ( ) const [override],
[virtual]
```

Reimplemented from [Digikam::AbstractMarkerTiler](#).

**6.937 Digikam::ItemMetadataAdjustmentHint Class Reference****Public Types**

- enum [AdjustmentStatus](#) { [AboutToEditMetadata](#) , [MetadataEditingFinished](#) , [MetadataEditingAborted](#) }

**Public Member Functions**

- [AdjustmentStatus](#) **adjustmentStatus** () const
- qlonglong **fileSize** () const
- qlonglong **id** () const
- bool **isAboutToEdit** () const
- bool **isEditingFinished** () const
- bool **isEditingFinishedAborted** () const
- [ItemMetadataAdjustmentHint](#) (qlonglong id, [AdjustmentStatus](#) status, const QDateTime &modification↔  
DateOnDisk, qlonglong fileSize)
- QDateTime **modificationDate** () const

**Protected Attributes**

- qlonglong **m\_fileSize** = 0
- qlonglong **m\_id** = 0
- QDateTime **m\_modificationDate**
- [AdjustmentStatus](#) **m\_status** = [AboutToEditMetadata](#)

**6.937.1 Member Enumeration Documentation****6.937.1.1 AdjustmentStatus**

```
enum Digikam::ItemMetadataAdjustmentHint::AdjustmentStatus
```

The file's has been edited writing out information from the database, i.e., the db is already guaranteed to contain all changed information in the file's metadata. There is no need for a full rescan, optimizations are possible.

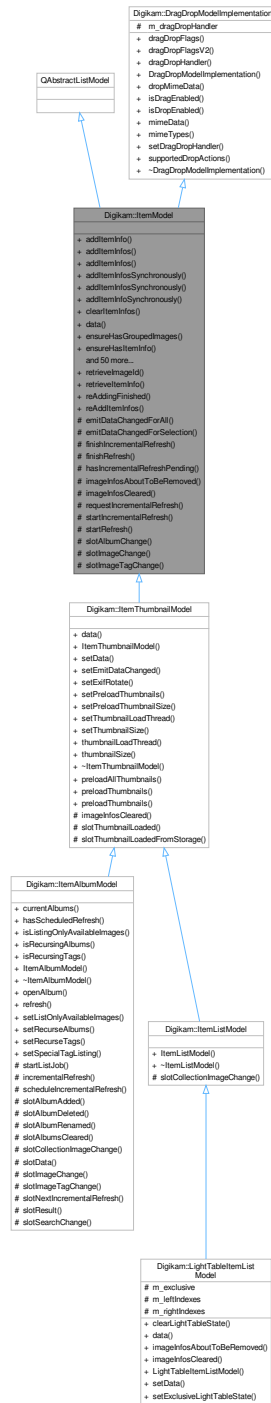
## Enumerator

AboutToEditMetadata	The file is about to be edited. Suspends scanning. The Finished hint must follow.
MetadataEditingFinished	The file's metadata has been edited as described above.
MetadataEditingAborted	The file's metadata has not been edited, despite sending AboutToEditMedata.



## 6.938 Digikam::ItemModel Class Reference

Inheritance diagram for Digikam::ItemModel:



### Public Types

- enum `ItemModelRoles` {
  - `ItemModelPointerRole` = Qt::UserRole , `ItemModelInternalId` = Qt::UserRole + 1 , `ThumbnailRole` = Qt::UserRole + 2 , `CreationDateRole` = Qt::UserRole + 3 ,

```

ExtraDataRole = Qt::UserRole + 5 , ExtraDataDuplicateCount = Qt::UserRole + 6 , LTLeftPanelRole = Qt::UserRole + 50 , LTRightPanelRole = Qt::UserRole + 51 ,
SubclassRoles = Qt::UserRole + 100 , FilterModelRoles = Qt::UserRole + 500 }

```

## Public Slots

- void **reAddingFinished** ()
- void **reAddItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)

## Signals

- void [allRefreshingFinished](#) ()
- void [imageChange](#) (const [ImageChangeset](#) &, const QItemSelection &)
- void [imageInfosAboutToBeAdded](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosAboutToBeRemoved](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosAdded](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosRemoved](#) (const QList< [ItemInfo](#) > &infos)
- void [imageTagChange](#) (const [ImageTagChangeset](#) &, const QItemSelection &)
- void [preprocess](#) (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &)
- void **processAdded** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &)
- void [readyForIncrementalRefresh](#) ()

## Public Member Functions

- void [addItemInfo](#) (const [ItemInfo](#) &info)
- void **addItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **addItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void **addItemInfosSynchronously** (const QList< [ItemInfo](#) > &infos)
- void **addItemInfosSynchronously** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void [addItemInfoSynchronously](#) (const [ItemInfo](#) &info)
- void [clearItemInfos](#) ()
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- void [ensureHasGroupedImages](#) (const [ItemInfo](#) &groupLeader)
- void [ensureHasItemInfo](#) (const [ItemInfo](#) &info)
- void **ensureHasItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **ensureHasItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **hasImage** (const [ItemInfo](#) &info) const
- bool **hasImage** (const [ItemInfo](#) &info, const QVariant &extraValue) const
- bool **hasImage** (qulonglong id) const
- bool **hasImage** (qulonglong id, const QVariant &extraValue) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- qulonglong **imageId** (const QModelIndex &index) const
- qulonglong **imageId** (int row) const
- QList< qulonglong > **imageIds** () const
- QList< qulonglong > **imageIds** (const QList< QModelIndex > &indexes) const
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfo](#) **imageInfo** (const QString &filePath) const
- [ItemInfo](#) **imageInfo** (int row) const
- [ItemInfo](#) & **imageInfoRef** (const QModelIndex &index) const
- [ItemInfo](#) & **imageInfoRef** (int row) const
- QList< [ItemInfo](#) > **imageInfos** () const

- `QList< ItemInfo > imageInfos` (const `QList< QModelIndex > &indexes`) const
- `QList< ItemInfo > imageInfos` (const `QString &filePath`) const
- `QModelIndex index` (int row, int column=0, const `QModelIndex &parent=QModelIndex()`) const override
- `QList< QModelIndex > indexesForImageId` (qulonglong id) const
- `QList< QModelIndex > indexesForItemInfo` (const `ItemInfo &info`) const
- `QList< QModelIndex > indexesForPath` (const `QString &filePath`) const
- `QModelIndex indexForImageId` (qulonglong id) const
- `QModelIndex indexForImageId` (qulonglong id, const `QVariant &extraValue`) const
- `QModelIndex indexForItemInfo` (const `ItemInfo &info`) const
- `QModelIndex indexForItemInfo` (const `ItemInfo &info`, const `QVariant &extraValue`) const
- `QModelIndex indexForPath` (const `QString &filePath`) const
- bool `isEmpty` () const
- bool `isRefreshing` () const
- int `itemCount` () const
- `ItemModel` (`QObject *const parent=nullptr`)
- bool `keepsFilePathCache` () const
- int `numberOfIndexesForImageId` (qulonglong id) const
- int `numberOfIndexesForItemInfo` (const `ItemInfo &info`) const
- void `removeIndex` (const `QModelIndex &indexes`)
- void `removeIndexes` (const `QList< QModelIndex > &indexes`)
- void `removeItemInfo` (const `ItemInfo &info`)
- void `removeItemInfos` (const `QList< ItemInfo > &infos`)
- void `removeItemInfos` (const `QList< ItemInfo > &infos`, const `QList< QVariant > &extraValues`)
- int `rowCount` (const `QModelIndex &parent=QModelIndex()`) const override
- void `setItemInfos` (const `QList< ItemInfo > &infos`)
- void `setKeepsFilePathCache` (bool keepCache)
- `DECLARE_MODEL_DRAG_DROP_METHODS` void `setPreprocessor` (`QObject *const processor`)
- void `setSendRemovalSignals` (bool send)
- void `setWatchFlags` (const `DatabaseFields::Set &set`)
- `QList< ItemInfo > uniqueItemInfos` () const
- void `unsetPreprocessor` (`QObject *const processor`)

## Public Member Functions inherited from `Digikam::DragDropModelImplementation`

- virtual `Qt::ItemFlags dragDropFlags` (const `QModelIndex &index`) const
- `Qt::ItemFlags dragDropFlagsV2` (const `QModelIndex &index`) const
- `AbstractItemDragDropHandler * dragDropHandler` () const
- `DragDropModelImplementation` ()=default
- bool `dropMimeData` (const `QMimeData *`, `Qt::DropAction`, int, int, const `QModelIndex &`)
- virtual bool `isDragEnabled` (const `QModelIndex &index`) const
- virtual bool `isDropEnabled` (const `QModelIndex &index`) const
- `QMimeData * mimeData` (const `QModelIndexList &indexes`) const
- `QStringList mimeTypes` () const
- void `setDragDropHandler` (`AbstractItemDragDropHandler *handler`)
- `Qt::DropActions supportedDropActions` () const

## Static Public Member Functions

- static qulonglong `retrievalImageId` (const `QModelIndex &index`)
- static `ItemInfo retrievalItemInfo` (const `QModelIndex &index`)

## Protected Slots

- virtual void **slotAlbumChange** (const [AlbumChangeset](#) &changeset)
- virtual void **slotImageChange** (const [ImageChangeset](#) &changeset)
- virtual void **slotImageTagChange** (const [ImageTagChangeset](#) &changeset)

## Protected Member Functions

- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelection** (const [QItemSelection](#) &selection)
- void **finishIncrementalRefresh** ()
- void **finishRefresh** ()
- bool **hasIncrementalRefreshPending** () const
- virtual void **imageInfosAboutToBeRemoved** (int, int)
- virtual void **imageInfosCleared** ()
- void **requestIncrementalRefresh** ()
- void **startIncrementalRefresh** ()
- void **startRefresh** ()

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr

## 6.938.1 Member Enumeration Documentation

### 6.938.1.1 ItemModelRoles

```
enum Digikam::ItemModel::ItemModelRoles
```

#### Enumerator

<a href="#">ItemModelPointerRole</a>	An <a href="#">ItemModel*</a> pointer to this model
<a href="#">ThumbnailRole</a>	Returns a thumbnail pixmap. May be implemented by subclasses. Returns either a valid pixmap or a null <a href="#">QVariant</a> .
<a href="#">CreationDateRole</a>	Returns a <a href="#">QDateTime</a> with the creation date
<a href="#">ExtraDataRole</a>	Return (optional) <a href="#">extraData</a> field
<a href="#">ExtraDataDuplicateCount</a>	Returns the number of duplicate indexes for the same image id
<a href="#">LLeftPanelRole</a>	Roles which are defined here but not implemented by <a href="#">ItemModel</a> Returns position of item in Left Light Table preview.
<a href="#">LRightPanelRole</a>	Returns position of item in Right Light Table preview.
<a href="#">SubclassRoles</a>	For use by subclasses
<a href="#">FilterModelRoles</a>	For use by filter models

## 6.938.2 Member Function Documentation

### 6.938.2.1 addItemInfo()

```
void Digikam::ItemModel::addItemInfo (
    const ItemInfo & info )
```

Main entry point for subclasses adding image infos to the model. If you list entries not unique per image id, you must add an `extraValue` so that every entry is unique by `imageId` and `extraValues`. Please note that these methods do not prevent addition of duplicate entries.

### 6.938.2.2 addItemInfoSynchronously()

```
void Digikam::ItemModel::addItemInfoSynchronously (
    const ItemInfo & info )
```

[addItemInfo\(\)](#) is asynchronous if a preprocessor is set. This method first adds the info, synchronously. Only afterwards, the preprocessor will have the opportunity to process it. This method also bypasses any incremental updates. Please note that these methods do not prevent addition of duplicate entries.

### 6.938.2.3 allRefreshingFinished

```
void Digikam::ItemModel::allRefreshingFinished ( ) [signal]
```

Signals that the model has finished currently with all scheduled refreshing, full or incremental, and all preprocessing. The model is in polished, clean situation right now.

### 6.938.2.4 clearItemInfos()

```
void Digikam::ItemModel::clearItemInfos ( )
```

Clears image infos and resets model.

### 6.938.2.5 ensureHasGroupedImages()

```
void Digikam::ItemModel::ensureHasGroupedImages (
    const ItemInfo & groupLeader )
```

Ensure that all images grouped on the given leader are contained in the model.

### 6.938.2.6 ensureHasItemInfo()

```
void Digikam::ItemModel::ensureHasItemInfo (
    const ItemInfo & info )
```

Add the given entries. Method returns immediately, the addition may happen later asynchronously. These methods prevent the addition of duplicate entries.

**6.938.2.7 imageChange**

```
void Digikam::ItemModel::imageChange (
    const ImageChangeset & ,
    const QItemSelection & ) [signal]
```

If an [ImageChangeset](#) affected indexes of this model with changes as set in `watchFlags()`, this signal contains the `changeset` and the affected indexes.

**6.938.2.8 imageInfo() [1/2]**

```
ItemInfo Digikam::ItemModel::imageInfo (
    const QModelIndex & index ) const
```

Returns the [ItemInfo](#) object, reference or image id from the underlying data pointed to by the index. If the index is not valid, `imageInfo` will return a null [ItemInfo](#), `imageId` will return 0, `imageInfoRef` must not be called with an invalid index.

**6.938.2.9 imageInfo() [2/2]**

```
ItemInfo Digikam::ItemModel::imageInfo (
    int row ) const
```

Returns the [ItemInfo](#) object, reference or image id from the underlying data of the given row (parent is the invalid `QModelIndex`, column is 0). Note that `imageInfoRef` will crash if index is invalid.

**6.938.2.10 imageInfosAboutToBeAdded**

```
void Digikam::ItemModel::imageInfosAboutToBeAdded (
    const QList< ItemInfo > & infos ) [signal]
```

Informs that `ItemInfos` will be added to the model. This signal is sent before the model data is changed and views are informed.

**6.938.2.11 imageInfosAboutToBeRemoved [1/2]**

```
void Digikam::ItemModel::imageInfosAboutToBeRemoved (
    const QList< ItemInfo > & infos ) [signal]
```

Informs that `ItemInfos` will be removed from the model. This signal is sent before the model data is changed and views are informed. Note: You need to explicitly enable sending of this signal. It is not sent in [clearItemInfos\(\)](#).

**6.938.2.12 imageInfosAboutToBeRemoved() [2/2]**

```
virtual void Digikam::ItemModel::imageInfosAboutToBeRemoved (
    int ,
    int ) [inline], [protected], [virtual]
```

Called before `rowsAboutToBeRemoved`

### 6.938.2.13 imageInfosAdded

```
void Digikam::ItemModel::imageInfosAdded (
    const QList< ItemInfo > & infos ) [signal]
```

Informs that ItemInfos have been added to the model. This signal is sent after the model data is changed and views are informed.

### 6.938.2.14 imageInfosCleared()

```
virtual void Digikam::ItemModel::imageInfosCleared ( ) [inline], [protected], [virtual]
```

Called when the internal storage is cleared

Reimplemented in [Digikam::ItemThumbnailModel](#).

### 6.938.2.15 imageInfosRemoved

```
void Digikam::ItemModel::imageInfosRemoved (
    const QList< ItemInfo > & infos ) [signal]
```

Informs that ItemInfos have been removed from the model. This signal is sent after the model data is changed and views are informed. \* Note: You need to explicitly enable sending of this signal. It is not sent in [clearItemInfos\(\)](#).

### 6.938.2.16 imageTagChange

```
void Digikam::ItemModel::imageTagChange (
    const ImageTagChangeset & ,
    const QItemSelection & ) [signal]
```

If an [ImageTagChangeset](#) affected indexes of this model, this signal contains the changeset and the affected indexes.

### 6.938.2.17 indexForItemInfo()

```
QModelIndex Digikam::ItemModel::indexForItemInfo (
    const ItemInfo & info ) const
```

Return the index for the given [ItemInfo](#) or id, if contained in this model.

### 6.938.2.18 indexForPath()

```
QModelIndex Digikam::ItemModel::indexForPath (
    const QString & filePath ) const
```

Returns the index or [ItemInfo](#) object from the underlying data for the given file path. This is fast if `keepsFilePathCache` is enabled. The file path is as returned by [ItemInfo.filePath\(\)](#). In case of multiple occurrences of the same file, the simpler variants return any one found first, use the QList methods to retrieve all occurrences.

### 6.938.2.19 isRefreshing()

```
bool Digikam::ItemModel::isRefreshing ( ) const
```

Returns true if this model is currently refreshing. For a preprocessor this means that, although the preprocessor may currently have processed all it got, more batches are to be expected.

### 6.938.2.20 preprocess

```
void Digikam::ItemModel::preprocess (
    const QList< ItemInfo > & infos,
    const QList< QVariant > & ) [signal]
```

Connect to this signal only if you are the current preprocessor.

### 6.938.2.21 readyForIncrementalRefresh

```
void Digikam::ItemModel::readyForIncrementalRefresh ( ) [signal]
```

Signals that the model is right now ready to start an incremental refresh. This is guaranteed only for the scope of emitting this signal.

### 6.938.2.22 removeIndex()

```
void Digikam::ItemModel::removeIndex (
    const QModelIndex & indexes )
```

Directly remove the given indexes or infos from the model.

### 6.938.2.23 requestIncrementalRefresh()

```
void Digikam::ItemModel::requestIncrementalRefresh ( ) [protected]
```

As soon as the model is ready to start an incremental refresh, the signal [readyForIncrementalRefresh\(\)](#) will be emitted. The signal will be emitted inline if the model is ready right now.

### 6.938.2.24 retrieveItemInfo()

```
ItemInfo Digikam::ItemModel::retrieveItemInfo (
    const QModelIndex & index ) [static]
```

Retrieves the `imageInfo` object from the `data()` method of the given index. The index may be from a `QSortFilterProxyModel` as long as an `ItemModel` is at the end.

### 6.938.2.25 setItemInfos()

```
void Digikam::ItemModel::setItemInfos (
    const QList< ItemInfo > & infos )
```

Clears and adds the infos.



### 6.938.2.26 setKeepsFilePathCache()

```
void Digikam::ItemModel::setKeepsFilePathCache (
    bool keepCache )
```

If a cache is kept, lookup by file path is fast, without a cache it is O(n). Default is false.

### 6.938.2.27 setPreprocessor()

```
void Digikam::ItemModel::setPreprocessor (
    QObject *const processor )
```

Install an object as a preprocessor for ItemInfos added to this model. For every QList of ItemInfos added to addItemInfo, the signal [preprocess\(\)](#) will be emitted. The preprocessor may process the items and shall then readd them by calling [reAddItemInfos\(\)](#). It may take some time to process. It shall discard any held infos when the [modelReset\(\)](#) signal is sent. It shall call [readdFinished\(\)](#) when no reset occurred and all infos on the way have been readded. This means that only after calling this method, you shall make three connections (preprocess -> your slot, your signal -> [reAddItemInfos](#), your signal -> [reAddingFinished](#)) and make or already hold a connection [modelReset\(\)](#) -> your slot. There is only one preprocessor at a time, a previously set object will be disconnected.

### 6.938.2.28 setSendRemovalSignals()

```
void Digikam::ItemModel::setSendRemovalSignals (
    bool send )
```

Enable sending of [imageInfosAboutToBeRemoved](#) and [imageInfosRemoved](#) signals. Default: false

### 6.938.2.29 setWatchFlags()

```
void Digikam::ItemModel::setWatchFlags (
    const DatabaseFields::Set & set )
```

Set a set of database fields to watch. If either of these is changed, [dataChanged\(\)](#) will be emitted. Default is no flag (no signal will be emitted).

### 6.938.2.30 startIncrementalRefresh()

```
void Digikam::ItemModel::startIncrementalRefresh ( ) [protected]
```

Starts an incremental refresh operation. You shall only call this method from a slot connected to [readyForIncrementalRefresh\(\)](#). To initiate an incremental refresh, call [requestIncrementalRefresh\(\)](#).

### 6.938.2.31 startRefresh()

```
void Digikam::ItemModel::startRefresh ( ) [protected]
```

Subclasses that add ItemInfos in batches shall call [startRefresh\(\)](#) when they start sending batches and [finishRefresh\(\)](#) when they have finished. No incremental refreshes will be started while listing. A [clearItemInfos\(\)](#) always stops listing, calling [finishRefresh\(\)](#) is then not necessary.

## 6.939 Digikam::ItemPosition Class Reference

### Public Member Functions

- double **accuracy** () const
- double **altitude** () const
- QString **altitudeFormatted** () const
- void **apply** ()
- QString **description** () const
- bool **hasAccuracy** () const
- bool **hasAltitude** () const
- bool **hasCoordinates** () const
- bool **hasOrientation** () const
- bool **hasRoll** () const
- bool **hasTilt** () const
- bool **isEmpty** () const
- bool **isNull** () const
- [ItemPosition](#) ()
- **ItemPosition** (const [CoreDbAccess](#) &access, qlonglong imageld)
- **ItemPosition** (const [ItemPosition](#) &other)
- [ItemPosition](#) (qlonglong imageld)
- QString **latitude** () const
- QString **latitudeFormatted** () const
- double **latitudeNumber** () const
- bool **latitudeUserPresentableNumbers** (int \*degrees, int \*minutes, double \*seconds, char \*direction↔  
Reference)
- QString **longitude** () const
- QString **longitudeFormatted** () const
- double **longitudeNumber** () const
- bool **longitudeUserPresentableNumbers** (int \*degrees, int \*minutes, double \*seconds, char \*direction↔  
Reference)
- [ItemPosition](#) & **operator=** (const [ItemPosition](#) &other)
- double **orientation** () const
- void **remove** ()
- void **removeAltitude** ()
- double **roll** () const
- void **setAccuracy** (double accuracy)
- void **setAltitude** (double altitude)
- void **setDescription** (const QString &description)
- bool **setLatitude** (const QString &latitude)
- bool **setLatitude** (double latitudeNumber)
- bool **setLongitude** (const QString &longitude)
- bool **setLongitude** (double longitudeNumber)
- void **setOrientation** (double orientation)
- void **setRoll** (double roll)
- void **setTilt** (double tilt)
- double **tilt** () const

### 6.939.1 Constructor & Destructor Documentation

#### 6.939.1.1 ItemPosition() [1/2]

```
Digikam::ItemPosition::ItemPosition ( )
```

Creates a null [ItemPosition](#) object

### 6.939.1.2 ItemPosition() [2/2]

```
Digikam::ItemPosition::ItemPosition (
    qlonglong imageId ) [explicit]
```

Creates an [ItemPosition](#) object for the given image. The information is read from the database.

## 6.939.2 Member Function Documentation

### 6.939.2.1 altitude()

```
double Digikam::ItemPosition::altitude ( ) const
```

The altitude in meters

### 6.939.2.2 altitudeFormatted()

```
QString Digikam::ItemPosition::altitudeFormatted ( ) const
```

Returns the altitude formatted in a user-presentable way in the form "43.45m"

### 6.939.2.3 apply()

```
void Digikam::ItemPosition::apply ( )
```

Apply all changes made to this object. (Also called from destructor)

### 6.939.2.4 isEmpty()

```
bool Digikam::ItemPosition::isEmpty ( ) const
```

An object is empty if no entry exists in the [ItemPosition](#) table for the referenced image, or if the object is null. An empty object is empty even if values have been set; it becomes not empty after calling [apply\(\)](#).

### 6.939.2.5 latitude()

```
QString Digikam::ItemPosition::latitude ( ) const
```

Returns latitude/longitude in the format as described by the XMP specification as "GPSCoordinate": A Text value in the form ?DDD,MM,SSk? or ?DDD,MM.mmk?. This provides lossless storage.

### 6.939.2.6 latitudeFormatted()

```
QString Digikam::ItemPosition::latitudeFormatted ( ) const
```

Returns the latitude/longitude in a user-presentable version, in the form "30°45'55.123" East"

### 6.939.2.7 latitudeNumber()

```
double Digikam::ItemPosition::latitudeNumber ( ) const
```

Returns latitude/longitude as a double in degrees. North and East have a positive sign, South and West negative. This provides high precision, with the usual floating point concerns, and possible problems finding the exact text form when converting *back* to fractions.

### 6.939.2.8 latitudeUserPresentableNumbers()

```
bool Digikam::ItemPosition::latitudeUserPresentableNumbers (
    int * degrees,
    int * minutes,
    double * seconds,
    char * directionReference )
```

Returns latitude/longitude as user-presentable numbers. This means that degrees and minutes are integer, the seconds fractional. Direction reference is 'N'/'S', 'E'/'W' resp. This is for the purpose of presenting to the user, there are no guarantees on precision. Returns true if the values have been changed.

### 6.939.2.9 remove()

```
void Digikam::ItemPosition::remove ( )
```

Removes the whole data set for the referenced image from the database. This object and any [ItemPosition](#) object created later will be empty.

### 6.939.2.10 removeAltitude()

```
void Digikam::ItemPosition::removeAltitude ( )
```

Removes the altitude for the referenced image from the database.

### 6.939.2.11 setAltitude()

```
void Digikam::ItemPosition::setAltitude (
    double altitude )
```

Set the altitude in meters

### 6.939.2.12 setLatitude() [1/2]

```
bool Digikam::ItemPosition::setLatitude (
    const QString & latitude )
```

Sets the latitude/longitude from the GPSCoordinate string as described by XMP. Returns true if the format is accepted.

**6.939.2.13 setLatitude()** [2/2]

```
bool Digikam::ItemPosition::setLatitude (
    double latitudeNumber )
```

Sets the latitude/longitude from a double floating point number, as described for [latitudeNumber\(\)](#) above. Returns true if the value is valid and accepted.

## 6.940 Digikam::ItemPreviewCanvas Class Reference

Inheritance diagram for Digikam::ItemPreviewCanvas:



### Public Member Functions

- `ItemInfo` `imageInfo` () const
- void `setFaceGroup` (`FaceGroup` \*const group)
- void `setItemInfo` (const `ItemInfo` &info)

## Public Member Functions inherited from [Digikam::DImgPreviewItem](#)

- **DImgPreviewItem** (QGraphicsItem \*const parent=nullptr)
- **DImgPreviewItemPrivate** ()=default
- void **init** ([DImgPreviewItem](#) \*const q)
- bool **isLoading** () const
- QString **path** () const
- void **reload** ()
- void **setDisplayingWidget** (QWidget \*const widget)
- void **setPath** (const QString &path, bool rePreview=false)
- void **setPreloadPaths** (const QStringList &pathsToPreload)
- void **setPreviewSettings** (const [PreviewSettings](#) &settings)
- State **state** () const
- QString **userLoadingHint** () const override

## Public Member Functions inherited from [Digikam::GraphicsDImgItem](#)

- QRectF **boundingRect** () const override
- void **clearCache** ()
- **GraphicsDImgItem** (QGraphicsItem \*const parent=nullptr)
- **GraphicsDImgItemPrivate** ()=default
- [DImg](#) **image** () const
- void **init** ([GraphicsDImgItem](#) \*const q)
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget) override
- void **setDisplayWidget** (QWidget \*const widget)
- void **setImage** (const [DImg](#) &img)
- void **sizeHasChanged** ()
- [ImageZoomSettings](#) \* **zoomSettings** ()
- const [ImageZoomSettings](#) \* **zoomSettings** () const

## Protected Member Functions

- void **hoverEnterEvent** (QGraphicsSceneHoverEvent \*e) override
- void **hoverLeaveEvent** (QGraphicsSceneHoverEvent \*e) override
- void **hoverMoveEvent** (QGraphicsSceneHoverEvent \*e) override

## Protected Member Functions inherited from [Digikam::DImgPreviewItem](#)

- **DImgPreviewItem** (DImgPreviewItemPrivate &dd, QGraphicsItem \*const parent=nullptr)

## Protected Member Functions inherited from [Digikam::GraphicsDImgItem](#)

- void **contextMenuEvent** (QGraphicsSceneContextMenuEvent \*e) override
- **GraphicsDImgItem** (GraphicsDImgItemPrivate &dd, QGraphicsItem \*const parent)

## Additional Inherited Members

## Public Types inherited from [Digikam::DImgPreviewItem](#)

- enum **State** { **NoImage** , **Loading** , **ImageLoaded** , **ImageLoadingFailed** }

### Signals inherited from [Digikam::DImgPreviewItem](#)

- void **loaded** ()
- void **loadingFailed** ()
- void **stateChanged** (int state)

### Signals inherited from [Digikam::GraphicsDImgItem](#)

- void **imageChanged** ()
- void **imageSizeChanged** (const QSizeF &size)
- void **showContextMenu** (QGraphicsSceneContextMenuEvent \*e)

### Public Attributes inherited from [Digikam::DImgPreviewItem](#)

- bool **exifRotate** = false
- QString **path**
- QStringList **pathsToPreload**
- [PreviewLoadThread](#) \* **preloadThread** = nullptr
- [PreviewSettings](#) **previewSettings**
- int **previewSize** = 1024
- [PreviewLoadThread](#) \* **previewThread** = nullptr
- DImgPreviewItem::State **state** = DImgPreviewItem::NoImage

### Public Attributes inherited from [Digikam::GraphicsDImgItem](#)

- [CachedPixmap](#) **cachedPixmap**
- [DImg](#) **image**
- [ImageZoomSettings](#) **zoomSettings**

### Protected Attributes inherited from [Digikam::GraphicsDImgItem](#)

- [GraphicsDImgItemPrivate](#) \*const **d\_ptr**



## 6.941 Digikam::ItemPreviewView Class Reference

Inheritance diagram for Digikam::ItemPreviewView:



### Public Types

- enum **Mode** { **IconViewPreview** , **LightTablePreview** }

## Signals

- void **signalAddToExistingQueue** (int)
- void **signalDeleteItem** ()
- void **signalEscapePreview** ()
- void **signalGotoAlbumAndItem** (const [ItemInfo](#) &)
- void **signalGotoDateAndItem** (const [ItemInfo](#) &)
- void **signalGotoTagAndItem** (int)
- void **signalNextItem** ()
- void **signalPopupTagsView** ()
- void **signalPreviewLoaded** (bool success)
- void **signalPrevItem** ()
- void **signalSlideShowCurrent** ()

## Signals inherited from [Digikam::GraphicsDImgView](#)

- void **activated** ()
- void **contentsMoved** (bool panningFinished)
- void **contentsMoving** (int, int)
- void **leftButtonClicked** ()
- void **leftButtonDoubleClicked** ()
- void **resized** ()
- void **rightButtonClicked** ()
- void **toNextImage** ()
- void **toPreviousImage** ()
- void **viewportRectChanged** (const [QRectF](#) &viewportRect)

## Public Member Functions

- [ItemInfo](#) **getItemInfo** () const
- [ItemPreviewView](#) (QWidget \*const parent, Mode mode=IconViewPreview, [Album](#) \*const currAlbum=nullptr)
- void **reload** ()
- void **setImagePath** (const [QString](#) &path=[QString](#)())
- void **setItemInfo** (const [ItemInfo](#) &info=[ItemInfo](#)()), const [ItemInfo](#) &previous=[ItemInfo](#)(), const [ItemInfo](#) &next=[ItemInfo](#)())
- void **setPreviousNextPaths** (const [QString](#) &previous, const [QString](#) &next)

## Public Member Functions inherited from [Digikam::GraphicsDImgView](#)

- int **contentsX** () const
- int **contentsY** () const
- void **drawText** (QPainter \*p, const [QRectF](#) &rect, const [QString](#) &text)
- void **fitToWindow** ()
- [GraphicsDImgView](#) (QWidget \*const parent=nullptr)
- [GraphicsDImgItem](#) \* **item** () const
- [SinglePhotoPreviewLayout](#) \* **layout** () const
- [DImgPreviewItem](#) \* **previewItem** () const
- void **scrollPointOnPoint** (const [QPointF](#) &scenePos, const [QPoint](#) &viewportPos)
- void **setContentPos** (int x, int y)
- void **setItem** ([GraphicsDImgItem](#) \*const item)
- void **toggleFullScreen** (bool set)
- [QRect](#) **visibleArea** () const

### Protected Member Functions

- bool [acceptsMouseClicked](#) (QMouseEvent \*e) override
- void [dragEnterEvent](#) (QDragEnterEvent \*e) override
- void [dragMoveEvent](#) (QDragMoveEvent \*e) override
- void [dropEvent](#) (QDropEvent \*e) override
- void [enterEvent](#) (QEnterEvent \*) override
- void [leaveEvent](#) (QEvent \*e) override
- void [mousePressEvent](#) (QMouseEvent \*e) override
- void [showEvent](#) (QShowEvent \*e) override

### Protected Member Functions inherited from [Digikam::GraphicsDImgView](#)

- void [continuePanning](#) (const QPoint &pos)
- void [drawForeground](#) (QPainter \*painter, const QRectF &rect) override
- void [finishPanning](#) ()
- void [installPanIcon](#) ()
- void [mouseDoubleClickEvent](#) (QMouseEvent \*) override
- void [mouseMoveEvent](#) (QMouseEvent \*) override
- void [mousePressEvent](#) (QMouseEvent \*) override
- void [mouseReleaseEvent](#) (QMouseEvent \*) override
- void [resizeEvent](#) (QResizeEvent \*) override
- void [scrollContentsBy](#) (int dx, int dy) override
- void [setScaleFitToWindow](#) (bool value)
- void [setShowText](#) (bool value)
- void [startPanning](#) (const QPoint &pos)
- void [wheelEvent](#) (QWheelEvent \*) override

### Additional Inherited Members

### Protected Slots inherited from [Digikam::GraphicsDImgView](#)

- void [slotContentsMoved](#) ()
- void [slotCornerButtonPressed](#) ()
- void [slotPanIconHidden](#) ()
- virtual void [slotPanIconSelectionMoved](#) (const QRect &, bool)

## 6.941.1 Member Function Documentation

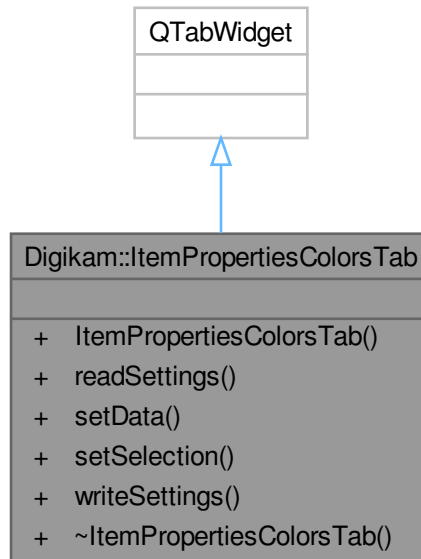
### 6.941.1.1 [acceptsMouseClicked\(\)](#)

```
bool Digikam::ItemPreviewView::acceptsMouseClicked (
    QMouseEvent * e ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::GraphicsDImgView](#).

## 6.942 Digikam::ItemPropertiesColorsTab Class Reference

Inheritance diagram for Digikam::ItemPropertiesColorsTab:

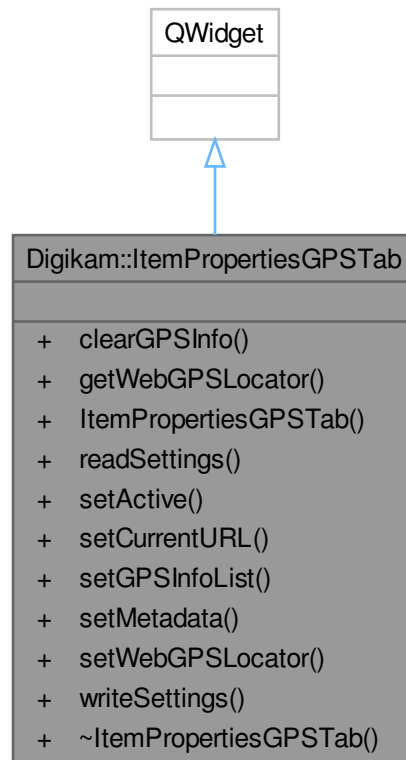


### Public Member Functions

- **ItemPropertiesColorsTab** (`QWidget *const parent`)
- void **readSettings** (`const KConfigGroup &group`)
- void **setData** (`const QUrl &url=QUrl()`, `const QRect &selectionArea=QRect()`, `DImg *const img=nullptr`)
- void **setSelection** (`const QRect &selectionArea`)
- void **writeSettings** (`KConfigGroup &group`)

## 6.943 Digikam::ItemPropertiesGPSTab Class Reference

Inheritance diagram for Digikam::ItemPropertiesGPSTab:



### Public Types

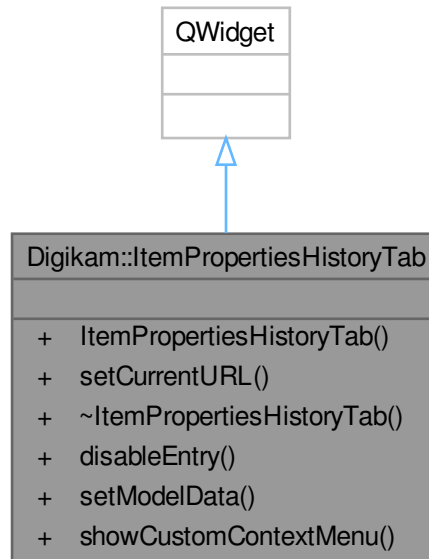
- enum `WebGPSLocator` {  
**MapQuest** = 0, **GoogleMaps**, **BingMaps**, **OpenStreetMap**,  
**LocalizeMaps** }

### Public Member Functions

- void **clearGPSInfo** ()
- int **getWebGPSLocator** () const
- **ItemPropertiesGPSTab** (QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **setActive** (const bool state)
- void **setCurrentURL** (const QUrl &url=QUrl())
- void **setGPSInfoList** (const GPSItemInfo::List &list)
- void **setMetadata** (**DMetadata** \*const meta, const QUrl &url)
- void **setWebGPSLocator** (int locator)
- void **writeSettings** (KConfigGroup &group)

## 6.944 Digikam::ItemPropertiesHistoryTab Class Reference

Inheritance diagram for Digikam::ItemPropertiesHistoryTab:



### Public Slots

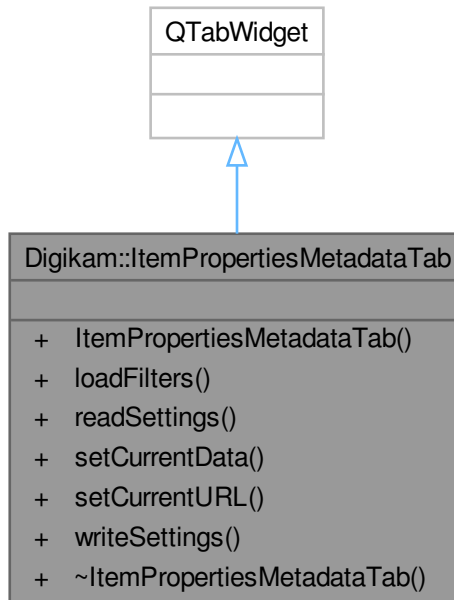
- void **disableEntry** (bool disable)
- void **setModelData** (const QList< [DImageHistory::Entry](#) > &entries)
- void **showCustomContextMenu** (const QPoint &position)

### Public Member Functions

- **ItemPropertiesHistoryTab** (QWidget \*const parent)
- void **setCurrentURL** (const QUrl &url=QUrl())

## 6.945 Digikam::ItemPropertiesMetadataTab Class Reference

Inheritance diagram for Digikam::ItemPropertiesMetadataTab:



### Signals

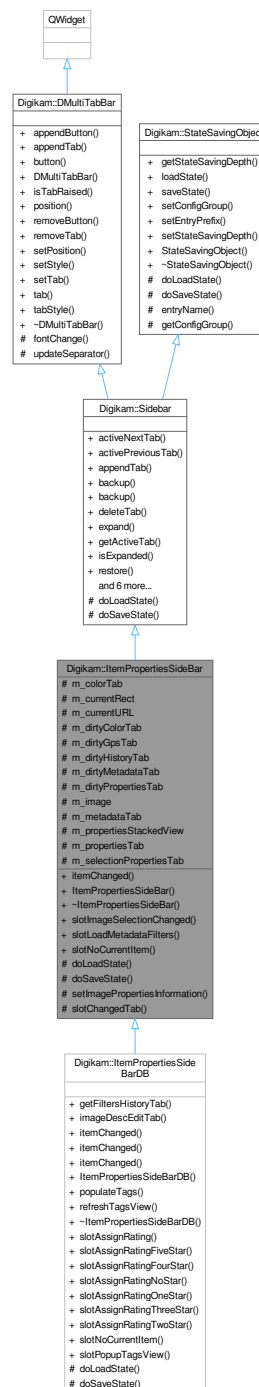
- void **signalSetupExifTool** ()
- void **signalSetupMetadataFilters** (int)

### Public Member Functions

- **ItemPropertiesMetadataTab** (QWidget \*const parent)
- void **loadFilters** ()
- void **readSettings** (const KConfigGroup &group)
- void **setCurrentData** ([DMetadata](#) \*const metadata=nullptr, const QUrl &url=QUrl())
- void **setCurrentURL** (const QUrl &url=QUrl())
- void **writeSettings** (KConfigGroup &group)

## 6.946 Digikam::ItemPropertiesSideBar Class Reference

Inheritance diagram for Digikam::ItemPropertiesSideBar:



### Public Slots

- void **slotImageSelectionChanged** (const QRect &rect)
- void **slotLoadMetadataFilters** ()
- virtual void **slotNoCurrentItem** ()



## Signals

- void **signalSetupExifTool** ()
- void **signalSetupMetadataFilters** (int)

## Signals inherited from [Digikam::Sidebar](#)

- void [signalChangedTab](#) (QWidget \*w)
- void [signalViewChanged](#) ()

## Public Member Functions

- virtual void **itemChanged** (const QUrl &url, const QRect &rect=QRect(), [DImg](#) \*const img=nullptr)
- **ItemPropertiesSideBar** (QWidget \*const parent, [SidebarSplitter](#) \*const splitter, Qt::Edge side=Qt::LeftEdge, bool mimimizedDefault=false)

## Public Member Functions inherited from [Digikam::Sidebar](#)

- void [activeNextTab](#) ()
- void [activePreviousTab](#) ()
- void [appendTab](#) (QWidget \*const w, const QIcon &pic, const QString &title)
- void [backup](#) ()
- void [backup](#) (const QList< QWidget \* > &thirdWidgetsToBackup, QList< int > \*const sizes)
- void [deleteTab](#) (QWidget \*const w)
- void [expand](#) ()
- QWidget \* [getActiveTab](#) () const
- bool [isExpanded](#) () const
- void [restore](#) ()
- void [restore](#) (const QList< QWidget \* > &thirdWidgetsToRestore, const QList< int > &sizes)
- void [setActiveTab](#) (QWidget \*const w)
- void [shrink](#) ()
- [Sidebar](#) (QWidget \*const parent, [SidebarSplitter](#) \*const sp, Qt::Edge side=Qt::LeftEdge, bool minimized↔ Default=false)
- [SidebarSplitter](#) \* **splitter** () const

## Public Member Functions inherited from [Digikam::DMultiTabBar](#)

- void [appendButton](#) (const QIcon &pic, int id=-1, QMenu \*const popup=nullptr, const QString &not\_used\_↔ yet=QString())
- void [appendTab](#) (const QIcon &pic, int id=-1, const QString &text=QString())
- [DMultiTabBarButton](#) \* [button](#) (int id) const
- **DMultiTabBar** (Qt::Edge pos, QWidget \*const parent=nullptr)
- bool [isTabRaised](#) (int id) const
- Qt::Edge [position](#) () const
- void [removeButton](#) (int id)
- void [removeTab](#) (int id)
- void [setPosition](#) (Qt::Edge pos)
- void [setStyle](#) ([TextStyle](#) style)
- void [setTab](#) (int id, bool state)
- [DMultiTabBarTab](#) \* [tab](#) (int id) const
- [TextStyle](#) [tabStyle](#) () const

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) `getStateSavingDepth ()` const
- void `loadState ()`
- void `saveState ()`
- virtual void `setConfigGroup (const KConfigGroup &group)`
- virtual void `setEntryPrefix (const QString &prefix)`
- void `setStateSavingDepth (const StateSavingDepth depth)`
- [StateSavingObject](#) (`QObject *const host`)
- virtual `~StateSavingObject ()`

## Protected Slots

- virtual void `slotChangedTab (QWidget *tab)`

## Protected Member Functions

- void `doLoadState ()` override
- void `doSaveState ()` override
- virtual void `setImagePropertiesInformation (const QUrl &url)`

## Protected Member Functions inherited from [Digikam::Sidebar](#)

- void `doLoadState ()` override
- void `doSaveState ()` override

## Protected Member Functions inherited from [Digikam::DMultiTabBar](#)

- virtual void `fontChange (const QFont &)`
- void `updateSeparator ()`

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString `entryName (const QString &base)` const
- KConfigGroup `getConfigGroup ()` const

## Protected Attributes

- [ItemPropertiesColorsTab](#) \* `m_colorTab` = nullptr
- QRect `m_currentRect`
- QUrl `m_currentURL`
- bool `m_dirtyColorTab` = false
- bool `m_dirtyGpsTab` = false
- bool `m_dirtyHistoryTab` = false
- bool `m_dirtyMetadataTab` = false
- bool `m_dirtyPropertiesTab` = false
- [DImg](#) \* `m_image` = nullptr
- [ItemPropertiesMetadataTab](#) \* `m_metadataTab` = nullptr
- QStackedWidget \* `m_propertiesStackedView` = nullptr
- [ItemPropertiesTab](#) \* `m_propertiesTab` = nullptr
- [ItemSelectionPropertiesTab](#) \* `m_selectionPropertiesTab` = nullptr

## Additional Inherited Members

### Public Types inherited from [Digikam::DMultiTabBar](#)

- enum [TextStyle](#) { [ActiveIconText](#) = 0 , [AllIconsText](#) = 2 }

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## 6.946.1 Member Function Documentation

### 6.946.1.1 doLoadState()

```
void Digikam::ItemPropertiesSideBar::doLoadState ( ) [override], [protected], [virtual]
```

load the last view state from disk - called by [StateSavingObject::loadState\(\)](#)

Implements [Digikam::StateSavingObject](#).

Reimplemented in [Digikam::ItemPropertiesSideBarDB](#).

### 6.946.1.2 doSaveState()

```
void Digikam::ItemPropertiesSideBar::doSaveState ( ) [override], [protected], [virtual]
```

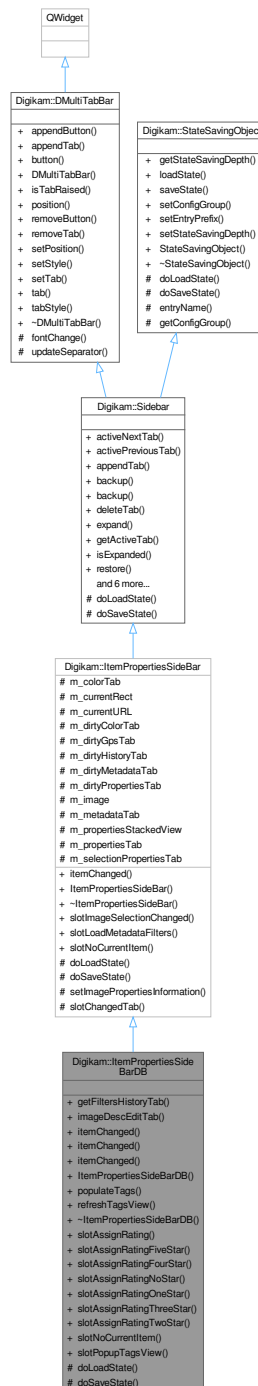
save the view state to disk - called by [StateSavingObject::saveState\(\)](#)

Implements [Digikam::StateSavingObject](#).

Reimplemented in [Digikam::ItemPropertiesSideBarDB](#).

## 6.947 Digikam::ItemPropertiesSideBarDB Class Reference

Inheritance diagram for Digikam::ItemPropertiesSideBarDB:



### Public Slots

- void **slotAssignRating** (int rating)
- void **slotAssignRatingFiveStar** ()

- void **slotAssignRatingFourStar** ()
- void **slotAssignRatingNoStar** ()
- void **slotAssignRatingOneStar** ()
- void **slotAssignRatingThreeStar** ()
- void **slotAssignRatingTwoStar** ()
- void **slotNoCurrentItem** () override
- void **slotPopupTagsView** ()

### Public Slots inherited from [Digikam::ItemPropertiesSideBar](#)

- void **slotImageSelectionChanged** (const QRect &rect)
- void **slotLoadMetadataFilters** ()
- virtual void **slotNoCurrentItem** ()

### Signals

- void **signalFirstItem** ()
- void **signalLastItem** ()
- void **signalNextItem** ()
- void **signalPrevItem** ()
- void **signalRightSideBarBusy** (bool busy)

### Signals inherited from [Digikam::ItemPropertiesSideBar](#)

- void **signalSetupExifTool** ()
- void **signalSetupMetadataFilters** (int)

### Signals inherited from [Digikam::Sidebar](#)

- void [signalChangedTab](#) (QWidget \*w)
- void [signalViewChanged](#) ()

### Public Member Functions

- [ItemPropertiesVersionsTab](#) \* **getFiltersHistoryTab** () const  
*This is for image editor to be able to update the filter list in sidebar.*
- [ItemDescEditTab](#) \* **imageDescEditTab** () const
- virtual void **itemChanged** (const [ItemInfo](#) &info, const QRect &rect=QRect(), [DImg](#) \*const img=nullptr, const [DImageHistory](#) &history=[DImageHistory](#)())
- virtual void **itemChanged** (const [ItemInfoList](#) &infos, const [ItemInfoList](#) &allInfos)
- void **itemChanged** (const QUrl &url, const QRect &rect=QRect(), [DImg](#) \*const img=nullptr) override
- **ItemPropertiesSideBarDB** (QWidget \*const parent, [SidebarSplitter](#) \*const splitter, Qt::Edge side=Qt::LeftEdge, bool mimimizedDefault=false)
- void **populateTags** ()
- void **refreshTagsView** ()

### Public Member Functions inherited from [Digikam::ItemPropertiesSideBar](#)

- **ItemPropertiesSideBar** (QWidget \*const parent, [SidebarSplitter](#) \*const splitter, Qt::Edge side=Qt::LeftEdge, bool mimimizedDefault=false)

## Public Member Functions inherited from [Digikam::Sidebar](#)

- void [activeNextTab](#) ()
- void [activePreviousTab](#) ()
- void [appendTab](#) (QWidget \*const w, const QIcon &pic, const QString &title)
- void [backup](#) ()
- void [backup](#) (const QList< QWidget \* > &thirdWidgetsToBackup, QList< int > \*const sizes)
- void [deleteTab](#) (QWidget \*const w)
- void [expand](#) ()
- QWidget \* [getActiveTab](#) () const
- bool [isExpanded](#) () const
- void [restore](#) ()
- void [restore](#) (const QList< QWidget \* > &thirdWidgetsToRestore, const QList< int > &sizes)
- void [setActiveTab](#) (QWidget \*const w)
- void [shrink](#) ()
- [Sidebar](#) (QWidget \*const parent, [SidebarSplitter](#) \*const sp, Qt::Edge side=Qt::LeftEdge, bool minimized↔ Default=false)
- [SidebarSplitter](#) \* [splitter](#) () const

## Public Member Functions inherited from [Digikam::DMultiTabBar](#)

- void [appendButton](#) (const QIcon &pic, int id=-1, QMenu \*const popup=nullptr, const QString &not\_used\_↔ yet=QString())
- void [appendTab](#) (const QIcon &pic, int id=-1, const QString &text=QString())
- [DMultiTabBarButton](#) \* [button](#) (int id) const
- [DMultiTabBar](#) (Qt::Edge pos, QWidget \*const parent=nullptr)
- bool [isTabRaised](#) (int id) const
- Qt::Edge [position](#) () const
- void [removeButton](#) (int id)
- void [removeTab](#) (int id)
- void [setPosition](#) (Qt::Edge pos)
- void [setStyle](#) (TextStyle style)
- void [setTab](#) (int id, bool state)
- [DMultiTabBarTab](#) \* [tab](#) (int id) const
- [TextStyle](#) [tabStyle](#) () const

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Protected Member Functions

- void [doLoadState](#) () override
- void [doSaveState](#) () override

## Protected Member Functions inherited from [Digikam::Sidebar](#)

- void [doLoadState](#) () override
- void [doSaveState](#) () override

## Protected Member Functions inherited from [Digikam::DMultiTabBar](#)

- virtual void [fontChange](#) (const QFont &)
- void [updateSeparator](#) ()

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## Additional Inherited Members

## Public Types inherited from [Digikam::DMultiTabBar](#)

- enum [TextStyle](#) { [ActiveIconText](#) = 0 , [AllIconsText](#) = 2 }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Slots inherited from [Digikam::ItemPropertiesSideBar](#)

- virtual void [slotChangedTab](#) (QWidget \*tab)

## Protected Attributes inherited from [Digikam::ItemPropertiesSideBar](#)

- [ItemPropertiesColorsTab](#) \* [m\\_colorTab](#) = nullptr
- QRect [m\\_currentRect](#)
- QUrl [m\\_currentURL](#)
- bool [m\\_dirtyColorTab](#) = false
- bool [m\\_dirtyGpsTab](#) = false
- bool [m\\_dirtyHistoryTab](#) = false
- bool [m\\_dirtyMetadataTab](#) = false
- bool [m\\_dirtyPropertiesTab](#) = false
- [DImg](#) \* [m\\_image](#) = nullptr
- [ItemPropertiesMetadataTab](#) \* [m\\_metadataTab](#) = nullptr
- QStackedWidget \* [m\\_propertiesStackedView](#) = nullptr
- [ItemPropertiesTab](#) \* [m\\_propertiesTab](#) = nullptr
- [ItemSelectionPropertiesTab](#) \* [m\\_selectionPropertiesTab](#) = nullptr

## 6.947.1 Member Function Documentation

### 6.947.1.1 doLoadState()

```
void Digikam::ItemPropertiesSideBarDB::doLoadState ( ) [override], [protected], [virtual]
```

load the last view state from disk - called by [StateSavingObject::loadState\(\)](#)

Reimplemented from [Digikam::ItemPropertiesSideBar](#).

### 6.947.1.2 doSaveState()

```
void Digikam::ItemPropertiesSideBarDB::doSaveState ( ) [override], [protected], [virtual]
```

save the view state to disk - called by [StateSavingObject::saveState\(\)](#)

Reimplemented from [Digikam::ItemPropertiesSideBar](#).

### 6.947.1.3 itemChanged()

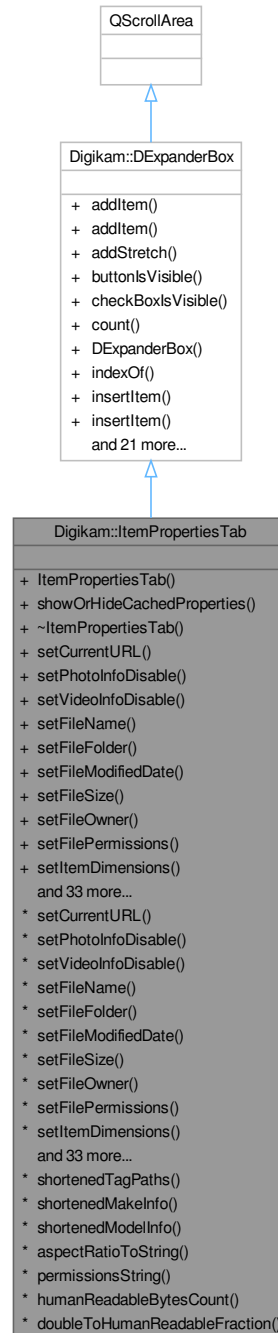
```
void Digikam::ItemPropertiesSideBarDB::itemChanged (
    const QUrl & url,
    const QRect & rect = QRect(),
    DImg *const img = nullptr ) [override], [virtual]
```

Reimplemented from [Digikam::ItemPropertiesSideBar](#).



## 6.948 Digikam::ItemPropertiesTab Class Reference

Inheritance diagram for Digikam::ItemPropertiesTab:



### Classes

- class [Private](#)

## Public Types

- enum **Section** {  
**FileProperties** = 0 , **ImageProperties** , **PhotoProperties** , **VideoProperties** ,  
**digiKamProperties** , **TagsProperties** , **LocationProperties** , **RightProperties** }

## Public Member Functions

- **ItemPropertiesTab** (QWidget \*const parent)
- void **showOrHideCachedProperties** ()
- void **setCurrentURL** (const QUrl &url=QUrl())  
*Setter methods (itempropertistab\_setters.cpp)*
- void **setPhotoInfoDisable** (const bool b)
- void **setVideoInfoDisable** (const bool b)
- void **setFileName** (const QString &str)
- void **setFileFolder** (const QString &str)
- void **setFileModifiedDate** (const QString &str)
- void **setFileSize** (const QString &str)
- void **setFileOwner** (const QString &str)
- void **setFilePermissions** (const QString &str)
- void **setItemDimensions** (const QString &str)
- void **setImageRatio** (const QString &str)
- void **setImageMime** (const QString &str)
- void **setImageBitDepth** (const QString &str)
- void **setImageColorMode** (const QString &str)
- void **setHasSidecar** (const QString &str)
- void **setHasGPSInfo** (const QString &str)
- void **setVersionnedInfo** (const QString &str)
- void **setGroupedInfo** (const QString &str)
- void **setPhotoMake** (const QString &str)
- void **setPhotoModel** (const QString &str)
- void **setPhotoDateTime** (const QString &str)
- void **setPhotoLens** (const QString &str)
- void **setPhotoAperture** (const QString &str)
- void **setPhotoFocalLength** (const QString &str)
- void **setPhotoExposureTime** (const QString &str)
- void **setPhotoSensitivity** (const QString &str)
- void **setPhotoExposureMode** (const QString &str)
- void **setPhotoFlash** (const QString &str)
- void **setPhotoWhiteBalance** (const QString &str)
- void **setVideoAspectRatio** (const QString &str)
- void **setVideoAudioBitRate** (const QString &str)
- void **setVideoAudioChannelType** (const QString &str)
- void **setVideoAudioCodec** (const QString &str)
- void **setVideoDuration** (const QString &str)
- void **setVideoFrameRate** (const QString &str)
- void **setVideoVideoCodec** (const QString &str)
- void **setTitle** (const QString &str)
- void **setCaption** (const QString &str)
- void **setPickLabel** (int pickId)
- void **setColorLabel** (int colorId)
- void **setRating** (int rating)
- void **setTags** (const QStringList &>tagPaths, const QStringList &>tagNames=QStringList(), const QStringList &peopleTagPaths=QStringList(), const QStringList &peopleTagNames=QStringList())
- void **setTemplate** (const [Template](#) &t)

## Public Member Functions inherited from Digikam::DExpanderBox

- void **addItem** (QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
  - void **addItem** (QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
  - void **addStretch** ()
  - bool **buttonIsVisible** (int index) const
  - bool **checkboxIsVisible** (int index) const
  - int **count** () const
  - **DExpanderBox** (QWidget \*const parent=nullptr)
  - int **indexOf** (DLabelExpander \*const widget) const
  - void **insertItem** (int index, QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
  - void **insertItem** (int index, QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
  - void **insertStretch** (int index)
  - bool **isChecked** (int index) const
  - bool **isItemEnabled** (int index) const
  - bool **isItemExpanded** (int index) const
  - QIcon **itemIcon** (int index) const
  - QString **itemText** (int index) const
  - QString **itemToolTip** (int index) const
  - virtual void **readSettings** (KConfigGroup &group)
  - void **removeItem** (int index)
  - void **setButtonIcon** (int index, const QIcon &icon)
  - void **setButtonVisible** (int index, bool b)
  - void **setCheckBoxVisible** (int index, bool b)
  - void **setChecked** (int index, bool b)
  - void **setItemEnabled** (int index, bool enabled)
  - void **setItemExpanded** (int index, bool b)
  - void **setItemIcon** (int index, const QIcon &icon)
  - void **setItemText** (int index, const QString &txt)
  - void **setItemToolTip** (int index, const QString &tip)
  - DLabelExpander \* **widget** (int index) const
  - virtual void **writeSettings** (KConfigGroup &group)
- 
- static QStringList **shortenedTagPaths** (const QStringList &tagPaths, QList< QVariant > \*identifiers=nullptr)  
*Helper methods (itempropertiestab\_helpers.cpp)*
  - static void **shortenedMakeInfo** (QString &make)
  - static void **shortenedModelInfo** (QString &model)
  - static bool **aspectRatioToString** (int width, int height, QString &arString)
  - static QString **permissionsString** (const QFileInfo &fi)
  - static QString **humanReadableBytesCount** (qint64 bytes, bool si=false)

## Additional Inherited Members

## Signals inherited from Digikam::DExpanderBox

- void **signalItemButtonPressed** (int index)
- void **signalItemExpanded** (int index, bool b)
- void **signalItemToggled** (int index, bool b)

## 6.948.1 Member Function Documentation

### 6.948.1.1 aspectRatioToString()

```
bool Digikam::ItemPropertiesTab::aspectRatioToString (
    int width,
    int height,
    QString & arString ) [static]
```

Write a string with aspect ratio information formatted

### 6.948.1.2 humanReadableBytesCount()

```
QString Digikam::ItemPropertiesTab::humanReadableBytesCount (
    quint64 bytes,
    bool si = false ) [static]
```

Return an human readable string of file size in 'bytes'. If 'si' is true, a decade of bytes is interpreted on base of 1000 byte, else 1024.

### 6.948.1.3 permissionsString()

```
QString Digikam::ItemPropertiesTab::permissionsString (
    const QFileInfo & fi ) [static]
```

Return file permissions string.

### 6.948.1.4 shortenedMakeInfo()

```
void Digikam::ItemPropertiesTab::shortenedMakeInfo (
    QString & make ) [static]
```

This methods shortens make an model camera info to prevent bloating GUI See bug #265231 for details.

### 6.948.1.5 shortenedTagPaths()

```
QStringList Digikam::ItemPropertiesTab::shortenedTagPaths (
    const QStringList & tagPaths,
    QList< QVariant > * identifiers = nullptr ) [static]
```

Shortens the tag paths by sorting and then cutting identical paths from the second and following paths (only the first item gives the full path). If you want to retain information about which tag path is sorted where, you can optionally give a QVariant list. This list shall contain an identifier for the tag path at the same index and will be resorted as the returned list.

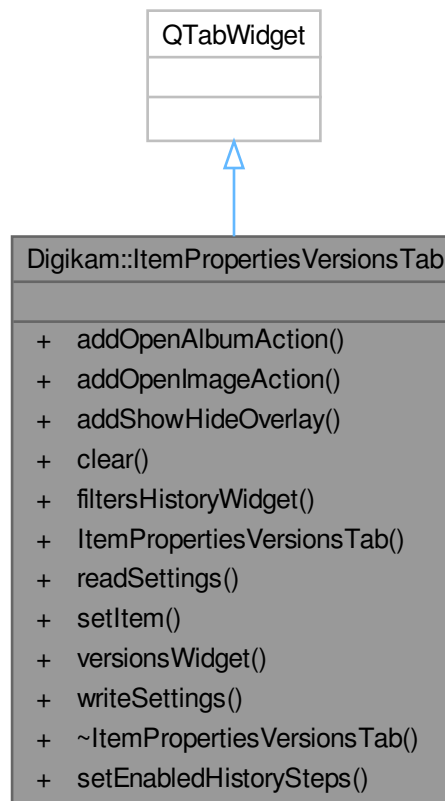
## 6.949 Digikam::ItemPropertiesTab::Private Class Reference

### Public Attributes

- [DTextLabelName](#) \* **caption** = nullptr
- [DToolTipStyleSheet](#) **cnt**
- [DTextLabelName](#) \* **colorLabel** = nullptr
- [DTextLabelName](#) \* **grouped** = nullptr
- [QLabel](#) \* **labelCaption** = nullptr
- [DTextLabelValue](#) \* **labelColorLabel** = nullptr
- [DTextLabelValue](#) \* **labelFile** = nullptr
- [DTextLabelValue](#) \* **labelFileModifiedDate** = nullptr
- [DTextLabelValue](#) \* **labelFileOwner** = nullptr
- [DTextLabelValue](#) \* **labelFilePermissions** = nullptr
- [DTextLabelValue](#) \* **labelFileSize** = nullptr
- [DTextLabelValue](#) \* **labelFolder** = nullptr
- [DTextLabelValue](#) \* **labelGroupedInfo** = nullptr
- [DTextLabelValue](#) \* **labelHasGPSInfo** = nullptr
- [DTextLabelValue](#) \* **labelHasSidecar** = nullptr
- [DTextLabelValue](#) \* **labelImageBitDepth** = nullptr
- [DTextLabelValue](#) \* **labelImageColorMode** = nullptr
- [DTextLabelValue](#) \* **labelImageDimensions** = nullptr
- [DTextLabelValue](#) \* **labelImageMime** = nullptr
- [DTextLabelValue](#) \* **labelImageRatio** = nullptr
- [QLabel](#) \* **labelLocation** = nullptr
- [QLabel](#) \* **labelPeoples** = nullptr
- [DTextLabelValue](#) \* **labelPhotoAperture** = nullptr
- [DTextLabelValue](#) \* **labelPhotoDateTime** = nullptr
- [DTextLabelValue](#) \* **labelPhotoExposureMode** = nullptr
- [DTextLabelValue](#) \* **labelPhotoExposureTime** = nullptr
- [DTextLabelValue](#) \* **labelPhotoFlash** = nullptr
- [DTextLabelValue](#) \* **labelPhotoFocalLength** = nullptr
- [DTextLabelValue](#) \* **labelPhotoLens** = nullptr
- [DTextLabelValue](#) \* **labelPhotoMake** = nullptr
- [DTextLabelValue](#) \* **labelPhotoModel** = nullptr
- [DTextLabelValue](#) \* **labelPhotoSensitivity** = nullptr
- [DTextLabelValue](#) \* **labelPhotoWhiteBalance** = nullptr
- [DTextLabelValue](#) \* **labelPickLabel** = nullptr
- [DTextLabelValue](#) \* **labelRating** = nullptr
- [QLabel](#) \* **labelRights** = nullptr
- [DTextLabelValue](#) \* **labelSymlink** = nullptr
- [QLabel](#) \* **labelTags** = nullptr
- [QLabel](#) \* **labelTitle** = nullptr
- [DTextLabelValue](#) \* **labelVersionnedInfo** = nullptr
- [DTextLabelValue](#) \* **labelVideoAspectRatio** = nullptr
- [DTextLabelValue](#) \* **labelVideoAudioBitRate** = nullptr
- [DTextLabelValue](#) \* **labelVideoAudioChannelType** = nullptr
- [DTextLabelValue](#) \* **labelVideoAudioCodec** = nullptr
- [DTextLabelValue](#) \* **labelVideoDuration** = nullptr
- [DTextLabelValue](#) \* **labelVideoFrameRate** = nullptr
- [DTextLabelValue](#) \* **labelVideoVideoCodec** = nullptr
- [DTextLabelName](#) \* **peoples** = nullptr
- [DTextLabelName](#) \* **pickLabel** = nullptr
- [DTextLabelName](#) \* **rating** = nullptr
- [DTextLabelName](#) \* **tags** = nullptr
- [DTextLabelName](#) \* **title** = nullptr
- [DTextLabelName](#) \* **versionned** = nullptr

## 6.950 Digikam::ItemPropertiesVersionsTab Class Reference

Inheritance diagram for Digikam::ItemPropertiesVersionsTab:



### Public Slots

- void `setEnabledHistorySteps` (int count)

### Signals

- void `actionTriggered` (const [ItemInfo](#) &info)
- void `imageSelected` (const [ItemInfo](#) &info)

### Public Member Functions

- void `addOpenAlbumAction` (const [ItemModel](#) \*referenceModel)
- void `addOpenImageAction` ()
- void `addShowHideOverlay` ()
- void `clear` ()
- [FiltersHistoryWidget](#) \* `filtersHistoryWidget` () const
- `ItemPropertiesVersionsTab` (QWidget \*const parent)
- void `readSettings` (KConfigGroup &group)
- void `setItem` (const [ItemInfo](#) &info, const [DImageHistory](#) &history)
- [VersionsWidget](#) \* `versionsWidget` () const
- void `writeSettings` (KConfigGroup &group)

## 6.951 Digikam::ItemQueryBuilder Class Reference

### Public Member Functions

- QString **buildQuery** (const QString &q, QList< QVariant > \*boundValues, [ItemQueryPostHooks](#) \*const hooks) const
- QString **buildQueryFromUrl** (const QUrl &url, QList< QVariant > \*boundValues) const
- QString **buildQueryFromXml** (const QString &xml, QList< QVariant > \*boundValues, [ItemQueryPostHooks](#) \*const hooks) const
- QString **convertFromUrlToXml** (const QUrl &url) const
- void **setImageTagPropertiesJoined** (bool isJoined)

### Static Public Member Functions

- static void **addNoEffectContent** (QString &sql, SearchXml::Operator op)
- static void **addSqlOperator** (QString &sql, SearchXml::Operator op, bool isFirst)
- static void **addSqlRelation** (QString &sql, SearchXml::Relation rel)

### Protected Member Functions

- bool **buildField** (QString &sql, [SearchXmlCachingReader](#) &reader, const QString &name, QList< QVariant > \*boundValues, [ItemQueryPostHooks](#) \*const hooks) const
- void **buildGroup** (QString &sql, [SearchXmlCachingReader](#) &reader, QList< QVariant > \*boundValues, [ItemQueryPostHooks](#) \*const hooks) const
- QString **possibleDate** (const QString &str, bool &exact) const

### Protected Attributes

- bool **m\_imageTagPropertiesJoined** = false
- QString **m\_longMonths** [12]
- QString **m\_shortMonths** [12]

## 6.951.1 Member Function Documentation

### 6.951.1.1 setImageTagPropertiesJoined()

```
void Digikam::ItemQueryBuilder::setImageTagPropertiesJoined (
    bool isJoined )
```

Use for special queries where ImageTagProperties table is JOIN'ed. (Default: false)

## 6.952 Digikam::ItemQueryPostHook Class Reference

### Public Member Functions

- virtual bool **checkPosition** (double, double)
- [ItemQueryPostHook](#) ()=default

## 6.952.1 Constructor & Destructor Documentation

### 6.952.1.1 ItemQueryPostHook()

```
Digikam::ItemQueryPostHook::ItemQueryPostHook ( ) [default]
```

This is the single hook, ItemQueryPostHookS is the container.

## 6.953 Digikam::ItemQueryPostHooks Class Reference

### Public Member Functions

- void [addHook](#) ([ItemQueryPostHook](#) \*const hook)
- bool [checkPosition](#) (double latitudeNumber, double longitudeNumber)

### Protected Attributes

- [QList](#)< [ItemQueryPostHook](#) \* > **m\_postHooks**

## 6.953.1 Member Function Documentation

### 6.953.1.1 addHook()

```
void Digikam::ItemQueryPostHooks::addHook (
    ItemQueryPostHook *const hook )
```

Called by [ItemQueryBuilder](#). Ownership of the object is passed.

### 6.953.1.2 checkPosition()

```
bool Digikam::ItemQueryPostHooks::checkPosition (
    double latitudeNumber,
    double longitudeNumber )
```

Call this method after passing the object to buildQuery and executing the statement. Returns true if the search is matched.



## 6.954 Digikam::ItemRatingOverlay Class Reference

Inheritance diagram for Digikam::ItemRatingOverlay:



### Signals

- void **ratingEdited** (const QList< QModelIndex > &indexes, int rating)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- **ItemRatingOverlay** (QObject \*const parent)
- [RatingWidget](#) \* **ratingWidget** () const

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotDataChanged** (const QModelIndex &, const QModelIndex &)
- void **slotRatingChanged** (int)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

## Protected Member Functions

- QWidget \* **createWidget** () override
- void **hide** () override
- void **setActive** (bool) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **updateRating** ()
- void **visualChange** () override
- void **widgetEnterEvent** () override
- void **widgetLeaveEvent** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual bool [checkIndex](#) (const QModelIndex &index) const
- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- void [widgetLeaveNotifyMultiple](#) ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

## Protected Attributes

- QPersistentModelIndex [m\\_index](#)

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.954.1 Member Function Documentation

### 6.954.1.1 [createWidget\(\)](#)

```
QWidget * Digikam::ItemRatingOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.954.1.2 [hide\(\)](#)

```
void Digikam::ItemRatingOverlay::hide ( ) [override], [protected], [virtual]
```

Called when the widget shall be hidden (mouse cursor left index, viewport, uninstalled etc.). Default implementation [hide\(\)](#)s [m\\_widget](#).

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.954.1.3 setActive()

```
void Digikam::ItemRatingOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.954.1.4 slotEntered()

```
void Digikam::ItemRatingOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.954.1.5 visualChange()

```
void Digikam::ItemRatingOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

### 6.954.1.6 widgetEnterEvent()

```
void Digikam::ItemRatingOverlay::widgetEnterEvent ( ) [override], [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.954.1.7 widgetLeaveEvent()

```
void Digikam::ItemRatingOverlay::widgetLeaveEvent ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.955 Digikam::ItemRotateOverlay Class Reference

Inheritance diagram for Digikam::ItemRotateOverlay:



### Signals

- void **signalRotate** (const QList< QModelIndex > &indexes)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- ItemRotateOverlayDirection **direction** () const
- bool **isLeft** () const
- bool **isRight** () const
- **ItemRotateOverlay** (ItemRotateOverlayDirection dir, QObject \*const parent)
- void **setActive** (bool active) override

## Public Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- [ItemViewHoverButton](#) \* **button** () const
- [HoverButtonDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Static Public Member Functions

- static [ItemRotateOverlay](#) \* **left** (QObject \*const parent)
- static [ItemRotateOverlay](#) \* **right** (QObject \*const parent)

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- [ItemViewHoverButton](#) \* **createButton** () override
- void **updateButton** (const QModelIndex &index) override
- void **widgetEnterEvent** () override
- void **widgetLeaveEvent** () override

### Protected Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- QWidget \* [createWidget](#) () override
- void [visualChange](#) () override

### Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- void [widgetLeaveNotifyMultiple](#) ()

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

### Additional Inherited Members

### Protected Slots inherited from [Digikam::HoverButtonDelegateOverlay](#)

- void [slotEntered](#) (const QModelIndex &index) override
- void [slotReset](#) () override

### Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void [slotEntered](#) (const QModelIndex &index)
- virtual void [slotLayoutChanged](#) ()
- virtual void [slotReset](#) ()
- virtual void [slotRowsRemoved](#) (const QModelIndex &parent, int start, int end)
- virtual void [slotViewportEntered](#) ()

### Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

### Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.955.1 Member Function Documentation

### 6.955.1.1 checkIndex()

```
bool Digikam::ItemRotateOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.955.1.2 createButton()

```
ItemViewHoverButton * Digikam::ItemRotateOverlay::createButton ( ) [override], [protected],
[virtual]
```

Create your widget here. Pass view() as parent.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.955.1.3 setActive()

```
void Digikam::ItemRotateOverlay::setActive (
    bool active ) [override], [virtual]
```

Will call [createButton\(\)](#).

Reimplemented from [Digikam::HoverButtonDelegateOverlay](#).

### 6.955.1.4 updateButton()

```
void Digikam::ItemRotateOverlay::updateButton (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.955.1.5 widgetEnterEvent()

```
void Digikam::ItemRotateOverlay::widgetEnterEvent ( ) [override], [protected], [virtual]
```

Called when a QEvent::Enter resp. QEvent::Leave event for the widget is received. The default implementation does nothing.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).



## 6.955.1.6 widgetLeaveEvent()

```
void Digikam::ItemRotateOverlay::widgetLeaveEvent ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.956 Digikam::ItemRotateOverlayButton Class Reference

Inheritance diagram for Digikam::ItemRotateOverlayButton:



### Public Member Functions

- **ItemRotateOverlayButton** (ItemRotateOverlayDirection dir, QAbstractItemView \*const parentView)
- QSize [sizeHint](#) () const override

### Public Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- QModelIndex [index](#) () const
- void [initIcon](#) ()
- **ItemViewHoverButton** (QAbstractItemView \*const parentView)
- void [reset](#) ()
- void [setIndex](#) (const QModelIndex &index)
- void [setVisible](#) (bool visible) override

### Protected Member Functions

- QIcon [icon](#) () override
- void [updateToolTip](#) () override

### Protected Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- void [enterEvent](#) (QEnterEvent \*event)
- void [leaveEvent](#) (QEvent \*event)
- void [paintEvent](#) (QPaintEvent \*event)
- void [setup](#) ()

### Protected Attributes

- ItemRotateOverlayDirection const [m\\_direction](#)

### Protected Attributes inherited from [Digikam::ItemViewHoverButton](#)

- QTimerLine \* [m\\_fadingTimeLine](#) = nullptr
- int [m\\_fadingValue](#) = 0
- QIcon [m\\_icon](#)
- QPersistentModelIndex [m\\_index](#)
- bool [m\\_isHovered](#) = false

### Additional Inherited Members

### Protected Slots inherited from [Digikam::ItemViewHoverButton](#)

- void [refreshIcon](#) ()
- void [setFadingValue](#) (int value)
- void [startFading](#) ()
- void [stopFading](#) ()

## 6.956.1 Member Function Documentation

### 6.956.1.1 icon()

```
QIcon Digikam::ItemRotateOverlayButton::icon ( ) [override], [protected], [virtual]
```

Return your icon here. Will be queried again on toggle.

Implements [Digikam::ItemViewHoverButton](#).

### 6.956.1.2 sizeHint()

```
QSize Digikam::ItemRotateOverlayButton::sizeHint ( ) const [override], [virtual]
```

Reimplement to match the size of your icon

Implements [Digikam::ItemViewHoverButton](#).

### 6.956.1.3 updateToolTip()

```
void Digikam::ItemRotateOverlayButton::updateToolTip ( ) [override], [protected], [virtual]
```

Optionally update tooltip here. Will be called again on state change.

Reimplemented from [Digikam::ItemViewHoverButton](#).

## 6.957 Digikam::ItemScanInfo Class Reference

### Public Member Functions

- bool **isNull** ( ) const

### Public Attributes

- int **albumID** = 0
- DatabaseItem::Category **category** = DatabaseItem::UndefinedCategory
- qlonglong **fileSize** = 0
- qlonglong **id** = 0
- QString **itemName**
- QDateTime **modificationDate**
- DatabaseItem::Status **status** = DatabaseItem::UndefinedStatus
- QString **uniqueHash**

## 6.958 Digikam::ItemScanner Class Reference

### Classes

- class [Private](#)

## Public Types

- enum **ScanMode** { **NewScan** , **ModifiedScan** , **Rescan** , **CleanScan** }

## Public Member Functions

- const [ItemScanInfo](#) & [itemScanInfo](#) () const
- [ItemScanner](#) (const [QFileInfo](#) &info)
- [ItemScanner](#) (const [QFileInfo](#) &info, const [ItemScanInfo](#) &iteminfo)
- [ItemScanner](#) (qulonglong imageid)
- void [loadFromDisk](#) ()
- void [setCategory](#) ([DatabaseItem::Category](#) category)

## Static Public Member Functions

- static [QString](#) [formatToString](#) (const [QString](#) &format)

## Operations on History Metadata

- bool [hasHistoryToResolve](#) () const
- void [scanBalooInfo](#) ()
  - *scanBalooInfo - retrieve tags, comments and rating from Baloo Desktop service.*
- static bool [resolveImageHistory](#) (qulonglong id, [QList](#)< qulonglong > \*needTaggingIds=nullptr)
- static bool **resolveImageHistory** (qulonglong imageid, const [QString](#) &historyXml, [QList](#)< qulonglong > \*needTaggingIds=nullptr)
- static void [tagItemHistoryGraph](#) (qulonglong id)
- static [DImageHistory](#) [resolvedImageHistory](#) (const [DImageHistory](#) &history, bool mustBeAvailable=false)
- static bool [sameReferredImage](#) (const [HistoryImageId](#) &id1, const [HistoryImageId](#) &id2)
- static [QList](#)< qulonglong > [resolveHistoryImageId](#) (const [HistoryImageId](#) &historyId)
- void **scanImageHistory** ()
- void **commitImageHistory** ()
- void **scanImageHistoryIfModified** ()
- [QString](#) **uniqueHash** () const

## Operations with Database

- void [newFile](#) (int albumId)
- void [newFileFullScan](#) (int albumId)
- void [rescan](#) ()
- void [cleanScan](#) ()
- void [commit](#) ()
- qulonglong [id](#) () const
- void [copiedFrom](#) (int albumId, qulonglong srcId)
- static void [sortByProximity](#) ([QList](#)< [ItemInfo](#) > &infos, const [ItemInfo](#) &subject)
- bool **copyFromSource** (qulonglong src)
- void **commitCopyImageAttributes** ()
- void **cleanDatabaseMetadata** ()
- void **prepareAddImage** (int albumId)
- bool **commitAddImage** ()

### Operations on File Metadata

- void [fileModified](#) ()
- static void [fillCommonContainer](#) (qulonglong imageid, [ImageCommonContainer](#) \*const container)
- static QDateTime [creationDateFromFilesystem](#) (const QFileInfo &info)
- void **prepareUpdateImage** ()
- void **commitUpdateImage** ()
- bool **scanFromIdenticalFile** ()
- void **scanFile** (ScanMode mode)
- void **scanItemInformation** ()
- void **commitItemInformation** ()

### Operations on Photo Metadata

- static QString [iptcCorePropertyName](#) (MetadataInfo::Field field)
- static MetadataFields **allImageMetadataFields** ()
- QString **detectImageFormat** () const
- void **scanImageMetadata** ()
- void **commitImageMetadata** ()
- void **scanItemPosition** ()
- void **commitItemPosition** ()
- void **scanItemComments** ()
- void **commitItemComments** ()
- void **scanItemCopyright** ()
- void **commitItemCopyright** ()
- void **scanIPTCCore** ()
- void **commitIPTCCore** ()
- void **scanTags** ()
- void **commitTags** ()
- void **scanFaces** ()
- void **commitFaces** ()
- bool **checkRatingFromMetadata** (const QVariant &ratingFromMetadata) const
- void **checkCreationDateFromMetadata** (QVariant &dateFromMetadata) const

### Operations on Video Metadata

- static void [fillVideoMetadataContainer](#) (qulonglong imageid, [VideoMetadataContainer](#) \*const container)
- void **scanVideoInformation** ()
- void **scanVideoMetadata** ()
- void **commitVideoMetadata** ()
- QString **detectVideoFormat** () const
- QString **detectAudioFormat** () const
- static MetadataFields **allVideoMetadataFields** ()

## 6.958.1 Constructor & Destructor Documentation

### 6.958.1.1 ItemScanner() [1/3]

```
Digikam::ItemScanner::ItemScanner (
    const QFileInfo & info,
    const ItemScanInfo & Iteminfo )
```

Construct an [ItemScanner](#) object from an existing QFileInfo and [ItemScanInfo](#) object. This constructor shall be used with [fileModified\(\)](#) or [fullScan\(\)](#).

### 6.958.1.2 ItemScanner() [2/3]

```
Digikam::ItemScanner::ItemScanner (
    const QFileInfo & info ) [explicit]
```

Construct an [ItemScanner](#) from an existing QFileInfo object. Use this constructor if you intend to call [newFile\(\)](#).

### 6.958.1.3 ItemScanner() [3/3]

```
Digikam::ItemScanner::ItemScanner (
    qlonglong imageid ) [explicit]
```

Construct an [ItemScanner](#) for an image in the database. File info, Scan info and the category will be retrieved from the database.

## 6.958.2 Member Function Documentation

### 6.958.2.1 cleanScan()

```
void Digikam::ItemScanner::cleanScan ( )
```

This is the same as [rescan\(\)](#) but the database metadata will be cleaned up if the corresponding metadata write option is enabled.

### 6.958.2.2 commit()

```
void Digikam::ItemScanner::commit ( )
```

Commits the scanned information to the database. You must call this after scanning was done for any changes to take effect. Only this method will perform write operations to the database.

### 6.958.2.3 copiedFrom()

```
void Digikam::ItemScanner::copiedFrom (
    int albumId,
    qlonglong srcId )
```

Similar to [newFile](#). Call this when you want [ItemScanner](#) to add a new file to the database which is a copy of another file, copying attributes from the src and rescanning other attributes as appropriate. Give the id of the album of the new file, and the id of the src file.

### 6.958.2.4 creationDateFromFilesystem()

```
QDateTime Digikam::ItemScanner::creationDateFromFilesystem (
    const QFileInfo & info ) [static]
```

Returns a suitable creation date from file system information. Use this as a fallback if metadata is not available.

### 6.958.2.5 fileModified()

```
void Digikam::ItemScanner::fileModified ( )
```

Call this when you have detected that a file in the database has been modified on disk. Only two groups of fields will be updated in the database:

- filesystem specific properties (those that signaled you that the file has been modified because their state on disk differed from the state in the database)
- image specific properties, for which a difference in the database independent from the actual file does not make sense (width/height, bit depth, color model)

### 6.958.2.6 fillCommonContainer()

```
void Digikam::ItemScanner::fillCommonContainer (
    qlonglong imageid,
    ImageCommonContainer *const container ) [static]
```

Returns File-metadata container with user-presentable information. These methods provide the reverse service: Not writing into the db, but reading from the db.

### 6.958.2.7 fillVideoMetadataContainer()

```
void Digikam::ItemScanner::fillVideoMetadataContainer (
    qlonglong imageid,
    VideoMetadataContainer *const container ) [static]
```

Returns Video container with user-presentable information. These methods provide the reverse service: Not writing into the db, but reading from the db.

### 6.958.2.8 formatToString()

```
QString Digikam::ItemScanner::formatToString (
    const QString & format ) [static]
```

Helper method to translate enum values to user presentable strings

### 6.958.2.9 hasHistoryToResolve()

```
bool Digikam::ItemScanner::hasHistoryToResolve ( ) const
```

Returns true if this file has been marked as needing history resolution at a later stage

### 6.958.2.10 id()

```
qlonglong Digikam::ItemScanner::id ( ) const
```

Returns the image id of the scanned file, if (yet) available.

### 6.958.2.11 iptcCorePropertyName()

```
QString Digikam::ItemScanner::iptcCorePropertyName (
    MetadataInfo::Field field ) [static]
```

Helper method to return official property name by which IPTC core properties are stored in the database ([ItemCopyright](#) and [ImageProperties](#) table). Allowed arguments: All `MetadataInfo::Fields` starting with "IptcCore..."

### 6.958.2.12 itemScanInfo()

```
const ItemScanInfo & Digikam::ItemScanner::itemScanInfo ( ) const
```

Provides access to the information retrieved by scanning. The validity depends on the previously executed scan.

### 6.958.2.13 loadFromDisk()

```
void Digikam::ItemScanner::loadFromDisk ( )
```

Loads data from disk (metadata, image file properties). This method is called from any of the main entry points above. You can call it before if you want to control the time when it is executed. Calling it a second time with data already loaded will do nothing.

### 6.958.2.14 newFile()

```
void Digikam::ItemScanner::newFile (
    int albumId )
```

Call this when you want [ItemScanner](#) to add a new file to the database and read all information into the database.

### 6.958.2.15 newFileFullScan()

```
void Digikam::ItemScanner::newFileFullScan (
    int albumId )
```

Call this when you want [ItemScanner](#) to add a new file to the database and read all information into the database. This variant will not use the unique hash to establish identify with an existing entry, but read all information newly from the file.

### 6.958.2.16 rescan()

```
void Digikam::ItemScanner::rescan ( )
```

Call this to take an existing image in the database, but re-read all information from the file into the database, possibly overwriting information there.



### 6.958.2.17 resolvedImageHistory()

```
DImageHistory Digikam::ItemScanner::resolvedImageHistory (
    const DImageHistory & history,
    bool mustBeAvailable = false ) [static]
```

All referred images of the given history will be resolved. In the returned history, the actions are the same, while each referred image actually exists in the collection (if `mustBeAvailable` is true, it is even in a currently available collection). That means the number of referred images may be less or greater than initially. Note that this history may have peculiar properties, like multiple Original or Current entries (if the source entry resolves to multiple collection images), so this history is only for internal use, not for storage.

### 6.958.2.18 resolveHistoryImageId()

```
QList< qlonglong > Digikam::ItemScanner::resolveHistoryImageId (
    const HistoryImageId & historyId ) [static]
```

Returns all image ids fulfilling the given image id.

### 6.958.2.19 resolveImageHistory()

```
bool Digikam::ItemScanner::resolveImageHistory (
    qlonglong id,
    QList< qlonglong > * needTaggingIds = nullptr ) [static]
```

Resolves the image history of the image id by filling the ImageRelations table for all contained referred images. If `needTaggingIds` is given, all ids marked for needing tagging of the history graph are added.

### 6.958.2.20 sameReferredImage()

```
bool Digikam::ItemScanner::sameReferredImage (
    const HistoryImageId & id1,
    const HistoryImageId & id2 ) [static]
```

Determines if the two ids refer to the same image. Does not check if such a referred image really exists.

### 6.958.2.21 setCategory()

```
void Digikam::ItemScanner::setCategory (
    DatabaseItem::Category category )
```

Inform the scanner about the category of the file. Required at least for `newFile()` calls, recommended for calls with the first constructor above as well.

### 6.958.2.22 sortByProximity()

```
void Digikam::ItemScanner::sortByProximity (
    QList< ItemInfo > & infos,
    const ItemInfo & subject ) [static]
```

Sort a list of infos by proximity to the given subject. Infos are near if they are e.g. in the same album. They are not near if they are e.g. in different collections.

### 6.958.2.23 tagItemHistoryGraph()

```
void Digikam::ItemScanner::tagItemHistoryGraph (
    qulonglong id ) [static]
```

Takes the history graph reachable from the given image, and assigns versioning tags to all entries based on history image types and graph structure

## 6.959 Digikam::ItemScanner::Private Class Reference

### Public Attributes

- [ItemScannerCommit](#) **commit**
- [QFileInfo](#) **fileInfo**
- bool **hasHistoryToResolve** = false
- bool **hasImage** = false
- bool **hasMetadata** = false
- [DImg](#) **img**
- bool **loadedFromDisk** = false
- [DMetadata](#) \* **metadata** = nullptr
- [ItemScanInfo](#) **scanInfo**
- [ItemScanner::ScanMode](#) **scanMode** = ModifiedScan
- [QElapsedTimer](#) **timer**

## 6.960 Digikam::ItemScannerCommit Class Reference

### Public Types

- enum **Operation** { **NoOp** , **AddItem** , **UpdateItem** }

### Public Attributes

- [CaptionsMap](#) **captions**
- bool **commitFaces** = false
- bool **commitImageMetadata** = false
- bool **commitIPTCCore** = false
- bool **commitItemComments** = false
- bool **commitItemCopyright** = false
- bool **commitItemInformation** = false
- bool **commitItemPosition** = false
- bool **commitVideoMetadata** = false
- qulonglong **copyImageAttributesId** = -1
- [Template](#) **copyrightTemplate**
- bool **hasColorTag** = false
- bool **hasPickTag** = false
- [QString](#) **headline**
- [QString](#) **historyXml**
- [DatabaseFields::ItemInformation](#) **imageInformationFields**
- [QVariantList](#) **imageInformationInfos**
- [QVariantList](#) **imageMetadataInfos**
- [QVariantList](#) **imagePositionInfos**
- [QVariantList](#) **iptcCoreMetadataInfos**
- [QMultiMap](#)< [QString](#), [QVariant](#) > **metadataFacesMap**
- [Operation](#) **operation** = NoOp
- [QList](#)< int > **tagIds**
- [CaptionsMap](#) **titles**
- [QString](#) **uuid**

## 6.961 Digikam::ItemSelectionOverlay Class Reference

Inheritance diagram for Digikam::ItemSelectionOverlay:



### Public Member Functions

- **ItemSelectionOverlay** (QObject \*const parent)
- void [setActive](#) (bool active) override

## Public Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- [ItemViewHoverButton](#) \* **button** () const
- [HoverButtonDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotClicked** (bool checked)
- void **slotSelectionChanged** (const QItemSelection &, const QItemSelection &)

## Protected Slots inherited from [Digikam::HoverButtonDelegateOverlay](#)

- void **slotEntered** (const QModelIndex &index) override
- void **slotReset** () override

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotEntered** (const QModelIndex &index)
- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

## Protected Member Functions

- [ItemViewHoverButton](#) \* **createButton** () override
- void **updateButton** (const QModelIndex &index) override

### Protected Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- QWidget \* [createWidget](#) () override
- void [visualChange](#) () override

### Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual bool [checkIndex](#) (const QModelIndex &index) const
- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual void [hide](#) ()
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- virtual void [widgetEnterEvent](#) ()
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- virtual void [widgetLeaveEvent](#) ()
- void [widgetLeaveNotifyMultiple](#) ()

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

### Additional Inherited Members

### Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void [hideNotification](#) ()
- void [requestNotification](#) (const QModelIndex &index, const QString &message)
- void [update](#) (const QModelIndex &index)

### Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.961.1 Member Function Documentation

### 6.961.1.1 createButton()

```
ItemViewHoverButton * Digikam::ItemSelectionOverlay::createButton ( ) [override], [protected], [virtual]
```

Create your widget here. Pass view() as parent.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.961.1.2 setActive()

```
void Digikam::ItemSelectionOverlay::setActive (
    bool active ) [override], [virtual]
```

Will call [createButton\(\)](#).

Reimplemented from [Digikam::HoverButtonDelegateOverlay](#).

### 6.961.1.3 updateButton()

```
void Digikam::ItemSelectionOverlay::updateButton (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implements [Digikam::HoverButtonDelegateOverlay](#).

## 6.962 Digikam::ItemSelectionOverlayButton Class Reference

Inheritance diagram for Digikam::ItemSelectionOverlayButton:



### Public Member Functions

- **ItemSelectionOverlayButton** (QAbstractItemView \*const parentView)
- QSize [sizeHint](#) () const override

## Public Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- `QModelIndex` **index** () const
- void **initIcon** ()
- **ItemViewHoverButton** (`QAbstractItemView *const` parentView)
- void **reset** ()
- void **setIndex** (const `QModelIndex &`index)
- void **setVisible** (bool visible) override

## Protected Member Functions

- `QIcon` **icon** () override
- void **updateToolTip** () override

## Protected Member Functions inherited from [Digikam::ItemViewHoverButton](#)

- void **enterEvent** (`QEnterEvent *event`)
- void **leaveEvent** (`QEvent *event`)
- void **paintEvent** (`QPaintEvent *event`)
- void **setup** ()

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ItemViewHoverButton](#)

- void **refreshIcon** ()
- void **setFadingValue** (int value)
- void **startFading** ()
- void **stopFading** ()

## Protected Attributes inherited from [Digikam::ItemViewHoverButton](#)

- `QTimeLine * m_fadingTimeLine` = nullptr
- int **m\_fadingValue** = 0
- `QIcon` **m\_icon**
- `QPersistentModelIndex` **m\_index**
- bool **m\_isHovered** = false

## 6.962.1 Member Function Documentation

### 6.962.1.1 icon()

```
QIcon Digikam::ItemSelectionOverlayButton::icon ( ) [override], [protected], [virtual]
```

Return your icon here. Will be queried again on toggle.

Implements [Digikam::ItemViewHoverButton](#).



### 6.962.1.2 sizeHint()

```
QSize Digikam::ItemSelectionOverlayButton::sizeHint ( ) const [override], [virtual]
```

Reimplement to match the size of your icon

Implements [Digikam::ItemViewHoverButton](#).

### 6.962.1.3 updateToolTip()

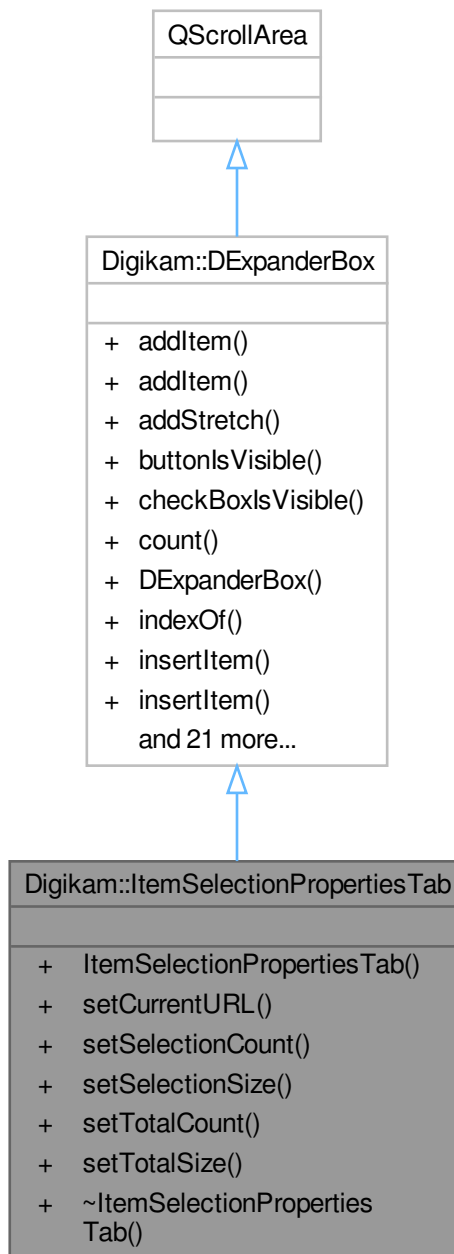
```
void Digikam::ItemSelectionOverlayButton::updateToolTip ( ) [override], [protected], [virtual]
```

Optionally update tooltip here. Will be called again on state change.

Reimplemented from [Digikam::ItemViewHoverButton](#).

## 6.963 Digikam::ItemSelectionPropertiesTab Class Reference

Inheritance diagram for Digikam::ItemSelectionPropertiesTab:



### Public Member Functions

- **ItemSelectionPropertiesTab** (QWidget \*const parent)
- void **setCurrentURL** (const QUrl &url=QUrl())

- void **setSelectionCount** (const QString &str)
- void **setSelectionSize** (const QString &str)
- void **setTotalCount** (const QString &str)
- void **setTotalSize** (const QString &str)

## Public Member Functions inherited from [Digikam::DExpanderBox](#)

- void [addItem](#) (QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **addItem** (QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **addStretch** ()
- bool **buttonIsVisible** (int index) const
- bool **checkboxIsVisible** (int index) const
- int **count** () const
- **DExpanderBox** (QWidget \*const parent=nullptr)
- int **indexOf** ([DLabelExpander](#) \*const widget) const
- void [insertItem](#) (int index, QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertItem** (int index, QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertStretch** (int index)
- bool **isChecked** (int index) const
- bool **isItemEnabled** (int index) const
- bool **isItemExpanded** (int index) const
- QIcon **itemIcon** (int index) const
- QString **itemText** (int index) const
- QString **itemToolTip** (int index) const
- virtual void **readSettings** (KConfigGroup &group)
- void **removeItem** (int index)
- void **setButtonIcon** (int index, const QIcon &icon)
- void **setButtonVisible** (int index, bool b)
- void **setCheckBoxVisible** (int index, bool b)
- void **setChecked** (int index, bool b)
- void **setItemEnabled** (int index, bool enabled)
- void **setItemExpanded** (int index, bool b)
- void **setItemIcon** (int index, const QIcon &icon)
- void **setItemText** (int index, const QString &txt)
- void **setItemToolTip** (int index, const QString &tip)
- [DLabelExpander](#) \* **widget** (int index) const
- virtual void **writeSettings** (KConfigGroup &group)

## Additional Inherited Members

## Signals inherited from [Digikam::DExpanderBox](#)

- void **signalItemButtonPressed** (int index)
- void **signalItemExpanded** (int index, bool b)
- void **signalItemToggled** (int index, bool b)

## 6.964 Digikam::ItemShortInfo Class Reference

### Public Member Functions

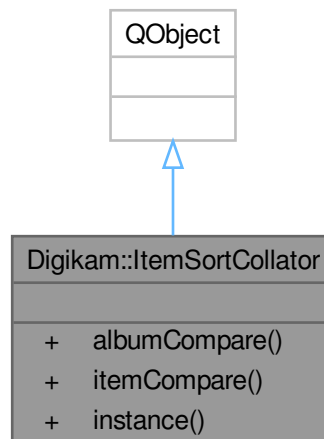
- bool **isNull** () const

### Public Attributes

- QString **album**
- int **albumID** = 0
- int **albumRootID** = 0
- qlonglong **id** = 0
- QString **itemName**

## 6.965 Digikam::ItemSortCollator Class Reference

Inheritance diagram for Digikam::ItemSortCollator:



### Public Member Functions

- int **albumCompare** (const QString &a, const QString &b, Qt::CaseSensitivity caseSensitive, bool natural) const
- int **itemCompare** (const QString &a, const QString &b, Qt::CaseSensitivity caseSensitive, bool natural) const

### Static Public Member Functions

- static `ItemSortCollator` \* **instance** ()

## Friends

- class `ItemSortCollatorCreator`

## 6.965.1 Member Function Documentation

### 6.965.1.1 instance()

```
ItemSortCollator * Digikam::ItemSortCollator::instance ( ) [static]
```

Global instance of internal item sort collator. All accessor methods are thread-safe.

## 6.966 Digikam::ItemSortSettings Class Reference

### Public Types

- enum `CategorizationMode` { `NoCategories` , `OneCategory` , `CategoryByAlbum` , `CategoryByFormat` , `CategoryByMonth` , `CategoryByFaces` }
- enum `SortOrder` { `AscendingOrder` = Qt::AscendingOrder , `DescendingOrder` = Qt::DescendingOrder , `DefaultOrder` }
- enum `SortRole` { `SortByFileName` , `SortByFilePath` , `SortByCreationDate` , `SortByModificationDate` , `SortByFileSize` , `SortByRating` , `SortByImageSize` , `SortByAspectRatio` , `SortByFaces` , `SortBySimilarity` , `SortByManualOrderAndName` , `SortByManualOrderAndDate` }

### Public Member Functions

- int `compare` (const `ItemInfo` &left, const `ItemInfo` &right) const
- int `compare` (const `ItemInfo` &left, const `ItemInfo` &right, `SortRole` sortRole) const  
— *Image Sorting* —
- int `compareCategories` (const `ItemInfo` &left, const `ItemInfo` &right, const `FaceTagsIface` &leftFace, const `FaceTagsIface` &rightFace) const
- bool `isCategorized` () const
- bool `lessThan` (const `ItemInfo` &left, const `ItemInfo` &right) const
- bool `lessThan` (const `QVariant` &left, const `QVariant` &right) const
- bool `operator==` (const `ItemSortSettings` &other) const
- void `setCategorizationMode` (`CategorizationMode` mode)  
— *Categories* —
- void `setCategorizationSortOrder` (`SortOrder` order)
- void `setSortOrder` (`SortOrder` order)
- void `setSortRole` (`SortRole` role)
- void `setStringTypeNatural` (bool natural)
- `DatabaseFields::Set` `watchFlags` () const  
— *Change notification* —

## Static Public Member Functions

- `template<typename T >`  
static int **compareByOrder** (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int [compareByOrder](#) (int compareResult, Qt::SortOrder sortOrder)
- `template<typename T >`  
static int **compareValue** (const T &a, const T &b)
- static Qt::SortOrder **defaultSortOrderForCategorizationMode** ([CategorizationMode](#) mode)
- static Qt::SortOrder **defaultSortOrderForSortRole** ([SortRole](#) role)
- `template<typename T >`  
static bool **lessThanByOrder** (const T &a, const T &b, Qt::SortOrder sortOrder)
- *Utilities* —
- static int **naturalCompare** (const QString &a, const QString &b, Qt::SortOrder sortOrder, Qt::CaseSensitivity caseSensitive=Qt::CaseSensitive, bool natural=true)

## Public Attributes

- Qt::CaseSensitivity **categorizationCaseSensitivity** = Qt::CaseSensitive
- [CategorizationMode](#) **categorizationMode** = [NoCategories](#)
- [SortOrder](#) **categorizationSortOrder** = [DefaultOrder](#)
- Qt::SortOrder **currentCategorizationSortOrder** = Qt::AscendingOrder  
*Only Ascending or Descending, never DefaultOrder.*
- Qt::SortOrder **currentSortOrder** = Qt::AscendingOrder
- Qt::CaseSensitivity **sortCaseSensitivity** = Qt::CaseSensitive
- [SortOrder](#) **sortOrder** = [DefaultOrder](#)
- [SortRole](#) **sortRole** = [SortByFileName](#)
- bool **strTypeNatural** = true

## 6.966.1 Member Enumeration Documentation

### 6.966.1.1 CategorizationMode

```
enum Digikam::ItemSortSettings::CategorizationMode
```

#### Enumerator

NoCategories	categorization switched off
OneCategory	all items in one global category

### 6.966.1.2 SortOrder

```
enum Digikam::ItemSortSettings::SortOrder
```

#### Enumerator

DefaultOrder	sort order depends on the chosen sort role
--------------	--

### 6.966.1.3 SortRole

```
enum Digikam::ItemSortSettings::SortRole
```

#### Enumerator

SortByImageSize	pixel number
SortByAspectRatio	width / height * 100000
SortByFaces	count of unconfirmed faces

## 6.966.2 Member Function Documentation

### 6.966.2.1 compare()

```
int Digikam::ItemSortSettings::compare (
    const ItemInfo & left,
    const ItemInfo & right ) const
```

Compares the ItemInfos left and right. Return -1 if left is less than right, 1 if left is greater than right, and 0 if left equals right comparing the current sort role's value. Adheres to set sort role and sort order.

### 6.966.2.2 compareByOrder()

```
static int Digikam::ItemSortSettings::compareByOrder (
    int compareResult,
    Qt::SortOrder sortOrder ) [inline], [static]
```

Takes a typical result from a compare method (0 is equal, -1 is less than, 1 is greater than) and applies the given sort order to it.

### 6.966.2.3 compareCategories()

```
int Digikam::ItemSortSettings::compareCategories (
    const ItemInfo & left,
    const ItemInfo & right,
    const FaceTagsIface & leftFace,
    const FaceTagsIface & rightFace ) const
```

Compares the categories of left and right. Return -1 if left is less than right, 0 if both fall in the same category, and 1 if left is greater than right. Adheres to set categorization mode and current category sort order. Face passed in to allow Categorization by Faces. Pass in an empty Face if not needed.

### 6.966.2.4 compareValue()

```
template<typename T >
static int Digikam::ItemSortSettings::compareValue (
    const T & a,
    const T & b ) [inline], [static]
```

Returns the usual compare result of -1, 0, or 1 for lessThan, equals and greaterThan.

### 6.966.2.5 lessThan() [1/2]

```
bool Digikam::ItemSortSettings::lessThan (
    const ItemInfo & left,
    const ItemInfo & right ) const
```

Returns true if left is less than right. Adheres to current sort role and sort order.

### 6.966.2.6 lessThan() [2/2]

```
bool Digikam::ItemSortSettings::lessThan (
    const QVariant & left,
    const QVariant & right ) const
```

Returns true if left QVariant is less than right. Adheres to current sort role and sort order. Use for extraValue, if necessary.

### 6.966.2.7 lessThanByOrder()

```
template<typename T >
static bool Digikam::ItemSortSettings::lessThanByOrder (
    const T & a,
    const T & b,
    Qt::SortOrder sortOrder ) [inline], [static]
```

Returns  $a < b$  if sortOrder is Ascending, or  $b < a$  if order is descending.

### 6.966.2.8 naturalCompare()

```
static int Digikam::ItemSortSettings::naturalCompare (
    const QString & a,
    const QString & b,
    Qt::SortOrder sortOrder,
    Qt::CaseSensitivity caseSensitive = Qt::CaseSensitive,
    bool natural = true ) [inline], [static]
```

Compares the two string by natural comparison and adheres to given sort order

### 6.966.2.9 watchFlags()

```
DatabaseFields::Set Digikam::ItemSortSettings::watchFlags ( ) const
```

Returns database fields a change in which would affect the current sorting.



## 6.967 Digikam::ItemTagPair Class Reference

### Public Member Functions

- void [addProperty](#) (const QString &key, const QString &value)
- QStringList [allValues](#) (const QStringList &keys) const  
*Returns value() concatenated for all given keys.*
- void [assignTag](#) ()
- void [clearProperties](#) ()  
*Removes all properties.*
- bool [hasAnyProperty](#) (const QStringList &keys) const  
*Returns true if any of the properties is set.*
- bool [hasProperty](#) (const QString &key) const  
*Returns true if the property is set.*
- bool [hasValue](#) (const QString &key, const QString &value) const  
*Returns true of the given property and value is set.*
- qlonglong [imageld](#) () const
- bool [isAssigned](#) () const
- bool [isNull](#) () const
- [ItemTagPair](#) ()
- [ItemTagPair](#) (const [ItemInfo](#) &info, int tagId)
- [ItemTagPair](#) (const [ItemTagPair](#) &other)
- [ItemTagPair](#) (qlonglong imageld, int tagId)
- [ItemTagPair](#) & [operator=](#) (const [ItemTagPair](#) &other)
- QMap< QString, QString > [properties](#) () const  
*Returns a map of all key->value pairs.*
- QStringList [propertyKeys](#) () const  
*Returns all set property keys.*
- void [removeProperties](#) (const QString &key)  
*Remove all occurrences of the property.*
- void [removeProperty](#) (const QString &key, const QString &value)  
*Remove all occurrences of the property.*
- void [setProperty](#) (const QString &key, const QString &value)  
*Set the given property. Replaces all previous occurrences of this property.*
- int [tagId](#) () const
- void [unAssignTag](#) ()
- QString [value](#) (const QString &key) const  
*Returns the value of the given property, or a null string if not set.*
- QStringList [values](#) (const QString &key) const  
*Returns a list of values with the given property.*

### Static Public Member Functions

- static QList< [ItemTagPair](#) > [availablePairs](#) (const [ItemInfo](#) &info)
- static QList< [ItemTagPair](#) > [availablePairs](#) (qlonglong imageld)

### 6.967.1 Constructor & Destructor Documentation

#### 6.967.1.1 ItemTagPair() [1/2]

```
Digikam::ItemTagPair::ItemTagPair ( )
```

This class provides a wrapper over the Database methods to access the properties of tag / image association. It is meant to be a short-lived object, it does not listen to external database changes. Creates a null pair.

### 6.967.1.2 ItemTagPair() [2/2]

```
Digikam::ItemTagPair::ItemTagPair (
    qlonglong imageId,
    int tagId )
```

Access the properties of the given image - tag pair

## 6.967.2 Member Function Documentation

### 6.967.2.1 addProperty()

```
void Digikam::ItemTagPair::addProperty (
    const QString & key,
    const QString & value )
```

Adds the given property. Does not change any previous occurrences of this property, allowing multiple properties with the same key. (duplicates of same key *and* value are not added, though)

### 6.967.2.2 assignTag()

```
void Digikam::ItemTagPair::assignTag ( )
```

Assigns the tag to the image

### 6.967.2.3 availablePairs()

```
QList< ItemTagPair > Digikam::ItemTagPair::availablePairs (
    qlonglong imageId ) [static]
```

Return all pairs for the given image for which entries exist. This list of tags may not be identical to the tags assigned to the image.

### 6.967.2.4 isAssigned()

```
bool Digikam::ItemTagPair::isAssigned ( ) const
```

Returns if the tag is assigned to the image

### 6.967.2.5 unAssignTag()

```
void Digikam::ItemTagPair::unAssignTag ( )
```

Removes the tag from the image

## 6.968 Digikam::ItemThumbnailBar Class Reference

Inheritance diagram for Digikam::ItemThumbnailBar:



### Public Slots

- void **assignRating** (const QList< QModelIndex > &index, int rating)
- void **slotDockLocationChanged** (Qt::DockWidgetArea area)

## Public Slots inherited from [Digikam::ItemCategorizedView](#)

- void [hintAt](#) (const [ItemInfo](#) &info)
- void [openAlbum](#) (const QList< [Album](#) \* > &album)
- void [setCurrentInfo](#) (const [ItemInfo](#) &info)
- void [setCurrentUrl](#) (const QUrl &url)
- void [setCurrentUrlWhenAvailable](#) (const QUrl &url)
- void [setCurrentWhenAvailable](#) (qulonglong imageld)
- void [setSelectedItemInfos](#) (const QList< [ItemInfo](#) > &infos)
- void [setSelectedUrls](#) (const QList< QUrl > &urlList)
- void [setThumbnailSize](#) (int size)

## Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void [copy](#) () override
- void [cut](#) () override
- void [hideIndexNotification](#) ()
- void [paste](#) () override
- void [showIndexNotification](#) (const QModelIndex &index, const QString &message)

## Public Slots inherited from [Digikam::DCategorizedView](#)

- void [reset](#) () override

## Public Member Functions

- QModelIndex [firstIndex](#) () const
- void [installOverlays](#) ()
- [ItemThumbnailBar](#) (QWidget \*const parent=nullptr)
- QModelIndex [lastIndex](#) () const
- QModelIndex [nextIndex](#) (const QModelIndex &index) const
- QModelIndex [previousIndex](#) (const QModelIndex &index) const
- void [setFlow](#) (QListView::Flow newFlow)
- void [setModelsFiltered](#) ([ItemModel](#) \*model, [ImageSortFilterModel](#) \*filterModel)
- void [setScrollBarPolicy](#) (Qt::ScrollBarPolicy policy)

## Public Member Functions inherited from [Digikam::ItemCategorizedView](#)

- void [addOverlay](#) ([ItemDelegateOverlay](#) \*overlay, [ItemDelegate](#) \*delegate=nullptr)  
*Add and remove an overlay. It will as well be removed automatically when destroyed. Unless you pass a different delegate, the current delegate will be used.*
- void [addSelectionOverlay](#) ([ItemDelegate](#) \*delegate=nullptr)
- [Album](#) \* [albumAt](#) (const QPoint &pos) const
- [ItemInfoList](#) [allItemInfos](#) () const
- QList< QUrl > [allUrls](#) () const
- [Album](#) \* [currentAlbum](#) () const
- [ItemInfo](#) [currentInfo](#) () const
- QUrl [currentUrl](#) () const
- [ItemDelegate](#) \* [delegate](#) () const
- QItemSelectionModel \* [getSelectionModel](#) () const
- [ItemAlbumFilterModel](#) \* [imageAlbumFilterModel](#) () const

- [ItemAlbumModel](#) \* **imageAlbumModel** () const  
*Returns 0 if the [ItemModel](#) is not an [ItemAlbumModel](#).*
- [ItemFilterModel](#) \* **imageFilterModel** () const  
*Returns any [ItemFilterMode](#) in chain. May not be [sourceModel\(\)](#)*
- [ItemModel](#) \* **imageModel** () const
- [ImageSortFilterModel](#) \* **imageSortFilterModel** () const
- [ItemThumbnailModel](#) \* **imageThumbnailModel** () const  
*Returns 0 if the [ItemModel](#) is not an [ItemThumbnailModel](#).*
- [QModelIndex](#) **indexForInfo** (const [ItemInfo](#) &info) const
- **ItemCategorizedView** (QWidget \*const parent=nullptr)
- [ItemInfo](#) **nextInfo** (const [ItemInfo](#) &info)
- [ItemInfo](#) **nextInOrder** (const [ItemInfo](#) &startingPoint, int nth)
- [ItemInfo](#) **previousInfo** (const [ItemInfo](#) &info)
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemInfoList](#) **selectedItemInfos** () const
- [ItemInfoList](#) **selectedItemInfosCurrentFirst** () const
- void **setModels** ([ItemModel](#) \*model, [ImageSortFilterModel](#) \*filterModel)
- virtual void **setThumbnailSize** (const [ThumbnailSize](#) &size)
- [ThumbnailSize](#) **thumbnailSize** () const
- void **toIndex** (const [QUrl](#) &url)

### Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **awayFromSelection** ()
- [DItemDelegate](#) \* **delegate** () const
- void **invertSelection** ()
- bool **isToolTipEnabled** () const
- **ItemViewCategorized** (QWidget \*const parent=nullptr)
- int **numberOfSelectedIndexes** () const
- void **scrollTo** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible) override
- void **scrollToRelaxed** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible)
- void **setInitialSelectedItem** (bool enabled)
- void **setScrollCurrentToCenter** (bool enabled)
- void **setScrollStepGranularity** (int factor)
- void **setSelectedIndexes** (const [QList](#)< [QModelIndex](#) > &indexes)
- void **setSpacing** (int spacing)
- void **setToolTipEnabled** (bool enabled)
- void **setUsePointingHandCursor** (bool useCursor)
- void **toFirstIndex** ()
- void **toIndex** (const [QModelIndex](#) &index)
- void **toLastIndex** ()
- void **toNextIndex** ()
- void **toPreviousIndex** ()

### Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual [QModelIndexList](#) **categorizedIndexesIn** (const [QRect](#) &rect) const
- virtual [QModelIndex](#) **categoryAt** (const [QPoint](#) &point) const
- [DCategoryDrawer](#) \* **categoryDrawer** () const
- virtual [QItemSelectionRange](#) **categoryRange** (const [QModelIndex](#) &index) const
- virtual [QRect](#) **categoryVisualRect** (const [QModelIndex](#) &index) const
- **DCategorizedView** (QWidget \*const parent=nullptr)
- [QModelIndex](#) **indexAt** (const [QPoint](#) &point) const override
- void **setCategoryDrawer** ([DCategoryDrawer](#) \*categoryDrawer)
- void **setDrawDraggedItems** (bool drawDraggedItems)
- void **setGridSize** (const [QSize](#) &size)
- void **setModel** ([QAbstractItemModel](#) \*model) override
- [QRect](#) **visualRect** (const [QModelIndex](#) &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

## Public Member Functions inherited from [Digikam::GroupingViewImplementation](#)

- [ItemInfoList](#) **getHiddenGroupedInfos** (const [ItemInfoList](#) &infos) const
- bool **needGroupResolving** ([OperationType](#) type, const [ItemInfoList](#) &infos) const
- [ItemInfoList](#) **resolveGrouping** (const [ItemInfoList](#) &infos) const

## Protected Member Functions

- bool **event** (QEvent \*) override
- bool **hasHiddenGroupedImages** (const [ItemInfo](#) &info) const override  
*must be implemented by parent view*
- void **slotSetupChanged** () override

## Protected Member Functions inherited from [Digikam::ItemCategorizedView](#)

- virtual void **activated** (const [ItemInfo](#) &info, Qt::KeyboardModifiers modifiers)  
*Reimplement these in a subclass.*
- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfoList](#) **imageInfos** (const QList< QModelIndex > &indexes) const
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- void **installDefaultModels** ()  
*install default [ItemAlbumModel](#) and filter model, ready for use*
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([ItemDelegate](#) \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override  
*Reimplement these in a subclass.*
- virtual void **showContextMenuOnInfo** (QContextMenuEvent \*event, const [ItemInfo](#) &info)
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** (DItemDelegate \*delegate)
- void **setToolTip** (ItemViewToolTip \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

## Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

## Additional Inherited Members

### Signals inherited from [Digikam::ItemCategorizedView](#)

- void **currentChanged** (const [ItemInfo](#) &info)
- void **deselected** (const QList< [ItemInfo](#) > &nowDeselectedInfos)
 

*Emitted when items are deselected. There may be other selected infos left. This signal is not emitted when the model is reset; then only selectionCleared is emitted.*
- void **imageActivated** (const [ItemInfo](#) &info)
 

*Emitted when the given image is activated. Info is never null.*
- void **modelChanged** ()
 

*Emitted when a new model is set.*
- void **selected** (const QList< [ItemInfo](#) > &newSelectedInfos)
 

*Emitted when new items are selected. The parameter includes only the newly selected infos, there may be other already selected infos.*

### Signals inherited from [Digikam::ItemViewCategorized](#)

- void [clicked](#) (const QMouseEvent \*e, const QModelIndex &index)
- void **entered** (const QMouseEvent \*e, const QModelIndex &index)
- void [keyPressed](#) (QKeyEvent \*e)
- void [selectionChanged](#) ()
- void [selectionCleared](#) ()
- void [viewportClicked](#) (const QMouseEvent \*e)
- void **zoomInStep** ()
- void **zoomOutStep** ()

### Protected Slots inherited from [Digikam::ItemCategorizedView](#)

- void **slotCurrentUrlTimer** ()
- void **slotItemInfosAdded** ()

### Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void **layoutAboutToBeChanged** ()
- void **layoutWasChanged** ()
- void **slotActivated** (const QModelIndex &index)
- void **slotClicked** (const QModelIndex &index)
- void **slotEntered** (const QModelIndex &index)
- virtual void **slotThemeChanged** ()

### Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override



## 6.968.1 Member Function Documentation

### 6.968.1.1 hasHiddenGroupedImages()

```
bool Digikam::ItemThumbnailBar::hasHiddenGroupedImages (
    const ItemInfo & ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::GroupingViewImplementation](#).

### 6.968.1.2 setModelsFiltered()

```
void Digikam::ItemThumbnailBar::setModelsFiltered (
    ItemModel * model,
    ImageSortFilterModel * filterModel )
```

This installs a duplicate filter model, if the [ItemModel](#) may contain duplicates. Otherwise, just use `setModels()`.

### 6.968.1.3 setScrollBarPolicy()

```
void Digikam::ItemThumbnailBar::setScrollBarPolicy (
    Qt::ScrollBarPolicy policy )
```

Sets the policy always for the one scroll bar which is relevant, depending on orientation.

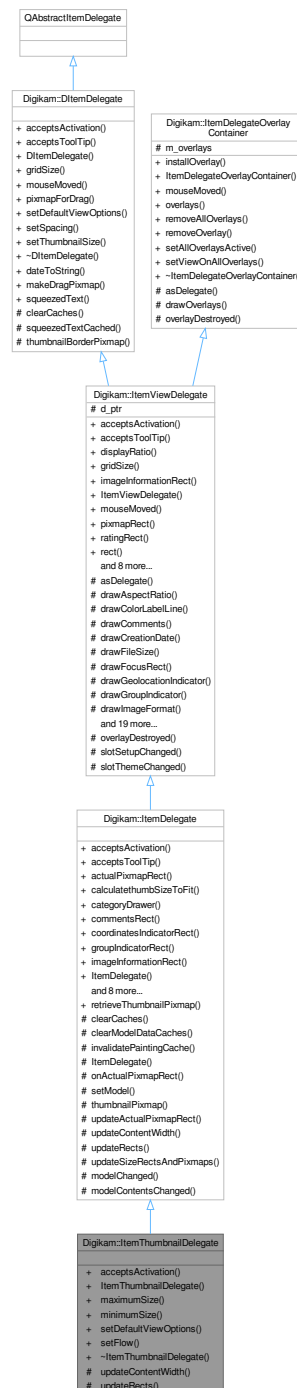
### 6.968.1.4 slotSetupChanged()

```
void Digikam::ItemThumbnailBar::slotSetupChanged ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

## 6.969 Digikam::ItemThumbnailDelegate Class Reference

Inheritance diagram for Digikam::ItemThumbnailDelegate:



### Public Member Functions

- bool `acceptsActivation` (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect) const override

- **ItemThumbnailDelegate** ([ItemCategorizedView](#) \*const parent)
- int **maximumSize** () const
- int **minimumSize** () const
- void **setDefaultViewOptions** (const [QStyleOptionViewItem](#) &option) override
- void **setFlow** ([QListView::Flow](#) flow)

## Public Member Functions inherited from [Digikam::ItemDelegate](#)

- bool **acceptsToolTip** (const [QPoint](#) &pos, const [QRect](#) &visualRect, const [QModelIndex](#) &index, [QRect](#) \*tooltipRect=nullptr) const override
- [QRect](#) **actualPixmapRect** (const [QModelIndex](#) &index) const
- int **calculatethumbSizeToFit** (int ws)
- [ItemCategoryDrawer](#) \* **categoryDrawer** () const
- [QRect](#) **commentsRect** () const
- [QRect](#) **coordinatesIndicatorRect** () const
- [QRect](#) **groupIndicatorRect** () const
- [QRect](#) **imageInformationRect** () const override
- **ItemDelegate** ([QWidget](#) \*const parent)
- void **paint** ([QPainter](#) \*painter, const [QStyleOptionViewItem](#) &option, const [QModelIndex](#) &index) const override
- [QPixmap](#) **pixmapForDrag** (const [QStyleOptionViewItem](#) &option, const [QList](#)< [QModelIndex](#) > &indexes) const override
- [QRect](#) **pixmapRect** () const override
- void **setSpacing** (int spacing) override
- void **setView** ([ItemCategorizedView](#) \*view)
- [QRect](#) **tagsRect** () const

## Public Member Functions inherited from [Digikam::ItemViewDelegate](#)

- bool **acceptsActivation** (const [QPoint](#) &pos, const [QRect](#) &visualRect, const [QModelIndex](#) &index, [QRect](#) \*activationRect=nullptr) const override
- bool **acceptsToolTip** (const [QPoint](#) &pos, const [QRect](#) &visualRect, const [QModelIndex](#) &index, [QRect](#) \*tooltipRect=nullptr) const override
- double **displayRatio** () const
- [QSize](#) **gridSize** () const override
- **ItemViewDelegate** ([QWidget](#) \*const parent)
- void **mouseMoved** ([QMouseEvent](#) \*e, const [QRect](#) &visualRect, const [QModelIndex](#) &index) override
- virtual [QRect](#) **ratingRect** () const
- [QRect](#) **rect** () const
- void **setDefaultViewOptions** (const [QStyleOptionViewItem](#) &option) override
- void **setRatingEdited** (const [QModelIndex](#) &index)
- void **setSpacing** (int spacing) override
- void **setThumbnailSize** (const [ThumbnailSize](#) &thumbSize) override
- [QSize](#) **sizeHint** (const [QStyleOptionViewItem](#) &option, const [QModelIndex](#) &index) const override
- int **spacing** () const
- [ThumbnailSize](#) **thumbnailSize** () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- **DItemDelegate** ([QObject](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Protected Member Functions

- void [updateContentWidth](#) () override
- void [updateRects](#) () override

## Protected Member Functions inherited from [Digikam::ItemDelegate](#)

- void [clearCaches](#) () override
- virtual void [clearModelDataCaches](#) ()
- void [invalidatePaintingCache](#) () override
- **ItemDelegate** ([ItemDelegate::ItemDelegatePrivate](#) &dd, QWidget \*const parent)
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- void [updateSizeRectsAndPixmaps](#) () override

## Protected Member Functions inherited from [Digikam::ItemViewDelegate](#)

- QAbstractItemDelegate \* [asDelegate](#) () override  
*Returns the delegate, typically, the derived class.*
- void **drawAspectRatio** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawColorLabelLine** (QPainter \*p, const QRect &pixRect, int colorId) const
- void **drawComments** (QPainter \*p, const QRect &commentsRect, const QString &comments) const
- void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
- void **drawGroupIndicator** (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
- void **drawImageFormat** (QPainter \*p, const QRect &r, const QString &f, bool drawTop) const
- void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawModificationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
- void **drawPanelSidelcon** (QPainter \*p, bool left, bool right) const
- void **drawPickLabelIcon** (QPainter \*p, const QRect &r, int pickLabel) const
- void **drawRating** (QPainter \*p, const QModelIndex &index, const QRect &ratingRect, int rating, bool isSelected) const
- void **drawSpecialInfo** (QPainter \*p, const QRect &r, const QString &text) const
- void **drawTags** (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const

- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail, bool isGrouped) const
- void **drawTitle** (QPainter \*p, const QRect &titleRect, const QString &title) const
- **ItemViewDelegate** (**ItemViewDelegatePrivate** &dd, QWidget \*const parent)
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- void **prepareRatingPixmaps** (bool composeOverBackground=true)
- QPixmap **ratingPixmap** (int rating, bool selected) const

### Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)

*Declare as slot in the derived class calling this method.*

### Additional Inherited Members

### Signals inherited from [Digikam::ItemViewDelegate](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

### Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

### Static Public Member Functions inherited from [Digikam::ItemDelegate](#)

- static QPixmap **retrieveThumbnailPixmap** (const QModelIndex &index, int thumbnailSize)

### Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

## Protected Slots inherited from [Digikam::ItemDelegate](#)

- void `modelChanged` ()
- void `modelContentsChanged` ()

## Protected Slots inherited from [Digikam::ItemViewDelegate](#)

- void `overlayDestroyed` (QObject \*o) override
- void `slotSetupChanged` ()
- void `slotThemeChanged` ()

## Protected Attributes inherited from [Digikam::ItemViewDelegate](#)

- `ItemViewDelegatePrivate` \*const `d_ptr` = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- `QList< ItemDelegateOverlay * >` `m_overlays`

## 6.969.1 Member Function Documentation

### 6.969.1.1 `acceptsActivation()`

```
bool Digikam::ItemThumbnailDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect ) const [override], [virtual]
```

Reimplemented from [Digikam::ItemDelegate](#).

### 6.969.1.2 `maximumSize()`

```
int Digikam::ItemThumbnailDelegate::maximumSize ( ) const
```

Returns the minimum or maximum viewport size in the limiting dimension, width or height, depending on current flow.

### 6.969.1.3 `setDefaultViewOptions()`

```
void Digikam::ItemThumbnailDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. `option.rect` shall be the viewport rectangle. Call on resize, font change.

Reimplemented from [Digikam::ItemDelegate](#).

#### 6.969.1.4 updateContentWidth()

```
void Digikam::ItemThumbnailDelegate::updateContentWidth ( ) [override], [protected], [virtual]
```

Reimplement this to set contentWidth. This is the maximum width of all content rectangles, typically excluding margins on both sides.

Reimplemented from [Digikam::ItemDelegate](#).

#### 6.969.1.5 updateRects()

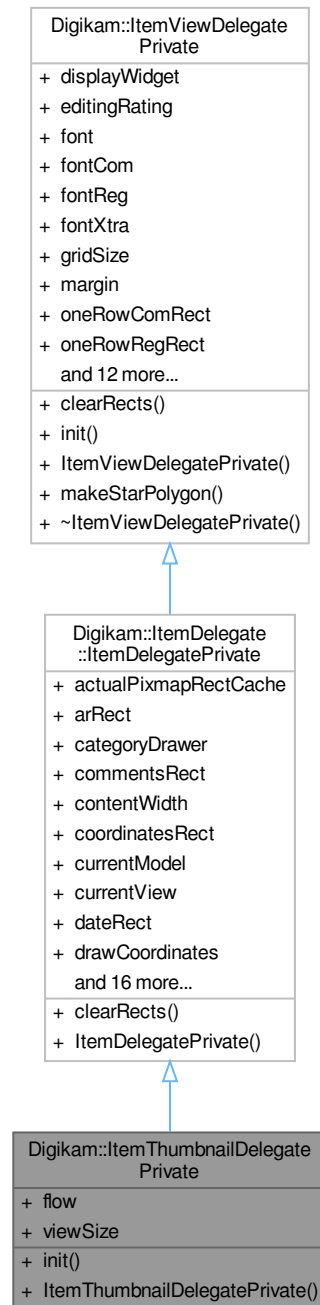
```
void Digikam::ItemThumbnailDelegate::updateRects ( ) [override], [protected], [virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The paint() method operates depending on these rects.

Implements [Digikam::ItemDelegate](#).

## 6.970 Digikam::ItemThumbnailDelegatePrivate Class Reference

Inheritance diagram for Digikam::ItemThumbnailDelegatePrivate:



### Public Member Functions

- void `init` ([ItemThumbnailDelegate](#) \*const q)



## Public Member Functions inherited from [Digikam::ItemDelegate::ItemDelegatePrivate](#)

- void [clearRects](#) () override

## Public Member Functions inherited from [Digikam::ItemViewDelegatePrivate](#)

- void [init](#) ([ItemViewDelegate](#) \*const \_q, QWidget \*const \_widget)
- void [makeStarPolygon](#) ()

## Public Attributes

- QListView::Flow **flow** = QListView::LeftToRight
- QRect **viewSize**

## Public Attributes inherited from [Digikam::ItemDelegate::ItemDelegatePrivate](#)

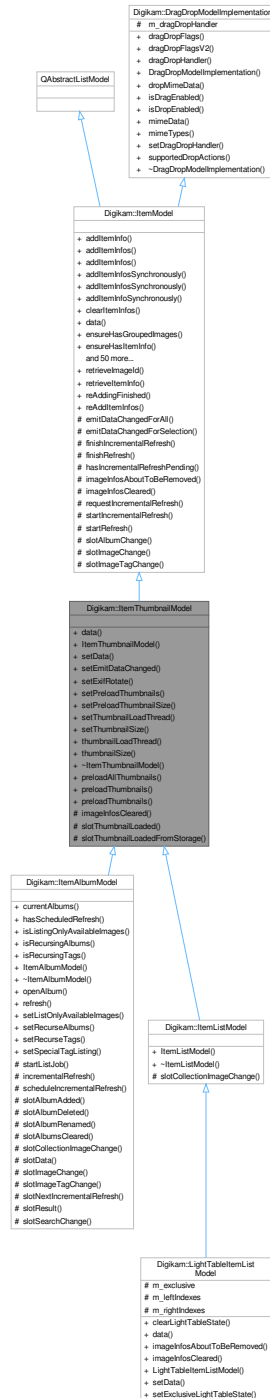
- QCache< int, QRect > **actualPixmapRectCache**
- QRect **arRect**
- [ItemCategoryDrawer](#) \* **categoryDrawer**
- QRect **commentsRect**
- int **contentWidth**
- QRect **coordinatesRect**
- QAbstractItemModel \* **currentModel**
- [ItemCategorizedView](#) \* **currentView**
- QRect **dateRect**
- bool **drawCoordinates**
- bool **drawFocusFrame**
- bool **drawImageFormat**
- bool **drawImageFormatTop**
- bool **drawMouseOverFrame**
- QRect **groupRect**
- QRect **imageInformationRect**
- QRect **modDateRect**
- QRect **nameRect**
- QRect **pickLabelRect**
- QRect **pixmapRect**
- bool **ratingOverThumbnail**
- QRect **resolutionRect**
- QRect **sizeRect**
- QRect **specialInfoRect**
- QRect **tagRect**
- QRect **titleRect**

## Public Attributes inherited from [Digikam::ItemViewDelegatePrivate](#)

- QWidget \* **displayWidget** = nullptr
  - QPersistentModelIndex **editingRating**
  - QFont **font**
  - QFont **fontCom**
  - QFont **fontReg**
  - QFont **fontXtra**
  - QSize **gridSize**
  - int **margin** = 5
  - QRect **oneRowComRect**
  - QRect **oneRowRegRect**
  - QRect **oneRowXtraRect**
  - [ItemViewDelegate](#) \* **q** = nullptr
  - int **radius** = 3
- constant values for drawing*
- QVector< QPixmap > **ratingPixmap** = QVector< QPixmap >(10)
  - QRect **ratingRect**
  - QRect **rect**
  - QPixmap **regPixmap**
  - QPixmap **selPixmap**
  - int **spacing** = 0
  - QPolygon **starPolygon**
  - QSize **starPolygonSize**
  - [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

## 6.971 Digikam::ItemThumbnailModel Class Reference

Inheritance diagram for Digikam::ItemThumbnailModel:



### Public Slots

- void **preloadAllThumbnails** ()
- void **preloadThumbnails** (const QList< [ItemInfo](#) > &)
- void **preloadThumbnails** (const QList< [QModelIndex](#) > &)

## Public Slots inherited from [Digikam::ItemModel](#)

- void **reAddingFinished** ()
- void **reAddItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)

## Signals

- void **thumbnailAvailable** (const QModelIndex &index, int requestedSize)
- void **thumbnailFailed** (const QModelIndex &index, int requestedSize)

## Signals inherited from [Digikam::ItemModel](#)

- void [allRefreshingFinished](#) ()
- void [imageChange](#) (const [ImageChangeset](#) &, const QItemSelection &)
- void [imageInfosAboutToBeAdded](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosAboutToBeRemoved](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosAdded](#) (const QList< [ItemInfo](#) > &infos)
- void [imageInfosRemoved](#) (const QList< [ItemInfo](#) > &infos)
- void [imageTagChange](#) (const [ImageTagChangeset](#) &, const QItemSelection &)
- void [preprocess](#) (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &)
- void **processAdded** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &)
- void [readyForIncrementalRefresh](#) ()

## Public Member Functions

- QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const override
- [ItemThumbnailModel](#) (QWidget \*const parent)
- bool [setData](#) (const QModelIndex &index, const QVariant &value, int role=Qt::DisplayRole) override
- void [setEmitDataChanged](#) (bool emitSignal)
- void **setExifRotate** (bool rotate)
- void [setPreloadThumbnails](#) (bool preload)
- void **setPreloadThumbnailSize** (const [ThumbnailSize](#) &thumbSize)  
*If you want to fix a size for preloading, do it here.*
- void [setThumbnailLoadThread](#) ([ThumbnailLoadThread](#) \*const thread)
- void **setThumbnailSize** (const [ThumbnailSize](#) &thumbSize)  
*Set the thumbnail size to use.*
- [ThumbnailLoadThread](#) \* **thumbnailLoadThread** () const
- [ThumbnailSize](#) **thumbnailSize** () const

## Public Member Functions inherited from [Digikam::ItemModel](#)

- void [addItemInfo](#) (const [ItemInfo](#) &info)
- void **addItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **addItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void **addItemInfosSynchronously** (const QList< [ItemInfo](#) > &infos)
- void **addItemInfosSynchronously** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- void [addItemInfoSynchronously](#) (const [ItemInfo](#) &info)
- void [clearItemInfos](#) ()
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- void [ensureHasGroupedImages](#) (const [ItemInfo](#) &groupLeader)
- void [ensureHasItemInfo](#) (const [ItemInfo](#) &info)

- void **ensureHasItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **ensureHasItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **hasImage** (const [ItemInfo](#) &info) const
- bool **hasImage** (const [ItemInfo](#) &info, const QVariant &extraValue) const
- bool **hasImage** (qulonglong id) const
- bool **hasImage** (qulonglong id, const QVariant &extraValue) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- qulonglong **imageId** (const QModelIndex &index) const
- qulonglong **imageId** (int row) const
- QList< qulonglong > **imageIds** () const
- QList< qulonglong > **imageIds** (const QList< QModelIndex > &indexes) const
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfo](#) **imageInfo** (const QString &filePath) const
- [ItemInfo](#) **imageInfo** (int row) const
- [ItemInfo](#) & **imageInfoRef** (const QModelIndex &index) const
- [ItemInfo](#) & **imageInfoRef** (int row) const
- QList< [ItemInfo](#) > **imageInfos** () const
- QList< [ItemInfo](#) > **imageInfos** (const QList< QModelIndex > &indexes) const
- QList< [ItemInfo](#) > **imageInfos** (const QString &filePath) const
- QModelIndex **index** (int row, int column=0, const QModelIndex &parent=QModelIndex()) const override
- QList< QModelIndex > **indexesForImageId** (qulonglong id) const
- QList< QModelIndex > **indexesForItemInfo** (const [ItemInfo](#) &info) const
- QList< QModelIndex > **indexesForPath** (const QString &filePath) const
- QModelIndex **indexForImageId** (qulonglong id) const
- QModelIndex **indexForImageId** (qulonglong id, const QVariant &extraValue) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info) const
- QModelIndex **indexForItemInfo** (const [ItemInfo](#) &info, const QVariant &extraValue) const
- QModelIndex **indexForPath** (const QString &filePath) const
- bool **isEmpty** () const
- bool **isRefreshing** () const
- int **itemCount** () const
- **ItemModel** (QObject \*const parent=nullptr)
- bool **keepsFilePathCache** () const
- int **numberOfIndexesForImageId** (qulonglong id) const
- int **numberOfIndexesForItemInfo** (const [ItemInfo](#) &info) const
- void **removeIndex** (const QModelIndex &indexes)
- void **removeIndexes** (const QList< QModelIndex > &indexes)
- void **removeItemInfo** (const [ItemInfo](#) &info)
- void **removeItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **removeItemInfos** (const QList< [ItemInfo](#) > &infos, const QList< QVariant > &extraValues)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **setKeepsFilePathCache** (bool keepCache)
- DECLARE\_MODEL\_DRAG\_DROP\_METHODS void **setPreprocessor** (QObject \*const processor)
- void **setSendRemovalSignals** (bool send)
- void **setWatchFlags** (const [DatabaseFields::Set](#) &set)
- QList< [ItemInfo](#) > **uniqueItemInfos** () const
- void **unsetPreprocessor** (QObject \*const processor)

## Public Member Functions inherited from [Digikam::DragDropModelImplementation](#)

- virtual Qt::ItemFlags [dragDropFlags](#) (const QModelIndex &index) const
- Qt::ItemFlags [dragDropFlagsV2](#) (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const
- [DragDropModelImplementation](#) ()=default
- bool **dropMimeData** (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool **isDragEnabled** (const QModelIndex &index) const
- virtual bool **isDropEnabled** (const QModelIndex &index) const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const
- QStringList **mimeTypes** () const
- void [setDragDropHandler](#) ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions [supportedDropActions](#) () const

## Protected Slots

- void **slotThumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QPixmap &thumb)
- void **slotThumbnailLoadedFromStorage** (const [LoadingDescription](#) &loadingDescription, const QPixmap &thumb)

## Protected Slots inherited from [Digikam::ItemModel](#)

- virtual void **slotAlbumChange** (const [AlbumChangeset](#) &changeset)
- virtual void **slotImageChange** (const [ImageChangeset](#) &changeset)
- virtual void **slotImageTagChange** (const [ImageTagChangeset](#) &changeset)

## Protected Member Functions

- void [imageInfosCleared](#) () override

## Protected Member Functions inherited from [Digikam::ItemModel](#)

- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelection** (const QItemSelection &selection)
- void **finishIncrementalRefresh** ()
- void **finishRefresh** ()
- bool **hasIncrementalRefreshPending** () const
- virtual void [imageInfosAboutToBeRemoved](#) (int, int)
- void [requestIncrementalRefresh](#) ()
- void [startIncrementalRefresh](#) ()
- void [startRefresh](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::ItemModel](#)

- enum [ItemModelRoles](#) {  
[ItemModelPointerRole](#) = Qt::UserRole , [ItemModelInternalId](#) = Qt::UserRole + 1 , [ThumbnailRole](#) = Qt::UserRole + 2 , [CreationDateRole](#) = Qt::UserRole + 3 ,  
[ExtraDataRole](#) = Qt::UserRole + 5 , [ExtraDataDuplicateCount](#) = Qt::UserRole + 6 , [LTLeftPanelRole](#) = Qt::UserRole + 50 , [LTRightPanelRole](#) = Qt::UserRole + 51 ,  
[SubclassRoles](#) = Qt::UserRole + 100 , [FilterModelRoles](#) = Qt::UserRole + 500 }

## Static Public Member Functions inherited from [Digikam::ItemModel](#)

- static `qulonglong retrievalmageld` (const `QModelIndex` &index)
- static `ItemInfo retrievalitemInfo` (const `QModelIndex` &index)

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- `AbstractItemDragDropHandler * m_dragDropHandler` = nullptr

## 6.971.1 Constructor & Destructor Documentation

### 6.971.1.1 ItemThumbnailModel()

```
Digikam::ItemThumbnailModel::ItemThumbnailModel (
    QWidget *const parent ) [explicit]
```

An [ItemModel](#) that supports thumbnail loading. You need to set a [ThumbnailLoadThread](#) to enable thumbnail loading. Adjust the thumbnail size to your needs. Note that `setKeepsFilePathCache` is enabled per default.

## 6.971.2 Member Function Documentation

### 6.971.2.1 data()

```
QVariant Digikam::ItemThumbnailModel::data (
    const QModelIndex & index,
    int role = Qt::DisplayRole ) const [override]
```

Handles the ThumbnailRole. If the pixmap is available, returns it in the QVariant. If it still needs to be loaded, returns a null QVariant and emits `thumbnailAvailable()` as soon as it is available.

### 6.971.2.2 imageInfosCleared()

```
void Digikam::ItemThumbnailModel::imageInfosCleared ( ) [override], [protected], [virtual]
```

Called when the internal storage is cleared

Reimplemented from [Digikam::ItemModel](#).

### 6.971.2.3 preloadThumbnails

```
void Digikam::ItemThumbnailModel::preloadThumbnails (
    const QList< ItemInfo > & infos ) [slot]
```

Preload thumbnail for the given infos resp. indexes. Note: Use `setPreloadThumbnails` to automatically preload all entries in the model. Note: This only ensures thumbnail generation. It is not guaranteed that pixmaps are stored in the cache. For thumbnails that are expect to be drawn immediately, include them in `prepareThumbnails()`. Note: Stops preloading of previously added thumbnails.

#### 6.971.2.4 setData()

```
bool Digikam::ItemThumbnailModel::setData (
    const QModelIndex & index,
    const QVariant & value,
    int role = Qt::DisplayRole ) [override]
```

You can override the current thumbnail size by giving an integer value for ThumbnailRole. Set a null QVariant to use the thumbnail size set by [setThumbnailSize\(\)](#) again. The index given here is ignored for this purpose.

#### 6.971.2.5 setEmitDataChanged()

```
void Digikam::ItemThumbnailModel::setEmitDataChanged (
    bool emitSignal )
```

Enable emitting dataChanged() when a thumbnail becomes available. The thumbnailAvailable() signal will be emitted in any case. Default is true.

#### 6.971.2.6 setPreloadThumbnails()

```
void Digikam::ItemThumbnailModel::setPreloadThumbnails (
    bool preload )
```

Enable preloading of thumbnails: If preloading is enabled, for every entry in the model a thumbnail generation is started. Default: false.

#### 6.971.2.7 setThumbnailLoadThread()

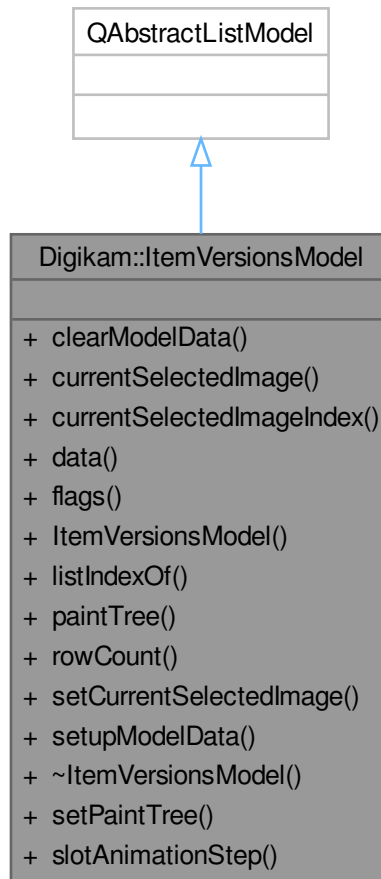
```
void Digikam::ItemThumbnailModel::setThumbnailLoadThread (
    ThumbnailLoadThread *const thread )
```

Enable thumbnail loading and set the thread that shall be used. The thumbnail size of this thread will be adjusted.



## 6.972 Digikam::ItemVersionsModel Class Reference

Inheritance diagram for Digikam::ItemVersionsModel:



### Public Slots

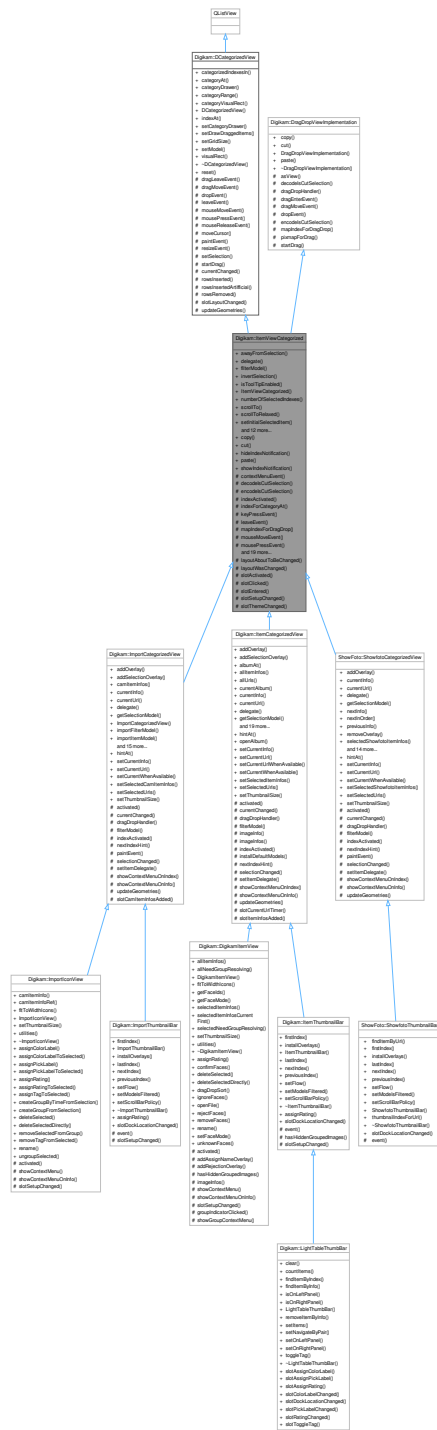
- void **setPaintTree** (bool paint)
- void **slotAnimationStep** ()

### Public Member Functions

- void **clearModelData** ()
- QString **currentSelectedImage** () const
- QModelIndex **currentSelectedImageIndex** () const
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- **ItemVersionsModel** (QObject \*const parent=nullptr)
- int **listIndexOf** (const QString &item) const
- bool **paintTree** () const
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setCurrentSelectedImage** (const QString &path)
- void **setupModelData** (QList< QPair< QString, int > > &data)

# 6.973 Digikam::ItemViewCategorized Class Reference

Inheritance diagram for Digikam::ItemViewCategorized:



### Public Slots

- void **copy** () override
- void **cut** () override

- void **hideIndexNotification** ()
- void **paste** () override
- void **showIndexNotification** (const QModelIndex &index, const QString &message)

### Public Slots inherited from [Digikam::DCategorizedView](#)

- void **reset** () override

### Signals

- void **clicked** (const QMouseEvent \*e, const QModelIndex &index)
- void **entered** (const QMouseEvent \*e, const QModelIndex &index)
- void **keyPressed** (QKeyEvent \*e)
- void **selectionChanged** ()
- void **selectionCleared** ()
- void **viewportClicked** (const QMouseEvent \*e)
- void **zoomInStep** ()
- void **zoomOutStep** ()

### Public Member Functions

- void **awayFromSelection** ()
- [DItemDelegate](#) \* **delegate** () const
- virtual [QSortFilterProxyModel](#) \* **filterModel** () const =0
- void **invertSelection** ()
- bool **isToolTipEnabled** () const
- [ItemViewCategorized](#) (QWidget \*const parent=nullptr)
- int **numberOfSelectedIndexes** () const
- void **scrollTo** (const QModelIndex &index, ScrollHint hint=EnsureVisible) override
- void **scrollToRelaxed** (const QModelIndex &index, ScrollHint hint=EnsureVisible)
- void **setInitialSelectedItem** (bool enabled)
- void **setScrollCurrentToCenter** (bool enabled)
- void **setScrollStepGranularity** (int factor)
- void **setSelectedIndexes** (const QList< QModelIndex > &indexes)
- void **setSpacing** (int spacing)
- void **setToolTipEnabled** (bool enabled)
- void **setUsePointingHandCursor** (bool useCursor)
- void **toFirstIndex** ()
- void **toIndex** (const QModelIndex &index)
- void **toLastIndex** ()
- void **toNextIndex** ()
- void **toPreviousIndex** ()

### Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual [QModelIndexList](#) **categorizedIndexesIn** (const QRect &rect) const
- virtual [QModelIndex](#) **categoryAt** (const QPoint &point) const
- [DCategoryDrawer](#) \* **categoryDrawer** () const
- virtual [QItemSelectionRange](#) **categoryRange** (const QModelIndex &index) const
- virtual [QRect](#) **categoryVisualRect** (const QModelIndex &index) const
- [DCategorizedView](#) (QWidget \*const parent=nullptr)
- [QModelIndex](#) **indexAt** (const QPoint &point) const override
- void **setCategoryDrawer** ([DCategoryDrawer](#) \*categoryDrawer)
- void **setDrawDraggedItems** (bool drawDraggedItems)
- void **setGridSize** (const QSize &size)
- void **setModel** ([QAbstractItemModel](#) \*model) override
- [QRect](#) **visualRect** (const QModelIndex &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

## Protected Slots

- void **layoutAboutToBeChanged** ()
- void **layoutWasChanged** ()
- void **slotActivated** (const QModelIndex &index)
- void **slotClicked** (const QModelIndex &index)
- void **slotEntered** (const QModelIndex &index)
- virtual void **slotSetupChanged** ()
- virtual void **slotThemeChanged** ()

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## Protected Member Functions

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- virtual void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- virtual QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const
- QPixmap  **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** (DItemDelegate \*delegate)
- void **setToolTip** (ItemViewToolTip \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index)  
*Reimplement these in a subclass.*
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

## Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- virtual [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const =0
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

### 6.973.1 Member Function Documentation

#### 6.973.1.1 clicked

```
void Digikam::ItemViewCategorized::clicked (
    const QMouseEvent * e,
    const QModelIndex & index ) [signal]
```

For overlays: Like the respective parent class signals, but with additional info. Do not change the mouse events.

#### 6.973.1.2 filterModel()

```
virtual QSortFilterProxyModel * Digikam::ItemViewCategorized::filterModel ( ) const [pure virtual]
```

Implemented in [ShowFoto::ShowfotoCategorizedView](#), and [Digikam::ImportCategorizedView](#).

#### 6.973.1.3 indexForCategoryAt()

```
QModelIndex Digikam::ItemViewCategorized::indexForCategoryAt (
    const QPoint & pos ) const [protected]
```

Returns an index that is representative for the category at position pos

#### 6.973.1.4 keyPressed

```
void Digikam::ItemViewCategorized::keyPressed (
    QKeyEvent * e ) [signal]
```

Remember you may want to check if the event is accepted or ignored. This signal is emitted after being handled by this widget. You can accept it if ignored.

#### 6.973.1.5 mapIndexForDragDrop()

```
QModelIndex Digikam::ItemViewCategorized::mapIndexForDragDrop (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Note: pure virtual [dragDropHandler\(\)](#) still open from [DragDropViewImplementation](#)

[cut\(\)](#), [copy\(\)](#), [paste\(\)](#), [dragEnterEvent\(\)](#), [dragMoveEvent\(\)](#), [dropEvent\(\)](#), [startDrag\(\)](#) are implemented by [DragDropViewImplementation](#)

Implements [Digikam::DragDropViewImplementation](#).

#### 6.973.1.6 nextIndexHint()

```
QModelIndex Digikam::ItemViewCategorized::nextIndexHint (
    const QModelIndex & indexToAnchor,
    const QItemSelectionRange & removed ) const [protected], [virtual]
```

Assuming the given indexes would be removed (hypothetically!), return the index to be selected instead, starting from anchor. The default implementation returns the next remaining sibling.

Reimplemented in [Digikam::ItemCategorizedView](#), [ShowFoto::ShowfotoCategorizedView](#), and [Digikam::ImportCategorizedView](#).

#### 6.973.1.7 pixmapForDrag()

```
QPixmap Digikam::ItemViewCategorized::pixmapForDrag (
    const QList< QModelIndex > & indexes ) const [override], [protected], [virtual]
```

Creates a pixmap for dragging the given indexes.

Implements [Digikam::DragDropViewImplementation](#).

#### 6.973.1.8 rowsRemoved()

```
void Digikam::ItemViewCategorized::rowsRemoved (
    const QModelIndex & parent,
    int start,
    int end ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::DCategorizedView](#).

### 6.973.1.9 scrollToRelaxed()

```
void Digikam::ItemViewCategorized::scrollToRelaxed (
    const QModelIndex & index,
    ScrollHint hint = EnsureVisible )
```

Like `scrollTo`, but only scrolls if the index is not visible, regardless of hint.

### 6.973.1.10 selectionChanged

```
void Digikam::ItemViewCategorized::selectionChanged ( ) [signal]
```

Emitted when any selection change occurs. Any of the signals below will be emitted before.

### 6.973.1.11 selectionCleared

```
void Digikam::ItemViewCategorized::selectionCleared ( ) [signal]
```

Emitted when the selection is completely cleared.

### 6.973.1.12 setInitialSelectedItem()

```
void Digikam::ItemViewCategorized::setInitialSelectedItem (
    bool enabled )
```

Ensure a initial selected item.

### 6.973.1.13 setScrollCurrentToCenter()

```
void Digikam::ItemViewCategorized::setScrollCurrentToCenter (
    bool enabled )
```

Scroll automatically the current index to center of the view.

### 6.973.1.14 setScrollStepGranularity()

```
void Digikam::ItemViewCategorized::setScrollStepGranularity (
    int factor )
```

Determine a step size for scrolling: The larger this number, the smaller and more precise is the scrolling. Default is 10.

### 6.973.1.15 setSpacing()

```
void Digikam::ItemViewCategorized::setSpacing (
    int spacing )
```

Sets the spacing. Does not use [setSpacing\(\)/spacing\(\)](#) from `QListView`

#### 6.973.1.16 `setUsePointingHandCursor()`

```
void Digikam::ItemViewCategorized::setUsePointingHandCursor (
    bool useCursor )
```

Set if the PointingHand Cursor should be shown over the activation area

#### 6.973.1.17 `showContextMenuOnIndex()`

```
void Digikam::ItemViewCategorized::showContextMenuOnIndex (
    QContextMenuEvent * event,
    const QModelIndex & index ) [protected], [virtual]
```

Reimplemented in [Digikam::ItemCategorizedView](#), [ShowFoto::ShowfotoCategorizedView](#), and [Digikam::ImportCategorizedView](#).

#### 6.973.1.18 `showToolTip()`

```
bool Digikam::ItemViewCategorized::showToolTip (
    const QModelIndex & index,
    QStyleOptionViewItem & option,
    QHelpEvent * e = nullptr ) [protected], [virtual]
```

Provides default behavior, can reimplement in a subclass. Returns true if a tooltip was shown. The help event is optional.

#### 6.973.1.19 `toFirstIndex()`

```
void Digikam::ItemViewCategorized::toFirstIndex ( )
```

Selects the index as current and scrolls to it

#### 6.973.1.20 `viewportClicked`

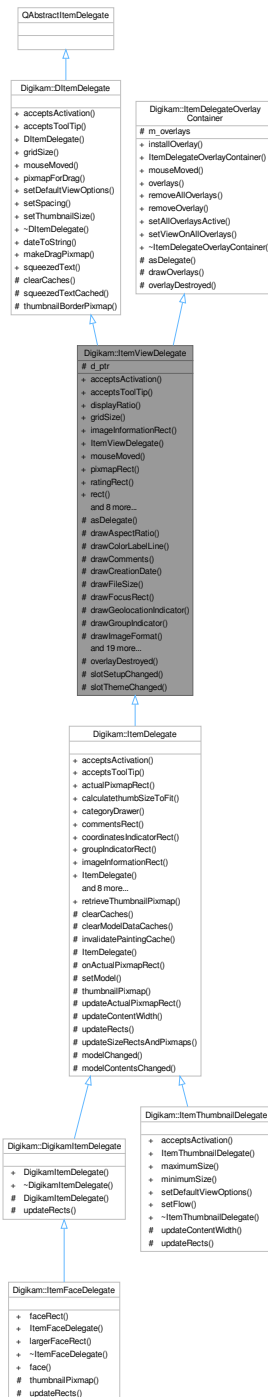
```
void Digikam::ItemViewCategorized::viewportClicked (
    const QMouseEvent * e ) [signal]
```

While [clicked\(\)](#) is emitted with a valid index, this corresponds to clicking on empty space



## 6.974 Digikam::ItemViewDelegate Class Reference

Inheritance diagram for Digikam::ItemViewDelegate:



### Signals

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

## Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Public Member Functions

- bool **acceptsActivation** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool **acceptsToolTip** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double **displayRatio** () const
- QSize **gridSize** () const override
- virtual QRect **imageInformationRect** () const
- **ItemViewDelegate** (QWidget \*const parent)
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect **pixmapRect** () const
- virtual QRect **ratingRect** () const
- QRect **rect** () const
- void **setDefaultViewOptions** (const QStyleOptionViewItem &option) override
- void **setRatingEdited** (const QModelIndex &index)
- void **setSpacing** (int spacing) override
- void **setThumbnailSize** (const ThumbnailSize &thumbSize) override
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int **spacing** () const
- **ThumbnailSize thumbnailSize** () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- **DItemDelegate** (QObject \*const parent=nullptr)
- virtual QPixmap **pixmapForDrag** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const =0

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** (ItemDelegateOverlay \*overlay)
- **ItemDelegateOverlayContainer** ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< ItemDelegateOverlay \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** (ItemDelegateOverlay \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Protected Slots

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

## Protected Member Functions

- QAbstractItemDelegate \* [asDelegate](#) () override  
*Returns the delegate, typically, the derived class.*
- void **drawAspectRatio** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawColorLabelLine** (QPainter \*p, const QRect &pixRect, int colorId) const
- void **drawComments** (QPainter \*p, const QRect &commentsRect, const QString &comments) const
- void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
- void **drawGroupIndicator** (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
- void **drawImageFormat** (QPainter \*p, const QRect &r, const QString &f, bool drawTop) const
- void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawModificationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
- void **drawPanelSidelcon** (QPainter \*p, bool left, bool right) const
- void **drawPickLabelIcon** (QPainter \*p, const QRect &r, int pickLabel) const
- void **drawRating** (QPainter \*p, const QModelIndex &index, const QRect &ratingRect, int rating, bool isSelected) const
- void **drawSpecialInfo** (QPainter \*p, const QRect &r, const QString &text) const
- void **drawTags** (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const
- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail, bool isGrouped) const
- void **drawTitle** (QPainter \*p, const QRect &titleRect, const QString &title) const
- virtual void **invalidatePaintingCache** ()
- **ItemViewDelegate** ([ItemViewDelegatePrivate](#) &dd, QWidget \*const parent)
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- void **prepareRatingPixmap** (bool composeOverBackground=true)
- QPixmap **ratingPixmap** (int rating, bool selected) const
- virtual void **updateSizeRectsAndPixmap** ()=0

## Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- virtual void **clearCaches** ()
- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)

*Declare as slot in the derived class calling this method.*

## Protected Attributes

- [ItemViewDelegatePrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- `QList< ItemDelegateOverlay * > m_overlays`

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static `QString dateToString (const QDateTime &datetime)`
- static `QPixmap makeDragPixmap (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())`
- static `QString squeezedText (const QFontMetrics &fm, int width, const QString &text)`

## 6.974.1 Member Function Documentation

### 6.974.1.1 `acceptsActivation()`

```
bool Digikam::ItemViewDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect = nullptr ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

### 6.974.1.2 `acceptsToolTip()`

```
bool Digikam::ItemViewDelegate::acceptsToolTip (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * tooltipRect = nullptr ) const [override], [virtual]
```

These methods take four parameters: The position on viewport, the rect on viewport, the index, and optionally a parameter into which, if the return value is true, a rectangle can be written for which the return value will be true as well.

Implements [Digikam::DItemDelegate](#).

### 6.974.1.3 `asDelegate()`

```
QAbstractItemDelegate * Digikam::ItemViewDelegate::asDelegate ( ) [override], [protected], [virtual]
```

Implements [Digikam::ItemDelegateOverlayContainer](#).

#### 6.974.1.4 drawThumbnail()

```
QRect Digikam::ItemViewDelegate::drawThumbnail (
    QPainter * p,
    const QRect & thumbRect,
    const QPixmap & background,
    const QPixmap & thumbnail,
    bool isGrouped ) const [protected]
```

Use the tool methods for painting in subclasses

#### 6.974.1.5 gridSize()

```
QSize Digikam::ItemViewDelegate::gridSize ( ) const [override], [virtual]
```

Returns the gridsize to be set by the view. It's sizeHint plus spacing.

Implements [Digikam::DItemDelegate](#).

#### 6.974.1.6 imageInformationRect()

```
QRect Digikam::ItemViewDelegate::imageInformationRect ( ) const [virtual]
```

Returns the area where the image information is drawn, or null if empty / not supported. The image information is textual or graphical information, but not the pixmap. The [ratingRect\(\)](#) will e.g. typically be contained in this area.

Reimplemented in [Digikam::ItemDelegate](#).

#### 6.974.1.7 mouseMoved()

```
void Digikam::ItemViewDelegate::mouseMoved (
    QMouseEvent * e,
    const QRect & visualRect,
    const QModelIndex & index ) [override], [virtual]
```

##### Note

to be called by [ItemViewCategorized](#) only

Implements [Digikam::DItemDelegate](#).

#### 6.974.1.8 pixmapRect()

```
QRect Digikam::ItemViewDelegate::pixmapRect ( ) const [virtual]
```

Returns the area where the pixmap is drawn, or null if not supported.

Reimplemented in [Digikam::ItemDelegate](#).

### 6.974.1.9 ratingPixmap()

```
QPixmap Digikam::ItemViewDelegate::ratingPixmap (
    int rating,
    bool selected ) const [protected]
```

Returns the relevant pixmap from the cached rating pixmaps.

### 6.974.1.10 ratingRect()

```
QRect Digikam::ItemViewDelegate::ratingRect ( ) const [virtual]
```

Returns the rectangle where the rating is drawn, or a null rectangle if not supported.

### 6.974.1.11 setDefaultViewOptions()

```
void Digikam::ItemViewDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. `option.rect` shall be the viewport rectangle. Call on resize, font change.

Implements [Digikam::DItemDelegate](#).

### 6.974.1.12 setRatingEdited()

```
void Digikam::ItemViewDelegate::setRatingEdited (
    const QModelIndex & index )
```

Can be used to temporarily disable drawing of the rating. Call with `QModelIndex()` afterwards.

### 6.974.1.13 setSpacing()

```
void Digikam::ItemViewDelegate::setSpacing (
    int spacing ) [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

### 6.974.1.14 setThumbnailSize()

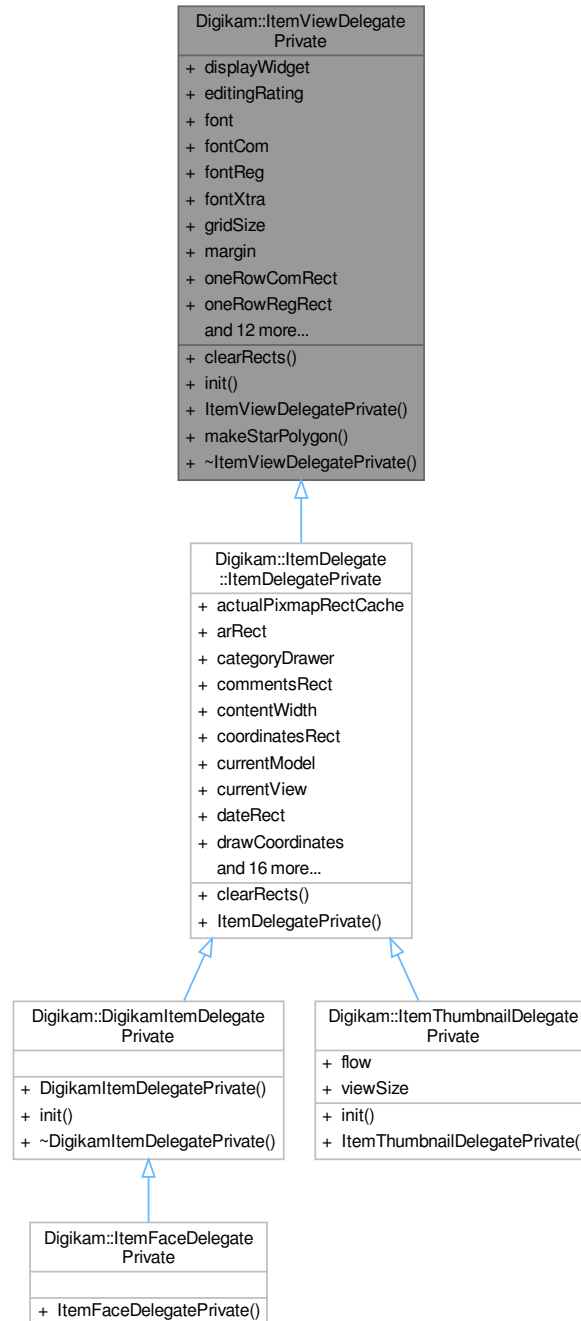
```
void Digikam::ItemViewDelegate::setThumbnailSize (
    const ThumbnailSize & thumbSize ) [override], [virtual]
```

You must set these options from the view

Implements [Digikam::DItemDelegate](#).

## 6.975 Digikam::ItemViewDelegatePrivate Class Reference

Inheritance diagram for Digikam::ItemViewDelegatePrivate:



### Public Member Functions

- virtual void `clearRects` ()
- void `init` (`ItemViewDelegate *const _q`, `QWidget *const _widget`)
- void `makeStarPolygon` ()

## Public Attributes

- QWidget \* **displayWidget** = nullptr
  - QPersistentModelIndex **editingRating**
  - QFont **font**
  - QFont **fontCom**
  - QFont **fontReg**
  - QFont **fontXtra**
  - QSize **gridSize**
  - int **margin** = 5
  - QRect **oneRowComRect**
  - QRect **oneRowRegRect**
  - QRect **oneRowXtraRect**
  - [ItemViewDelegate](#) \* **q** = nullptr
  - int **radius** = 3
- constant values for drawing*
- QVector< QPixmap > **ratingPixmaps** = QVector< QPixmap >(10)
  - QRect **ratingRect**
  - QRect **rect**
  - QPixmap **regPixmap**
  - QPixmap **selPixmap**
  - int **spacing** = 0
  - QPolygon **starPolygon**
  - QSize **starPolygonSize**
  - [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

## 6.975.1 Member Function Documentation

### 6.975.1.1 clearRects()

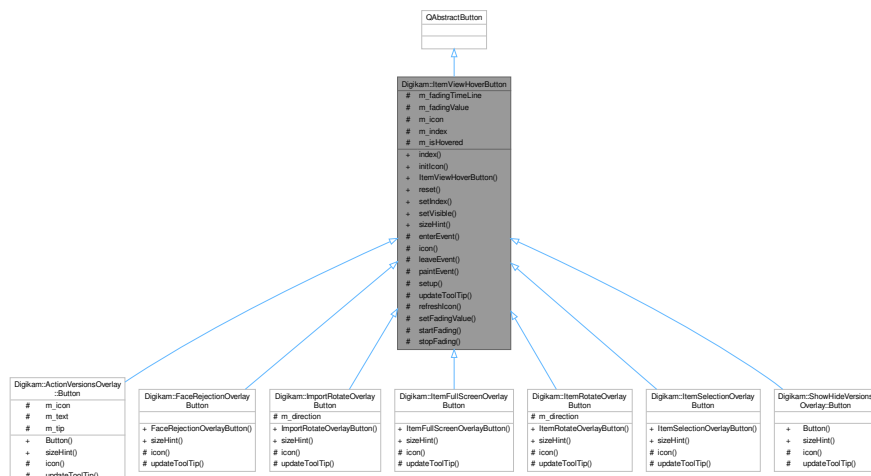
```
void Digikam::ItemViewDelegatePrivate::clearRects ( ) [virtual]
```

Resets cached rects. Remember to reimplement in subclass for added rects.

Reimplemented in [Digikam::ItemDelegate::ItemDelegatePrivate](#).

## 6.976 Digikam::ItemViewHoverButton Class Reference

Inheritance diagram for Digikam::ItemViewHoverButton:





## Public Member Functions

- QModelIndex **index** () const
- void **initIcon** ()
- **ItemViewHoverButton** (QAbstractItemView \*const parentView)
- void **reset** ()
- void **setIndex** (const QModelIndex &index)
- void **setVisible** (bool visible) override
- QSize **sizeHint** () const override=0

## Protected Slots

- void **refreshIcon** ()
- void **setFadingValue** (int value)
- void **startFading** ()
- void **stopFading** ()

## Protected Member Functions

- void **enterEvent** (QEnterEvent \*event)
- virtual QIcon **icon** ()=0
- void **leaveEvent** (QEvent \*event)
- void **paintEvent** (QPaintEvent \*event)
- void **setup** ()
- virtual void **updateToolTip** ()

## Protected Attributes

- QTimeline \* **m\_fadingTimeLine** = nullptr
- int **m\_fadingValue** = 0
- QIcon **m\_icon**
- QPersistentModelIndex **m\_index**
- bool **m\_isHovered** = false

## 6.976.1 Member Function Documentation

### 6.976.1.1 icon()

```
virtual QIcon Digikam::ItemViewHoverButton::icon ( ) [protected], [pure virtual]
```

Return your icon here. Will be queried again on toggle.

Implemented in [Digikam::FaceRejectionOverlayButton](#), [Digikam::ItemFullScreenOverlayButton](#), [Digikam::ItemRotateOverlayButton](#), [Digikam::ItemSelectionOverlayButton](#), and [Digikam::ImportRotateOverlayButton](#).

### 6.976.1.2 setup()

```
void Digikam::ItemViewHoverButton::setup ( ) [protected]
```

to call in children class constructors to init signal/slot connections.

### 6.976.1.3 sizeHint()

```
QSize Digikam::ItemViewHoverButton::sizeHint ( ) const [override], [pure virtual]
```

Reimplement to match the size of your icon

Implemented in [Digikam::FaceRejectionOverlayButton](#), [Digikam::ItemFullScreenOverlayButton](#), [Digikam::ItemRotateOverlayButton](#), [Digikam::ItemSelectionOverlayButton](#), and [Digikam::ImportRotateOverlayButton](#).

### 6.976.1.4 updateToolTip()

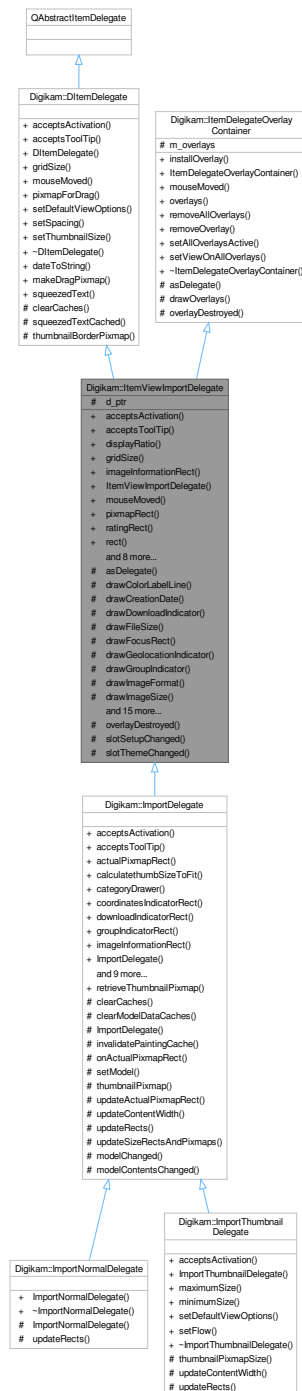
```
void Digikam::ItemViewHoverButton::updateToolTip ( ) [protected], [virtual]
```

Optionally update tooltip here. Will be called again on state change.

Reimplemented in [Digikam::FaceRejectionOverlayButton](#), [Digikam::ItemFullScreenOverlayButton](#), [Digikam::ItemRotateOverlayButton](#), [Digikam::ItemSelectionOverlayButton](#), and [Digikam::ImportRotateOverlayButton](#).

## 6.977 Digikam::ItemViewImportDelegate Class Reference

Inheritance diagram for Digikam::ItemViewImportDelegate:



### Signals

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

## Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Public Member Functions

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double **displayRatio** () const
- QSize [gridSize](#) () const override
- virtual QRect [imageInformationRect](#) () const
- **ItemViewImportDelegate** (QWidget \*const parent)
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect [pixmapRect](#) () const
- virtual QRect [ratingRect](#) () const
- QRect **rect** () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setRatingEdited](#) (const QModelIndex &index)
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from [DItemDelegate](#)*
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int **spacing** () const
- [ThumbnailSize](#) **thumbnailSize** () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- **DItemDelegate** (QObject \*const parent=nullptr)
- virtual QPixmap **pixmapForDrag** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const =0

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Protected Slots

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

## Protected Member Functions

- QAbstractItemDelegate \* [asDelegate](#) () override  
*Returns the delegate, typically, the derived class.*
- void **drawColorLabelLine** (QPainter \*p, const QRect &pixRect, int colorId) const
- void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawDownloadIndicator** (QPainter \*p, const QRect &r, int itemType) const
- void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
- void **drawGroupIndicator** (QPainter \*p, const QRect &r, int numberOfGroupedImages, bool open) const
- void **drawImageFormat** (QPainter \*p, const QRect &dimsRect, const QString &mime) const
- void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawLockIndicator** (QPainter \*p, const QRect &r, int lockStatus) const
- void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
- void **drawPickLabelIcon** (QPainter \*p, const QRect &r, int pickLabel) const
- void **drawRating** (QPainter \*p, const QModelIndex &index, const QRect &[ratingRect](#), int rating, bool isSelected) const
- void **drawTags** (QPainter \*p, const QRect &r, const QString &tagsString, bool isSelected) const
- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail) const  
*Use the tool methods for painting in subclasses.*
- virtual void [invalidatePaintingCache](#) ()  
*reimplement these in subclasses*
- **ItemViewImportDelegate** ([ItemViewImportDelegatePrivate](#) &dd, QWidget \*const parent)
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- void [prepareRatingPixmap](#) (bool composeOverBackground=true)
- QPixmap **ratingPixmap** (int rating, bool selected) const  
*Returns the relevant pixmap from the cached rating pixmaps.*
- virtual void **updateSizeRectsAndPixmap** ()=0

## Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- virtual void **clearCaches** ()
- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)  
*Declare as slot in the derived class calling this method.*

## Protected Attributes

- [ItemViewImportDelegatePrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- `QList< ItemDelegateOverlay * > m_overlays`

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static `QString dateToString (const QDateTime &datetime)`
- static `QPixmap makeDragPixmap (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())`
- static `QString squeezedText (const QFontMetrics &fm, int width, const QString &text)`

## 6.977.1 Member Function Documentation

### 6.977.1.1 `acceptsActivation()`

```
bool Digikam::ItemViewImportDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect = nullptr ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

### 6.977.1.2 `acceptsToolTip()`

```
bool Digikam::ItemViewImportDelegate::acceptsToolTip (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * tooltipRect = nullptr ) const [override], [virtual]
```

These methods take four parameters: The position on viewport, the rect on viewport, the index, and optionally a parameter into which, if the return value is true, a rectangle can be written for which the return value will be true as well.

Implements [Digikam::DItemDelegate](#).

### 6.977.1.3 `asDelegate()`

```
QAbstractItemDelegate * Digikam::ItemViewImportDelegate::asDelegate ( ) [override], [protected], [virtual]
```

Implements [Digikam::ItemDelegateOverlayContainer](#).

#### 6.977.1.4 gridSize()

```
QSize Digikam::ItemViewImportDelegate::gridSize ( ) const [override], [virtual]
```

Returns the gridsize to be set by the view. It's sizeHint plus spacing.

Implements [Digikam::DItemDelegate](#).

#### 6.977.1.5 imageInformationRect()

```
QRect Digikam::ItemViewImportDelegate::imageInformationRect ( ) const [virtual]
```

Returns the area where the image information is drawn, or null if empty / not supported. The image information is textual or graphical information, but not the pixmap. The [ratingRect\(\)](#) will e.g. typically be contained in this area.

Reimplemented in [Digikam::ImportDelegate](#).

#### 6.977.1.6 invalidatePaintingCache()

```
void Digikam::ItemViewImportDelegate::invalidatePaintingCache ( ) [protected], [virtual]
```

Reimplemented in [Digikam::ImportDelegate](#).

#### 6.977.1.7 mouseMoved()

```
void Digikam::ItemViewImportDelegate::mouseMoved (
    QMouseEvent * e,
    const QRect & visualRect,
    const QModelIndex & index ) [override], [virtual]
```

##### Note

to be called by [ItemViewCategorized](#) only

Implements [Digikam::DItemDelegate](#).

#### 6.977.1.8 pixmapRect()

```
QRect Digikam::ItemViewImportDelegate::pixmapRect ( ) const [virtual]
```

Returns the area where the pixmap is drawn, or null if not supported

Reimplemented in [Digikam::ImportDelegate](#).

#### 6.977.1.9 prepareRatingPixmaps()

```
void Digikam::ItemViewImportDelegate::prepareRatingPixmaps (
    bool composeOverBackground = true ) [protected]
```

Please call this method after [prepareBackground\(\)](#) and when `d->ratingPixmap` is set

#### 6.977.1.10 ratingRect()

```
QRect Digikam::ItemViewImportDelegate::ratingRect ( ) const [virtual]
```

Returns the rectangle where the rating is drawn, or a null rectangle if not supported.

#### 6.977.1.11 setDefaultViewOptions()

```
void Digikam::ItemViewImportDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. option.rect shall be the viewport rectangle. Call on resize, font change.

Implements [Digikam::DItemDelegate](#).

#### 6.977.1.12 setRatingEdited()

```
void Digikam::ItemViewImportDelegate::setRatingEdited (
    const QModelIndex & index )
```

Can be used to temporarily disable drawing of the rating. Call with QModelIndex() afterwards.

#### 6.977.1.13 setSpacing()

```
void Digikam::ItemViewImportDelegate::setSpacing (
    int spacing ) [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

#### 6.977.1.14 setThumbnailSize()

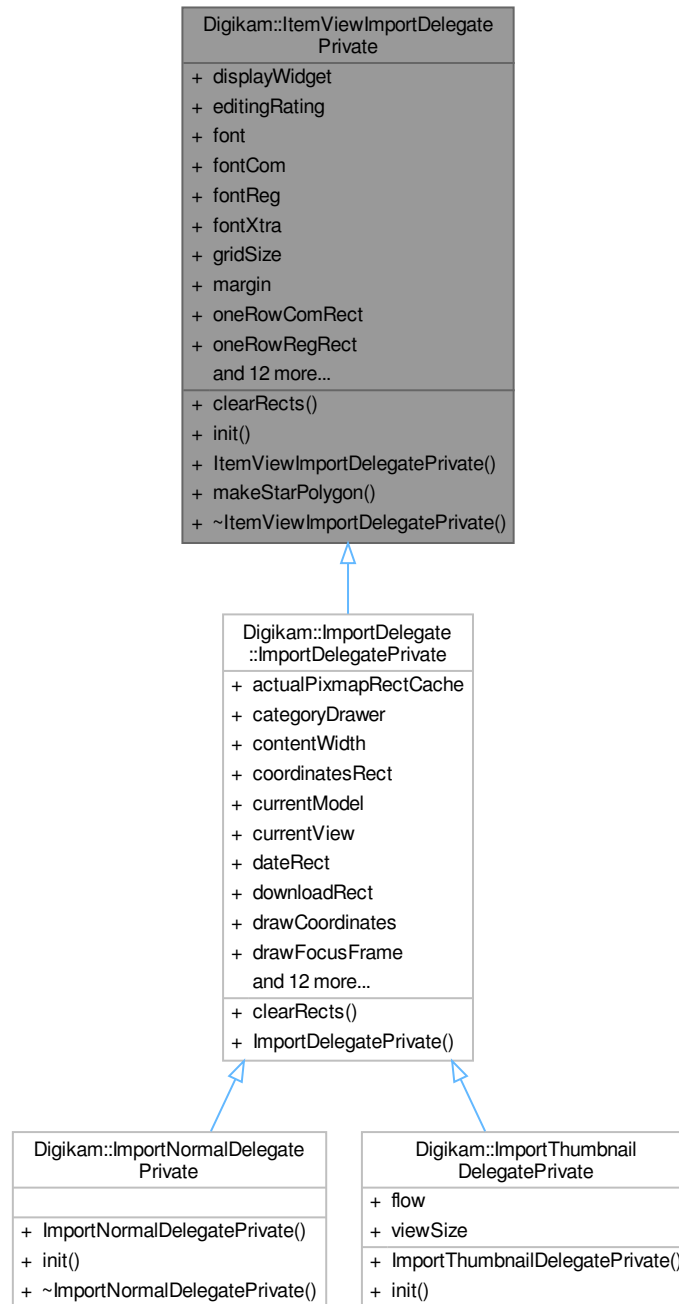
```
void Digikam::ItemViewImportDelegate::setThumbnailSize (
    const ThumbnailSize & thumbSize ) [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).



## 6.978 Digikam::ItemViewImportDelegatePrivate Class Reference

Inheritance diagram for Digikam::ItemViewImportDelegatePrivate:



### Public Member Functions

- virtual void `clearRects` ()  
*Resets cached rects. Remember to reimplement in subclass for added rects.*
- void `init` (`ItemViewImportDelegate *const _q`, `QWidget *const _widget`)
- void `makeStarPolygon` ()

## Public Attributes

- QWidget \* **displayWidget** = nullptr
- QPersistentModelIndex **editingRating**
- QFont **font**
- QFont **fontCom**
- QFont **fontReg**
- QFont **fontXtra**
- QSize **gridSize**
- int **margin** = 5
- QRect **oneRowComRect**
- QRect **oneRowRegRect**
- QRect **oneRowXtraRect**
- [ItemViewImportDelegate](#) \* **q** = nullptr
- int **radius** = 3
  - constant values for drawing*
- QVector< QPixmap > **ratingPixmap** = QVector< QPixmap >(10)
- QRect **ratingRect**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- QPolygon **starPolygon**
- QSize **starPolygonSize**
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

## 6.978.1 Member Function Documentation

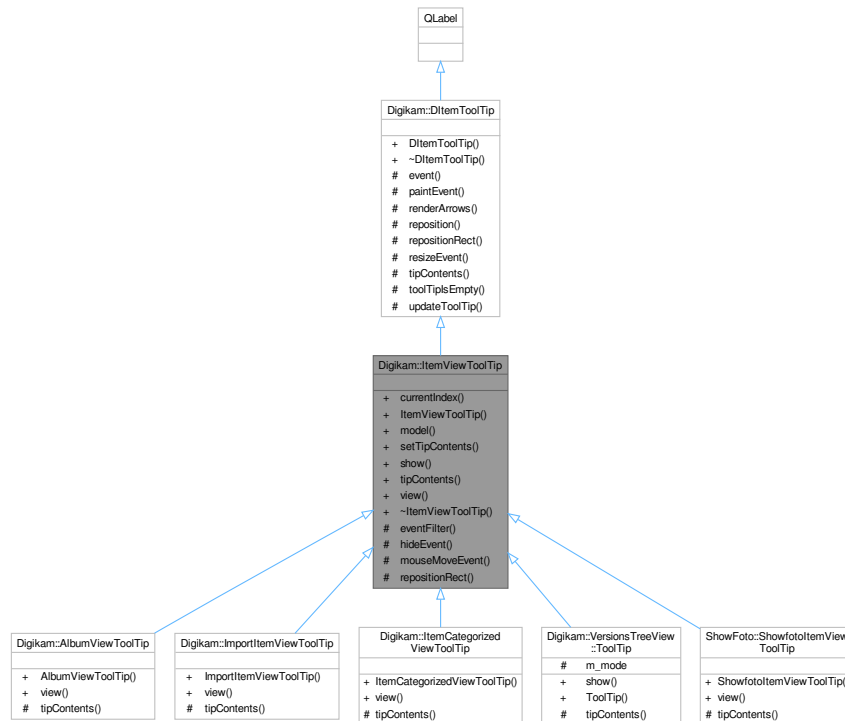
### 6.978.1.1 clearRects()

```
void Digikam::ItemViewImportDelegatePrivate::clearRects ( ) [virtual]
```

Reimplemented in [Digikam::ImportDelegate::ImportDelegatePrivate](#).

## 6.979 Digikam::ItemViewToolTip Class Reference

Inheritance diagram for Digikam::ItemViewToolTip:



### Public Member Functions

- QModelIndex **currentIndex** () const
- **ItemViewToolTip** (QAbstractItemView \*const view)
- QAbstractItemModel \* **model** () const
- void **setTipContents** (const QString &tipContents)
- void **show** (const QStyleOptionViewItem &option, const QModelIndex &index)
- QString **tipContents** () override
- QAbstractItemView \* **view** () const

### Public Member Functions inherited from Digikam::DItemToolTip

- **DItemToolTip** (QWidget \*const parent=nullptr)

### Protected Member Functions

- bool **eventFilter** (QObject \*o, QEvent \*e) override
- void **hideEvent** (QHideEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*e) override
- QRect **repositionRect** () override

## Protected Member Functions inherited from [Digikam::DItemToolTip](#)

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- void **resizeEvent** (QResizeEvent \*) override
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()

### 6.979.1 Member Function Documentation

#### 6.979.1.1 repositionRect()

```
QRect Digikam::ItemViewToolTip::repositionRect ( ) [override], [protected], [virtual]
```

Implements [Digikam::DItemToolTip](#).

#### 6.979.1.2 show()

```
void Digikam::ItemViewToolTip::show (
    const QStyleOptionViewItem & option,
    const QModelIndex & index )
```

Show the tooltip for the given item. The rect of the given option is taken as area for which the tooltip is shown.

#### 6.979.1.3 tipContents()

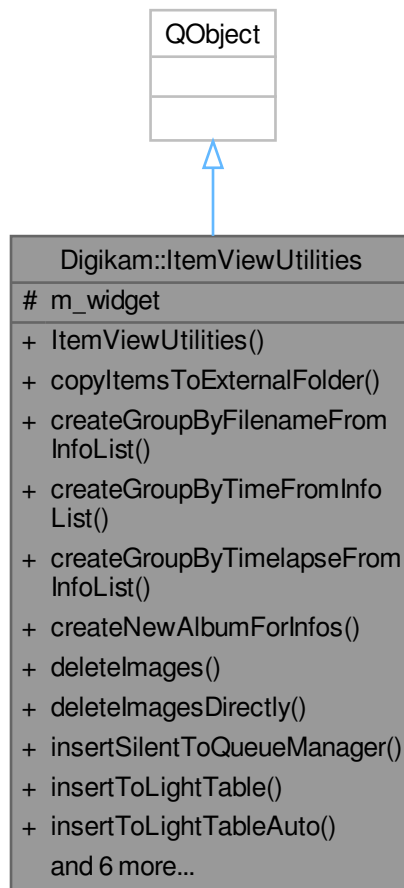
```
QString Digikam::ItemViewToolTip::tipContents ( ) [override], [virtual]
```

Default implementation is based on `setTipContents()`. Reimplement if you dynamically provide the contents.

Implements [Digikam::DItemToolTip](#).

## 6.980 Digikam::ItemViewUtilities Class Reference

Inheritance diagram for Digikam::ItemViewUtilities:



### Public Types

- enum **DeleteMode** { **DeletePermanently** = 1 , **DeleteUseTrash** = 2 }

### Public Slots

- void **copyItemsToExternalFolder** (const QList< [ItemInfo](#) > &infos)
- void **createGroupByFilenameFromInfoList** (const [ItemInfoList](#) &itemInfoList)
- void **createGroupByTimeFromInfoList** (const [ItemInfoList](#) &itemInfoList)
- void **createGroupByTimelapseFromInfoList** (const [ItemInfoList](#) &itemInfoList)
- void **createNewAlbumForInfos** (const QList< [ItemInfo](#) > &infos, [Album](#) \*currentAlbum)
- bool **deleteImages** (const QList< [ItemInfo](#) > &infos, const DeleteMode deleteMode)
- void **deleteImagesDirectly** (const QList< [ItemInfo](#) > &infos, const DeleteMode deleteMode)
- void **insertSilentToQueueManager** (const QList< [ItemInfo](#) > &list, const [ItemInfo](#) &currentInfo, int queueid)

- void **insertToLightTable** (const QList< [ItemInfo](#) > &list, const [ItemInfo](#) &current, bool addTo)
- void **insertToLightTableAuto** (const QList< [ItemInfo](#) > &all, const QList< [ItemInfo](#) > &selected, const [ItemInfo](#) &current)
- void **insertToQueueManager** (const QList< [ItemInfo](#) > &list, const [ItemInfo](#) &currentInfo, bool newQueue)
- void **notifyFileContentChanged** (const QList< QUrl > &urls)
- void **openInfos** (const [ItemInfo](#) &info, const QList< [ItemInfo](#) > &allInfosToOpen, [Album](#) \*currentAlbum)
- void **openInfosWithDefaultApplication** (const QList< [ItemInfo](#) > &allInfosToOpen)
- void **rename** (const QUrl &imageUrl, const QString &newName, bool overwrite=false)
- void **setAsAlbumThumbnail** ([Album](#) \*album, const [ItemInfo](#) &itemInfo)

### Signals

- void **editorCurrentUrlChanged** (const QUrl &url)
- void **signalImagesDeleted** (const QList< qlonglong > &imageIds)

### Public Member Functions

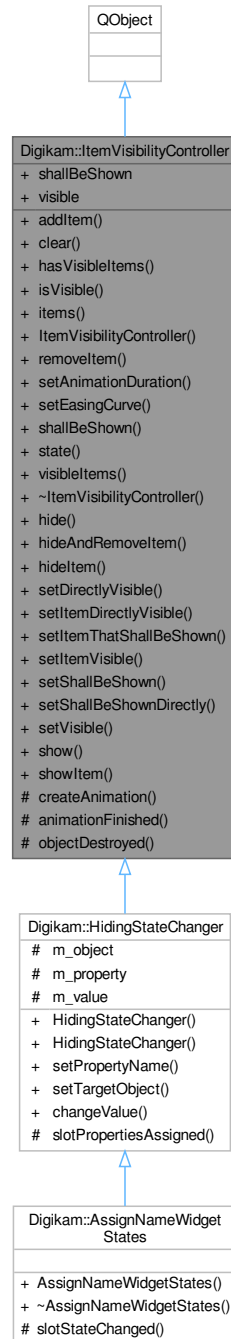
- **ItemViewUtilities** (QWidget \*const parentWidget)

### Protected Attributes

- QWidget \* **m\_widget** = nullptr

## 6.981 Digikam::ItemVisibilityController Class Reference

Inheritance diagram for Digikam::ItemVisibilityController:



### Public Types

- enum `IncludeFadingOutMode` { `IncludeFadingOut` , `ExcludeFadingOut` }
- enum `State` { `Hidden` , `FadingIn` , `Visible` , `FadingOut` }

### Public Slots

- void **hide** ()
- void **hideAndRemoveItem** (QObject \*item)
- void **hideItem** (QObject \*item)
- void **setDirectlyVisible** (bool visible)
- void **setItemDirectlyVisible** (QObject \*item, bool visible)
- void **setItemThatShallBeShown** (QObject \*item)
- void **setItemVisible** (QObject \*item, bool visible)
- void **setShallBeShown** (bool shallBeShown)
- void **setShallBeShownDirectly** (bool shallBeShown)
- void **setVisible** (bool visible)
- void **show** ()
- void **showItem** (QObject \*item)

### Signals

- void **hiddenAndRemoved** (QObject \*item)
- void **propertiesAssigned** (bool visible)
- void **propertiesAssigned** (QObject \*item, bool visible)

### Public Member Functions

- void **addItem** (QObject \*const object)
- void **clear** ()
- bool **hasVisibleItems** (**IncludeFadingOutMode** mode=**IncludeFadingOut**) const
- bool **isVisible** () const
- QList< QObject \* > **items** () const
- **ItemVisibilityController** (QObject \*const parent=nullptr)
- void **removeItem** (QObject \*const object)
- void **setAnimationDuration** (int msec)
- void **setEasingCurve** (const QEasingCurve &easing)
- bool **shallBeShown** () const
- **State** **state** () const
- QList< QObject \* > **visibleItems** (**IncludeFadingOutMode** mode=**IncludeFadingOut**) const

### Protected Slots

- void **animationFinished** ()
- void **objectDestroyed** (QObject \*)

### Protected Member Functions

- virtual QPropertyAnimation \* **createAnimation** (QObject \*item)

### Properties

- bool **shallBeShown**
- bool **visible**

## 6.981.1 Member Enumeration Documentation

### 6.981.1.1 IncludeFadingOutMode

```
enum Digikam::ItemVisibilityController::IncludeFadingOutMode
```



## Enumerator

IncludeFadingOut	In addition to items visible or fading in, return those fading out.
ExcludeFadingOut	Do not return those items currently fading out (soon to be hidden)

**6.981.1.2 State**

```
enum Digikam::ItemVisibilityController::State
```

This class handles complex visibility situations for items. There is a 3-tiered approach: 1) shallBeShown determines if the items shall at any time be shown. If it is false, items will never be shown. Default is true, so you can ignore this setting. 2) visible determines if the items shall be shown now. Only takes effect if shallBeShown is true. Default is false: Initially, controlled items are hidden. 3) Opacity and individual item visibility: When showing, items are first set to individually visible, then their opacity is increased from 0 to 1. When hiding, opacity is first decreased from 1 to 0, then they are set individually to hidden. Different types of items can be handled:

- a group of items with an "opacity" and "visible" property
- a single item with an "opacity" and "visible" property
- a proxy object with these properties (see above)

**6.981.2 Member Function Documentation****6.981.2.1 addItem()**

```
void Digikam::ItemVisibilityController::addItem (
    QObject *const object )
```

Add and remove objects. The given objects shall provide an "opacity" and a "visible" property. You can, for convenience, use a [ItemVisibilityControllerPropertyObject](#) as a value container, if your items do not provide these properties directly. No ownership is taken, so the objects should live as long as this object is used.

**6.981.2.2 clear()**

```
void Digikam::ItemVisibilityController::clear ( )
```

Remove all animations

**6.981.2.3 createAnimation()**

```
QPropertyAnimation * Digikam::ItemVisibilityController::createAnimation (
    QObject * item ) [protected], [virtual]
```

Creates the animation for showing and hiding the given item. The item is given for information only, you do not need to use it. The default implementation creates an animation for "opacity" from 0.0 to 1.0, using default easing curve and duration, which can and will be changed by setEasingCurve and setAnimationDuration.

#### 6.981.2.4 hasVisibleItems()

```
bool Digikam::ItemVisibilityController::hasVisibleItems (
    IncludeFadingOutMode mode = IncludeFadingOut ) const
```

This returns the "result" of isVisible and shallBeShown: Something is indeed visible on the scene. Also returns false if no items are available.

#### 6.981.2.5 hiddenAndRemoved

```
void Digikam::ItemVisibilityController::hiddenAndRemoved (
    QObject * item ) [signal]
```

Emitted when hideAndRemoveItem has finished

#### 6.981.2.6 hideAndRemoveItem

```
void Digikam::ItemVisibilityController::hideAndRemoveItem (
    QObject * item ) [slot]
```

Hide the item, and then remove it. When finished, [hiddenAndRemoved\(\)](#) is emitted.

#### 6.981.2.7 items()

```
QList< QObject * > Digikam::ItemVisibilityController::items ( ) const
```

Returns all items under control

#### 6.981.2.8 propertiesAssigned [1/2]

```
void Digikam::ItemVisibilityController::propertiesAssigned (
    bool visible ) [signal]
```

Emitted when the (main) transition has finished

#### 6.981.2.9 propertiesAssigned [2/2]

```
void Digikam::ItemVisibilityController::propertiesAssigned (
    QObject * item,
    bool visible ) [signal]
```

Emitted when a transition for a single item finished (see setItemVisible())

#### 6.981.2.10 setEasingCurve()

```
void Digikam::ItemVisibilityController::setEasingCurve (
    const QEasingCurve & easing )
```

Allows to change the default parameters of all animations.

### 6.981.2.11 setItemThatShallBeShown

```
void Digikam::ItemVisibilityController::setItemThatShallBeShown (
    QObject * item ) [slot]
```

Sets a single item to be shown. Calling setVisible() will effectively effect only this single item, as if calling setItemVisible(). Reset by calling with 0 or [setShallBeShown\(\)](#).

### 6.981.2.12 setShallBeShown

```
void Digikam::ItemVisibilityController::setShallBeShown (
    bool shallBeShown ) [slot]
```

Adjusts the first condition - the items are shown if shallBeShown is true and isVisible is true

### 6.981.2.13 show

```
void Digikam::ItemVisibilityController::show ( ) [slot]
```

Adjusts the main condition. All items are affected. If any items were shown or hidden separately, they will be resynchronized. "Directly" means no animation is employed.

### 6.981.2.14 showItem

```
void Digikam::ItemVisibilityController::showItem (
    QObject * item ) [slot]
```

Shows or hides a single item. The item's status is changed individually. The next call to the "global" method will take precedence again. "Directly" means no animation is employed.

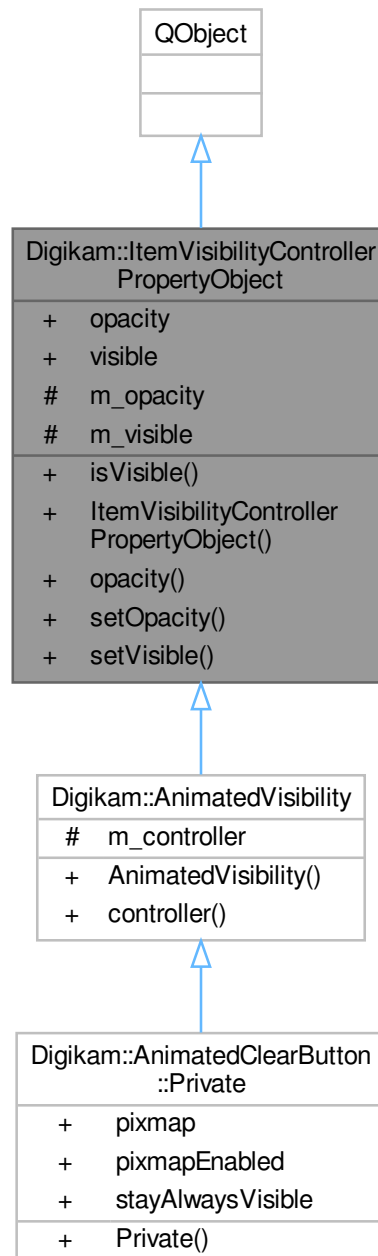
### 6.981.2.15 visibleItems()

```
QList< QObject * > Digikam::ItemVisibilityController::visibleItems (
    IncludeFadingOutMode mode = IncludeFadingOut ) const
```

Returns all currently visible items.

## 6.982 Digikam::ItemVisibilityControllerPropertyObject Class Reference

Inheritance diagram for Digikam::ItemVisibilityControllerPropertyObject:



### Signals

- void **opacityChanged** ()
- void **visibleChanged** ()

### Public Member Functions

- bool **isVisible** () const
- [ItemVisibilityControllerPropertyObject](#) (QObject \*const parent=nullptr)
- qreal **opacity** () const
- void **setOpacity** (qreal opacity)
- void **setVisible** (bool visible)

### Protected Attributes

- qreal **m\_opacity** = 0.0
- bool **m\_visible** = false

### Properties

- qreal **opacity**
- bool **visible**

## 6.982.1 Constructor & Destructor Documentation

### 6.982.1.1 ItemVisibilityControllerPropertyObject()

```
Digikam::ItemVisibilityControllerPropertyObject::ItemVisibilityControllerPropertyObject (
    QObject *const parent = nullptr ) [explicit]
```

You can use this object as a container providing the properties set by [ItemVisibilityController](#). Connect to the signals accordingly, e.g. to trigger a repaint.

## 6.983 Digikam::JPEGUtils::digikam\_source\_mgr Struct Reference

### Public Attributes

- JOCTET **eoI** [2]
- struct jpeg\_source\_mgr **pub**

## 6.984 Digikam::JPEGUtils::JpegRotator Class Reference

### Public Member Functions

- bool [autoExifTransform](#) ()
- bool [exifTransform](#) (const [MetaEngineRotation](#) &matrix)
- bool [exifTransform](#) ([TransformAction](#) action)
- [JpegRotator](#) (const QString &file)
- void [setCurrentOrientation](#) ([MetaEngine::ImageOrientation](#) orientation)
- void [setDestinationFile](#) (const QString &dest)
- void [setDocumentName](#) (const QString &documentName)
- [~JpegRotator](#) ()

## Protected Member Functions

- bool **performJpegTransform** ([TransformAction](#) action, const QString &src, const QString &dest)
- void **updateMetadata** (const QString &fileName, const [MetaEngineRotation](#) &matrix)

## Protected Attributes

- QString **m\_destFile**
- QString **m\_documentName**
- QString **m\_file**
- [DMetadata](#) \* **m\_metadata** = nullptr
- [MetaEngine::ImageOrientation](#) **m\_orientation** = MetaEngine::ORIENTATION\_UNSPECIFIED
- QSize **m\_originalSize**

## 6.984.1 Constructor & Destructor Documentation

### 6.984.1.1 JpegRotator()

```
Digikam::JPEGUtils::JpegRotator::JpegRotator (
    const QString & file ) [explicit]
```

Create a [JpegRotator](#) reading from the given file. Per default, it will replace the file, read the current orientation from the metadata, and use the src file name as documentName.

### 6.984.1.2 ~JpegRotator()

```
Digikam::JPEGUtils::JpegRotator::~~JpegRotator ( )
```

Destructor

## 6.984.2 Member Function Documentation

### 6.984.2.1 autoExifTransform()

```
bool Digikam::JPEGUtils::JpegRotator::autoExifTransform ( )
```

Rotate the JPEG file's content according to the current orientation, resetting the current orientation to normal. The final result of loading the image does not change.

### 6.984.2.2 exifTransform() [1/2]

```
bool Digikam::JPEGUtils::JpegRotator::exifTransform (
    const MetaEngineRotation & matrix )
```

Rotate the given image by the given [Matrix](#). The matrix describes the final transformation, it is not adjusted by current rotation.

### 6.984.2.3 exifTransform() [2/2]

```
bool Digikam::JPEGUtils::JpegRotator::exifTransform (
    TransformAction action )
```

Rotate the given image by the given TransformAction. The current orientation will be taken into account

### 6.984.2.4 setCurrentOrientation()

```
void Digikam::JPEGUtils::JpegRotator::setCurrentOrientation (
    MetaEngine::ImageOrientation orientation )
```

Per default, the orientation is read from the metadata of the file. You can override this value

### 6.984.2.5 setDestinationFile()

```
void Digikam::JPEGUtils::JpegRotator::setDestinationFile (
    const QString & dest )
```

Set the destination file. By default, the source file will be overwritten by atomic operation if the operation had succeeded.

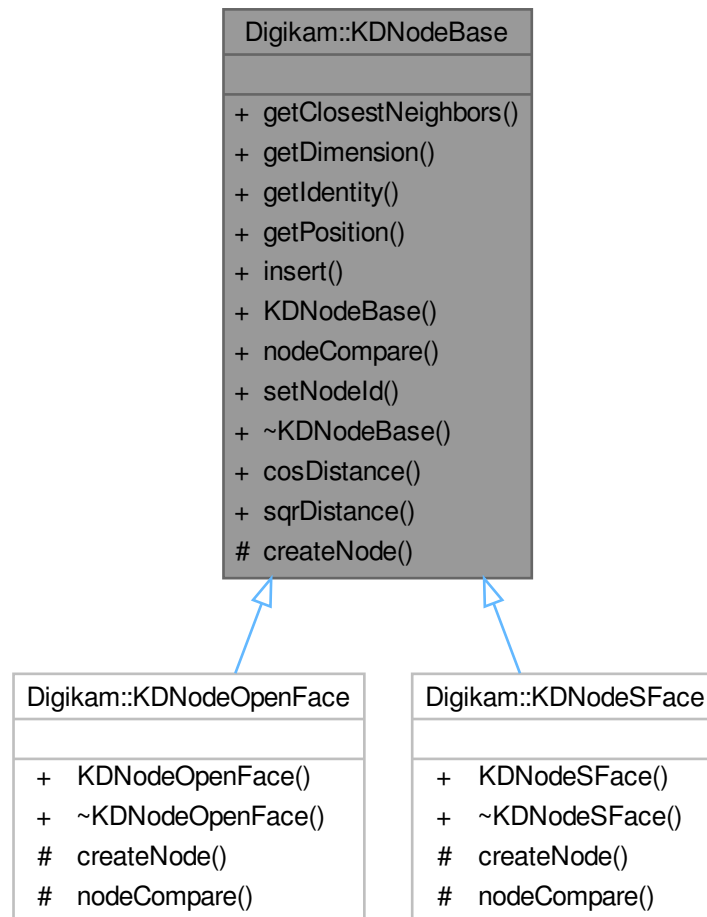
### 6.984.2.6 setDocumentName()

```
void Digikam::JPEGUtils::JpegRotator::setDocumentName (
    const QString & documentName )
```

Set the Exif document name of the destination file. Default value is the source's file name

## 6.985 Digikam::KNodeBase Class Reference

Inheritance diagram for Digikam::KNodeBase:



### Classes

- struct [NodeCompareResult](#)

### Public Member Functions

- double [getClosestNeighbors](#) (QMap< double, QVector< int > > &neighborList, const cv::Mat &position, float sqRange, float cosThreshold, int maxNbNeighbors) const
- int [getDimension](#) ()
- int [getIdentity](#) ()
- cv::Mat [getPosition](#) () const
- [KNodeBase](#) \* [insert](#) (const cv::Mat &nodePos, const int identity)
- [KNodeBase](#) (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension)
- virtual [NodeCompareResult](#) [nodeCompare](#) (const cv::Mat &queryPosition, const cv::Mat &currentPosition, float sqRange, float cosThreshold, int nbDimension) const =0
- void [setNodeId](#) (int id)



### Static Public Member Functions

- static float **cosDistance** (const float \*const pos1, const float \*const pos2, int dimension)
- static float **sqrDistance** (const float \*const pos1, const float \*const pos2, int dimension)

### Protected Member Functions

- virtual [KNodeBase](#) \* **createNode** (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension)=0

## 6.985.1 Member Function Documentation

### 6.985.1.1 createNode()

```
virtual KNodeBase * Digikam::KNodeBase::createNode (
    const cv::Mat & nodePos,
    const int identity,
    int splitAxis,
    int dimension ) [protected], [pure virtual]
```

Pure virtual functions to be overridden in child classes.

Implemented in [Digikam::KNodeOpenFace](#), and [Digikam::KNodeSFace](#).

### 6.985.1.2 getClosestNeighbors()

```
double Digikam::KNodeBase::getClosestNeighbors (
    QMap< double, QVector< int > > & neighborList,
    const cv::Mat & position,
    float sqRange,
    float cosThreshold,
    int maxNbNeighbors ) const
```

Return a list of closest neighbors, limited by maxNbNeighbors and sqRange.

### 6.985.1.3 getIdentity()

```
int Digikam::KNodeBase::getIdentity ( )
```

Return identity of the node.

### 6.985.1.4 getPosition()

```
cv::Mat Digikam::KNodeBase::getPosition ( ) const
```

Return position vector of a node.

### 6.985.1.5 insert()

```
KDNodeBase * Digikam::KDNodeBase::insert (
    const cv::Mat & nodePos,
    const int identity )
```

Insert a new node to the sub-tree.

### 6.985.1.6 setNodeId()

```
void Digikam::KDNodeBase::setNodeId (
    int id )
```

Set database entry ID of the node.

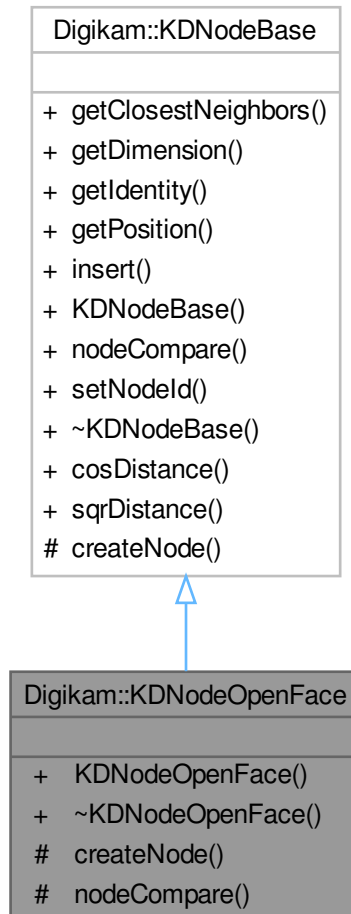
## 6.986 Digikam::KDNodeBase::NodeCompareResult Struct Reference

### Public Attributes

- double **distance1**
- double **distance2**
- bool **result**

## 6.987 Digikam::KNodeOpenFace Class Reference

Inheritance diagram for Digikam::KNodeOpenFace:



### Public Member Functions

- **KNodeOpenFace** (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension)

### Public Member Functions inherited from [Digikam::KNodeBase](#)

- double [getClosestNeighbors](#) (QMap< double, QVector< int > > &neighborList, const cv::Mat &position, float sqRange, float cosThreshold, int maxNbNeighbors) const
- int [getDimension](#) ()
- int [getIdentity](#) ()
- cv::Mat [getPosition](#) () const
- [KNodeBase](#) \* [insert](#) (const cv::Mat &nodePos, const int identity)
- [KNodeBase](#) (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension)
- void [setNodeId](#) (int id)

### Protected Member Functions

- [KNodeBase](#) \* [createNode](#) (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension) override
- [KNodeBase::NodeCompareResult](#) [nodeCompare](#) (const cv::Mat &queryPosition, const cv::Mat &currentPosition, float sqRange, float cosThreshold, int nbDimension) const override

### Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::KNodeBase](#)

- static float **cosDistance** (const float \*const pos1, const float \*const pos2, int dimension)
- static float **sqrDistance** (const float \*const pos1, const float \*const pos2, int dimension)

## 6.987.1 Member Function Documentation

### 6.987.1.1 createNode()

```
KNodeBase * Digikam::KNodeOpenFace::createNode (
    const cv::Mat & nodePos,
    const int identity,
    int splitAxis,
    int dimension ) [override], [protected], [virtual]
```

Pure virtual functions to be overridden in child classes.

Implements [Digikam::KNodeBase](#).

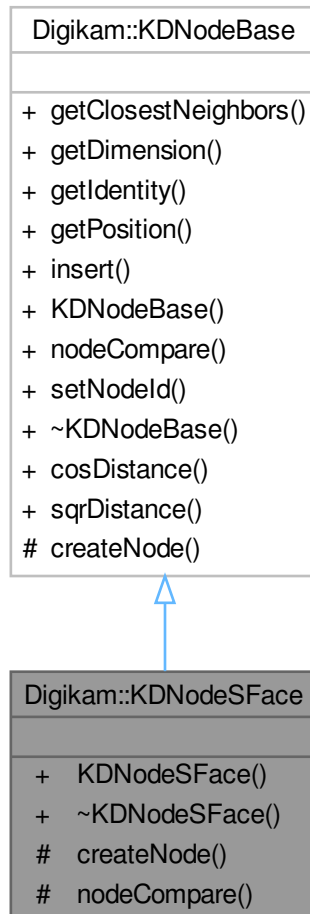
### 6.987.1.2 nodeCompare()

```
KNodeBase::NodeCompareResult Digikam::KNodeOpenFace::nodeCompare (
    const cv::Mat & queryPosition,
    const cv::Mat & currentPosition,
    float sqRange,
    float cosThreshold,
    int nbDimension ) const [override], [protected], [virtual]
```

Implements [Digikam::KNodeBase](#).

## 6.988 Digikam::KNodeSFace Class Reference

Inheritance diagram for Digikam::KNodeSFace:



### Public Member Functions

- **KNodeSFace** (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension)

### Public Member Functions inherited from [Digikam::KNodeBase](#)

- double [getClosestNeighbors](#) (QMap< double, QVector< int > > &neighborList, const cv::Mat &position, float sqRange, float cosThreshold, int maxNbNeighbors) const
- int [getDimension](#) ()
- int [getIdentity](#) ()
- cv::Mat [getPosition](#) () const
- [KNodeBase](#) \* [insert](#) (const cv::Mat &nodePos, const int identity)
- [KNodeBase](#) (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension)
- void [setNodeId](#) (int id)

### Protected Member Functions

- [KNodeBase](#) \* [createNode](#) (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension) override
- [KNodeBase::NodeCompareResult](#) [nodeCompare](#) (const cv::Mat &queryPosition, const cv::Mat &currentPosition, float sqRange, float cosThreshold, int nbDimension) const override

### Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::KNodeBase](#)

- static float **cosDistance** (const float \*const pos1, const float \*const pos2, int dimension)
- static float **sqrDistance** (const float \*const pos1, const float \*const pos2, int dimension)

## 6.988.1 Member Function Documentation

### 6.988.1.1 createNode()

```
KNodeBase * Digikam::KNodeSFace::createNode (
    const cv::Mat & nodePos,
    const int identity,
    int splitAxis,
    int dimension ) [override], [protected], [virtual]
```

Pure virtual functions to be overridden in child classes.

Implements [Digikam::KNodeBase](#).

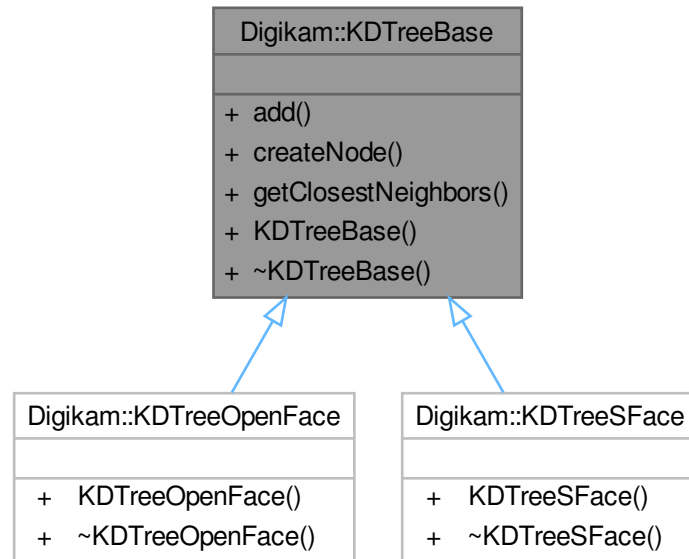
### 6.988.1.2 nodeCompare()

```
KNodeBase::NodeCompareResult Digikam::KNodeSFace::nodeCompare (
    const cv::Mat & queryPosition,
    const cv::Mat & currentPosition,
    float sqRange,
    float cosThreshold,
    int nbDimension ) const [override], [protected], [virtual]
```

Implements [Digikam::KNodeBase](#).

## 6.989 Digikam::KDTreeBase Class Reference

Inheritance diagram for Digikam::KDTreeBase:



### Public Member Functions

- virtual [KDNodeBase](#) \* [add](#) (const cv::Mat &position, const int identity)  
*add new node to KD-Tree*
- virtual [KDNodeBase](#) \* [createNode](#) (const cv::Mat &nodePos, const int identity, int splitAxis, int dimension)=0  
*create an ew node*
- virtual QMap< double, QVector< int > > [getClosestNeighbors](#) (const cv::Mat &position, float sqRange, int maxNbNeighbors) const
- [KDTreeBase](#) (int dim, int kdtreeThreshold=KDTREE\_MAP\_THRESHOLD)  
*Constructor of the class implementing the KD-Tree for vector space partitioning.*

### 6.989.1 Constructor & Destructor Documentation

#### 6.989.1.1 KDTreeBase()

```

Digikam::KDTreeBase::KDTreeBase (
    int dim,
    int kdtreeThreshold = KDTREE_MAP_THRESHOLD ) [explicit]
  
```

#### Parameters

<i>dim</i>	The dimension of the tree.
<i>kdtreeThreshold</i>	The KD-Tree threshold. Above this value, we start using the KD-Tree instead of the vector. If the vector grows to default KDTREE_MAP_THRESHOLD items, start using the KDTree.

**Note**

Due to sparse data density in the tree, we initially use a vector of nodes to compare the target to the samples once we have achieved a suitable data density we delete the vector (but not the nodes) and begin using the tree.

Using this to compare brute force vs kdtree performance due to sparse data in k-dimensions (128 dimensions for face features).

**6.989.2 Member Function Documentation****6.989.2.1 add()**

```
KDNodeBase * Digikam::KDTreeBase::add (
    const cv::Mat & position,
    const int identity ) [virtual]
```

**Parameters**

<i>position</i>	K-dimension vector
<i>identity</i>	identity of this face vector

**Returns**

the KD-Tree node base instance

**6.989.2.2 createNode()**

```
virtual KDNodeBase * Digikam::KDTreeBase::createNode (
    const cv::Mat & nodePos,
    const int identity,
    int splitAxis,
    int dimension ) [pure virtual]
```

**Parameters**

<i>nodePos</i>	extracted face vectors
<i>identity</i>	identity of this face vector
<i>splitAxis</i>	current axis/dimension of the vector
<i>dimension</i>	number of dimensions (usually 128)

**Returns**

the KD-Tree node base instance

**6.989.2.3 getClosestNeighbors()**

```
QMap< double, QVector< int > > Digikam::KDTreeBase::getClosestNeighbors (
    const cv::Mat & position,
```



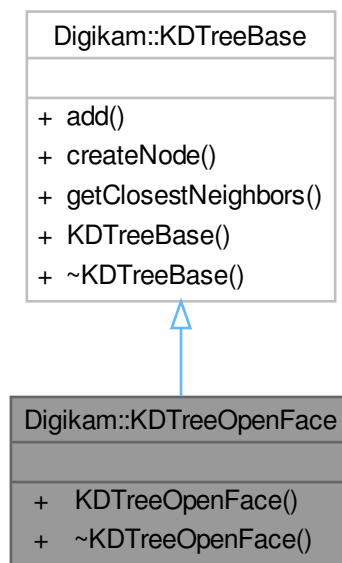
```
float sqRange,
int maxNbNeighbors ) const [virtual]
```

**Returns**

Map of N-nearest neighbors, sorted by distance

**6.990 Digikam::KDTreeOpenFace Class Reference**

Inheritance diagram for Digikam::KDTreeOpenFace:

**Public Member Functions**

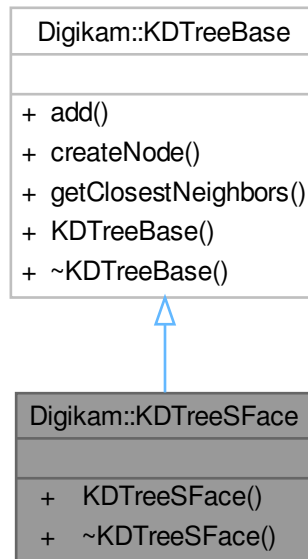
- `KDTreeOpenFace` (int dim, int threshold=KDTREE\_MAP\_THRESHOLD)

**Public Member Functions inherited from [Digikam::KDTreeBase](#)**

- virtual `KDNodeBase * add` (const cv::Mat &position, const int identity)  
*add new node to KD-Tree*
- virtual `QMap< double, QVector< int > > getClosestNeighbors` (const cv::Mat &position, float sqRange, int maxNbNeighbors) const
- `KDTreeBase` (int dim, int kdtreeThreshold=KDTREE\_MAP\_THRESHOLD)  
*Constructor of the class implementing the KD-Tree for vector space partitioning.*

## 6.991 Digikam::KDTreeSFace Class Reference

Inheritance diagram for Digikam::KDTreeSFace:



### Public Member Functions

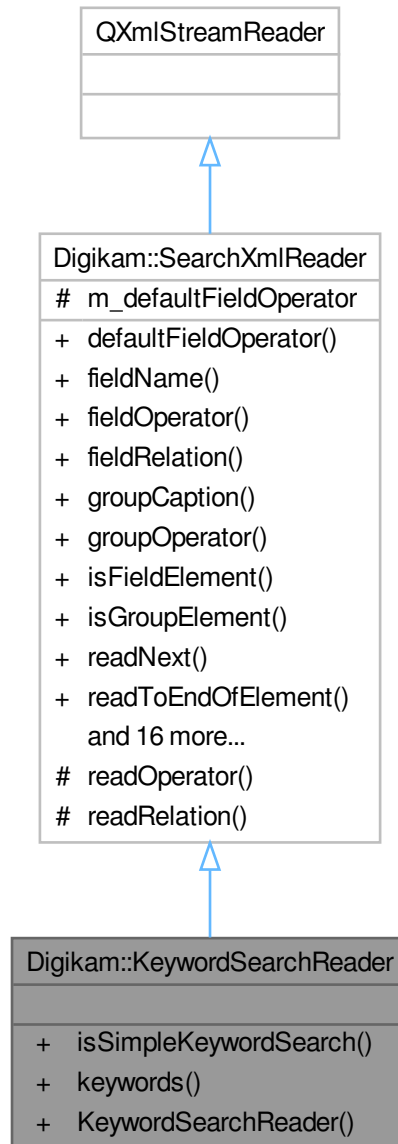
- **KDTreeSFace** (int dim, int threshold=KDTREE\_MAP\_THRESHOLD)

### Public Member Functions inherited from [Digikam::KDTreeBase](#)

- virtual [KDNodeBase](#) \* **add** (const cv::Mat &position, const int identity)  
*add new node to KD-Tree*
- virtual QMap< double, QVector< int > > **getClosestNeighbors** (const cv::Mat &position, float sqRange, int maxNbNeighbors) const
- [KDTreeBase](#) (int dim, int kdtreeThreshold=KDTREE\_MAP\_THRESHOLD)  
*Constructor of the class implementing the KD-Tree for vector space partitioning.*

## 6.992 Digikam::KeywordSearchReader Class Reference

Inheritance diagram for Digikam::KeywordSearchReader:



### Public Member Functions

- `bool isSimpleKeywordSearch ()`  
Checks if the XML is a simple keyword search, compatible with `keywords()`.
- `QStringList keywords ()`  
Returns the keywords from this search, merged in a list.
- `KeywordSearchReader (const QString &xml)`

## Public Member Functions inherited from [Digikam::SearchXmlReader](#)

- SearchXml::Operator [defaultFieldOperator](#) () const
- QString [fieldName](#) () const
- SearchXml::Operator [fieldOperator](#) () const
- SearchXml::Relation [fieldRelation](#) () const
- QString [groupCaption](#) () const
- SearchXml::Operator [groupOperator](#) () const
- bool [isFieldElement](#) () const
- bool [isGroupElement](#) () const
- SearchXml::Element [readNext](#) ()
- void [readToEndOfElement](#) ()
- void [readToFirstField](#) ()
- bool [readToStartOfElement](#) (const QString &name)
- **SearchXmlReader** (const QString &xml)
- QString [value](#) ()
- QDateTime [valueToDateTime](#) ()
- QList< QDateTime > [valueToDateTimeList](#) ()
- double [valueToDouble](#) ()
- QList< double > [valueToDoubleList](#) ()
- QList< double > [valueToDoubleOrDoubleList](#) ()
- int [valueToInt](#) ()
- QList< int > [valueToIntList](#) ()
- QList< int > [valueToIntOrIntList](#) ()
- qlonglong [valueToLongLong](#) ()
- QList< qlonglong > [valueToLongLongList](#) ()
- QStringList [valueToStringList](#) ()
- QList< QString > [valueToStringOrStringList](#) ()

### Additional Inherited Members

## Protected Member Functions inherited from [Digikam::SearchXmlReader](#)

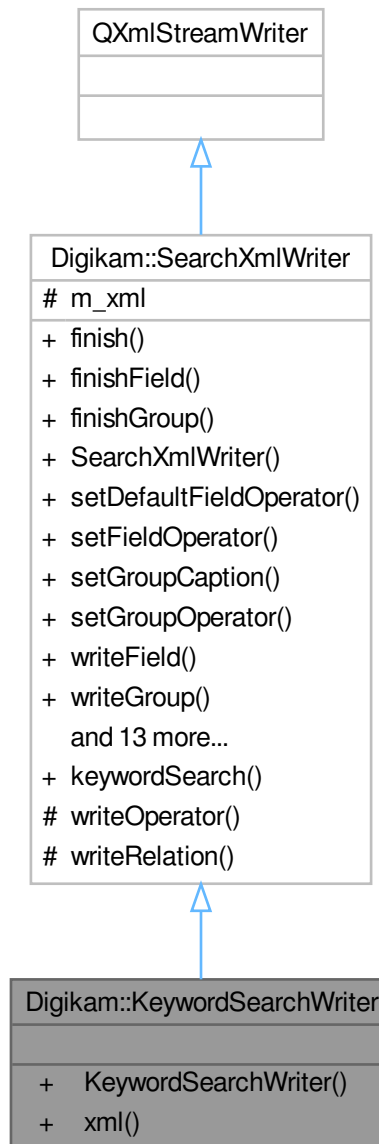
- SearchXml::Operator [readOperator](#) (const QString &, SearchXml::Operator) const
- SearchXml::Relation [readRelation](#) (const QString &, SearchXml::Relation) const

## Protected Attributes inherited from [Digikam::SearchXmlReader](#)

- SearchXml::Operator [m\\_defaultFieldOperator](#)

## 6.993 Digikam::KeywordSearchWriter Class Reference

Inheritance diagram for Digikam::KeywordSearchWriter:



### Public Member Functions

- QString `xml` (const QStringList &keywordList)

### Public Member Functions inherited from [Digikam::SearchXmlWriter](#)

- void `finish` ()

- void [finishField](#) ()
- void [finishGroup](#) ()
- [SearchXmlWriter](#) ()
- void [setDefaultFieldOperator](#) (SearchXml::Operator op)
- void [setFieldOperator](#) (SearchXml::Operator op)
- void [setGroupCaption](#) (const QString &caption)
- void [setGroupOperator](#) (SearchXml::Operator op)
- void [writeField](#) (const QString &name, SearchXml::Relation relation)
- void [writeGroup](#) ()
- void [writeValue](#) (const QDateTime &dateTime)
- void [writeValue](#) (const QList< double > &valueList, int precision=8)
- void [writeValue](#) (const QList< float > &valueList, int precision=6)
- void [writeValue](#) (const QList< int > &valueList)
- void [writeValue](#) (const QList< QDateTime > &valueList)
- void [writeValue](#) (const QList< qlonglong > &valueList)
- void [writeValue](#) (const QString &value)
- void [writeValue](#) (const QStringList &valueList)
- void [writeValue](#) (double value, int precision=8)
- void [writeValue](#) (float value, int precision=6)
- void [writeValue](#) (int value)
- void [writeValue](#) (qlonglong value)
- QString [xml](#) () const

#### Additional Inherited Members

#### Static Public Member Functions inherited from [Digikam::SearchXmlWriter](#)

- static QString [keywordSearch](#) (const QString &keyword)

#### Protected Member Functions inherited from [Digikam::SearchXmlWriter](#)

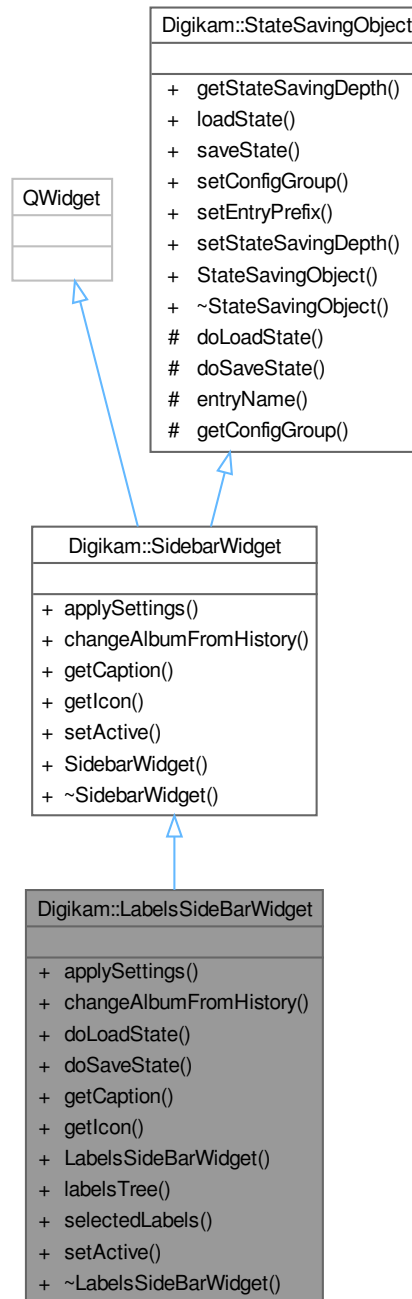
- void [writeOperator](#) (const QString &, SearchXml::Operator)
- void [writeRelation](#) (const QString &, SearchXml::Relation)

#### Protected Attributes inherited from [Digikam::SearchXmlWriter](#)

- QString [m\\_xml](#)

## 6.994 Digikam::LabelsSideBarWidget Class Reference

Inheritance diagram for Digikam::LabelsSideBarWidget:



### Public Member Functions

- void `applySettings()` override
- void `changeAlbumFromHistory` (const `QList< Album * >` &album) override

- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- **LabelsSideBarWidget** (QWidget \*const parent)
- [LabelsTreeView](#) \* **labelsTree** ()
- QHash< LabelsTreeView::Labels, QList< int > > **selectedLabels** ()
- void [setActive](#) (bool active) override

### Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

### Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.994.1 Member Function Documentation

### 6.994.1.1 [applySettings\(\)](#)

```
void Digikam::LabelsSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).



### 6.994.1.2 changeAlbumFromHistory()

```
void Digikam::LabelsSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.994.1.3 doLoadState()

```
void Digikam::LabelsSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.994.1.4 doSaveState()

```
void Digikam::LabelsSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.994.1.5 getCaption()

```
const QString Digikam::LabelsSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.994.1.6 getIcon()

```
const QIcon Digikam::LabelsSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.994.1.7 setActive()

```
void Digikam::LabelsSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

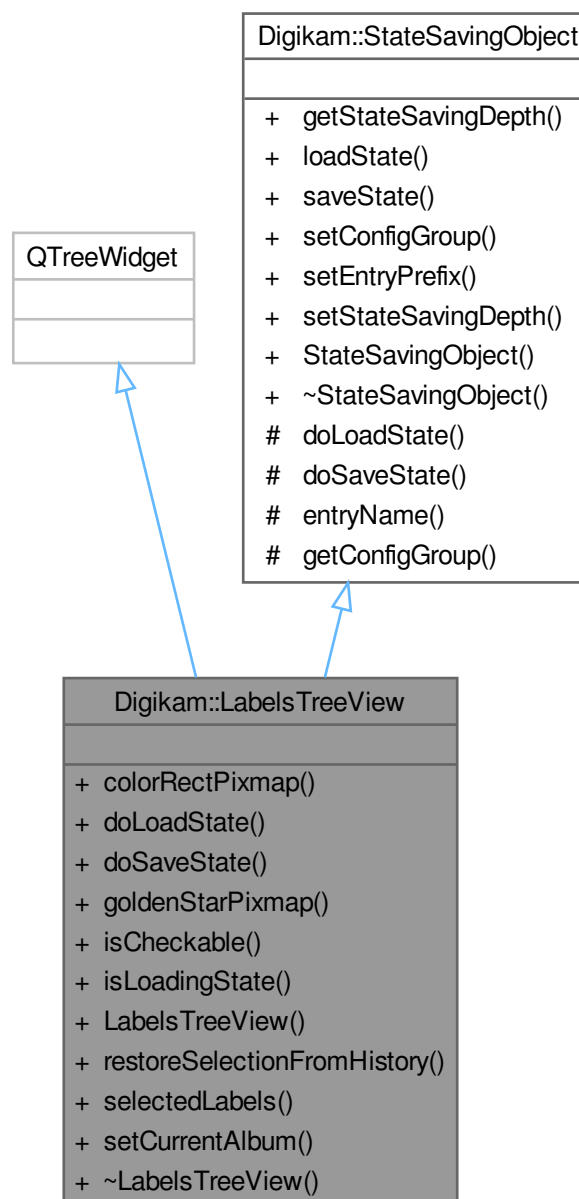
## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.995 Digikam::LabelsTreeView Class Reference

Inheritance diagram for Digikam::LabelsTreeView:



## Public Types

- enum **Labels** { **Ratings** = 0 , **Picks** , **Colors** }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Signals

- void **signalSetCurrentAlbum** ()

## Public Member Functions

- QPixmap [colorRectPixmap](#) (const QColor &color) const  
*Creates a 30\*30 rectangular pixmap with specific color.*
- void [doLoadState](#) () override  
*Loading and saving state function inherited from [StateSavingObject](#).*
- void [doSaveState](#) () override
- QPixmap [goldenStarPixmap](#) (bool fillin=true) const
- bool [isCheckable](#) () const
- bool [isLoadingState](#) () const
- **LabelsTreeView** (QWidget \*const parent=nullptr, bool setCheckable=false)
- void [restoreSelectionFromHistory](#) (QHash< Labels, QList< int > > neededLabels)  
*Restores the selection state from the [AlbumHistory](#) class.*
- QHash< Labels, QList< int > > [selectedLabels](#) ()  
*Provide the current selection from the tree-view hierarchy.*
- void **setCurrentAlbum** ()  
*Emits a signal to the search handler to set the Current album from currently selected labels.*

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.995.1 Member Function Documentation

### 6.995.1.1 [colorRectPixmap\(\)](#)

```
QPixmap Digikam::LabelsTreeView::colorRectPixmap (
    const QColor & color ) const
```

**Parameters**

<i>color</i>	wanted to be set
--------------	------------------

**Returns**

pixmap has a rectangle filled with the color

**6.995.1.2 doLoadState()**

```
void Digikam::LabelsTreeView::doLoadState ( ) [override], [virtual]
```

Implements [Digikam::StateSavingObject](#).

**6.995.1.3 doSaveState()**

```
void Digikam::LabelsTreeView::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

**6.995.1.4 goldenStarPixmap()**

```
QPixmap Digikam::LabelsTreeView::goldenStarPixmap (
    bool fillin = true ) const
```

**Returns**

a QPixmap of a 30\*30 pixels golden star used for rating and widget icon

**6.995.1.5 isCheckable()**

```
bool Digikam::LabelsTreeView::isCheckable ( ) const
```

**Returns**

true if the tree widget is checkable and false if not

**6.995.1.6 isLoadingState()**

```
bool Digikam::LabelsTreeView::isLoadingState ( ) const
```

**Returns**

true if Loading state function is running

**6.995.1.7 restoreSelectionFromHistory()**

```
void Digikam::LabelsTreeView::restoreSelectionFromHistory (
    QHash< Labels, QList< int > > neededLabels )
```

## Parameters

<code>neededLabels</code>	is a QHash to restore the selection from it, the hash is formatted just like the hash generated from
---------------------------	--

## See also

[selectedLabels\(\)](#)

## 6.995.1.8 selectedLabels()

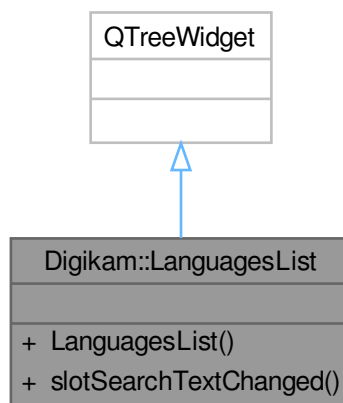
```
QHash< LabelsTreeView::Labels, QList< int > > Digikam::LabelsTreeView::selectedLabels ( )
```

## Returns

a QHash with three keys: "Ratings", "Picks", and "Colors", every key dedicated to an int list which holds the rows selected

## 6.996 Digikam::LanguagesList Class Reference

Inheritance diagram for Digikam::LanguagesList:



## Public Slots

- void **slotSearchTextChanged** (const [SearchTextSettings](#) &settings)

## Signals

- void **signalSearchResult** (int)

**Public Member Functions**

- [LanguagesList](#) (QWidget \*const parent)

## 6.997 Digikam::LcmsLock Class Reference

**Public Member Functions**

- [LcmsLock](#) ()

### 6.997.1 Constructor & Destructor Documentation

#### 6.997.1.1 LcmsLock()

Digikam::LcmsLock::LcmsLock ( )

Obtain an [LcmsLock](#) if you access not clearly thread-safe LittleCMS methods

## 6.998 Digikam::LensDistortionFilter Class Reference

Inheritance diagram for Digikam::LensDistortionFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **LensDistortionFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, double main=0.0, double edge=0.0, double rescale=0.0, double brighten=0.0, int center\_x=0, int center\_y=0)
- **LensDistortionFilter** (QObject \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }



## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.998.1 Member Function Documentation

### 6.998.1.1 filterAction()

```
FilterAction Digikam::LensDistortionFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.998.1.2 filterIdentifier()

```
QString Digikam::LensDistortionFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.998.1.3 readParameters()

```
void Digikam::LensDistortionFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.999 Digikam::LensDistortionPixelAccess Class Reference

### Public Member Functions

- **LensDistortionPixelAccess** ([DImg](#) \*srcImage)
- void **pixelAccessGetCubic** (double srcX, double srcY, double brighten, uchar \*dst)

### Protected Member Functions

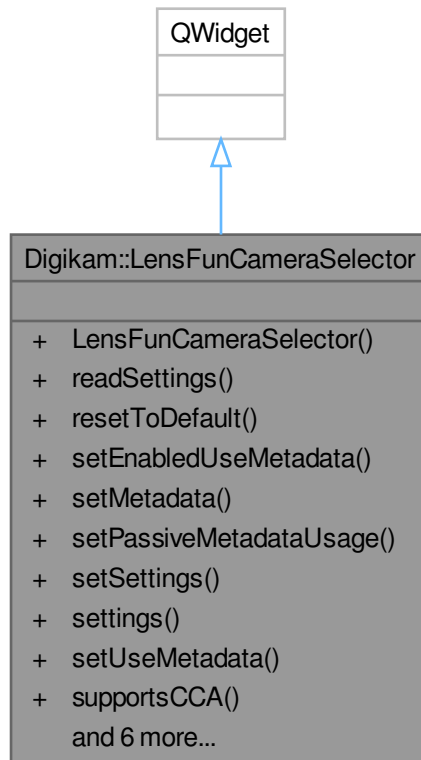
- void **cubicInterpolate** (uchar \*src, int rowStride, uchar \*dst, bool sixteenBit, double dx, double dy, double brighten)
- uchar \* **pixelAccessAddress** (int i, int j)
- void **pixelAccessDoEdge** (int i, int j)
- void **pixelAccessReposition** (int xInt, int yInt)
- void **pixelAccessSelectRegion** (int n)

### 6.999.1 Detailed Description

[LensDistortionPixelAccess](#) class: solving the eternal problem: random, cubic-interpolated, sub-pixel coordinate access to an image. Assuming that accesses are at least slightly coherent, [LensDistortionPixelAccess](#) keeps [LensDistortionPixelAccessRegions](#) buffers, each containing a [LensDistortionPixelAccessWidth](#) x [LensDistortionPixelAccessHeight](#) region of pixels. Buffer[0] is always checked first, so move the last accessed region into that position. When a request arrives which is outside all the regions, get a new region. The new region is placed so that the requested pixel is positioned at [[LensDistortionPixelAccessXOffset](#), [LensDistortionPixelAccessYOffset](#)] in the region.

## 6.1000 Digikam::LensFunCameraSelector Class Reference

Inheritance diagram for Digikam::LensFunCameraSelector:



### Public Types

- typedef `QMap< QString, QString >` **Device**

### Signals

- void **signalLensSettingsChanged** ()

### Public Member Functions

- **LensFunCameraSelector** (`QWidget *const parent=nullptr`)
- void **readSettings** (`const KConfigGroup &group`)
- void **resetToDefault** ()
- void **setEnabledUseMetadata** (`bool b`)
- void **setMetadata** (`const MetaEngineData &`)
- void **setPassiveMetadataUsage** (`bool b`)
- void **setSettings** (`const LensFunContainer &settings`)

- [LensFunContainer settings](#) ()
- void **setUseMetadata** (bool b)
- bool **supportsCCA** () const
- bool **supportsDistortion** () const
- bool **supportsGeometry** () const
- bool **supportsVig** () const
- bool **useMetadata** () const
- void **writeSettings** (KConfigGroup &group)

## 6.1000.1 Member Function Documentation

### 6.1000.1.1 setPassiveMetadataUsage()

```
void Digikam::LensFunCameraSelector::setPassiveMetadataUsage (
    bool b )
```

Special mode used with BQM which processes multiple items at the same time.

## 6.1001 Digikam::LensFunContainer Class Reference

### Public Attributes

- double **aperture** = -1.0
- QString **cameraMake**
- QString **cameraModel**
- double **cropFactor** = -1.0
- bool **filterCCA** = true  
*Chromatic Aberration Corrections.*
- bool **filterDST** = true  
*Distortion Corrections.*
- bool **filterGEO** = true  
*Geometry Corrections.*
- bool **filterVIG** = true  
*Vignetting Corrections.*
- double **focalLength** = -1.0
- QString **lensModel**
- double **subjectDistance** = -1.0

## 6.1002 Digikam::LensFunFilter Class Reference

Inheritance diagram for Digikam::LensFunFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **LensFunFilter** (*DImg* \*const orgImage, *QObject* \*const parent, const [LensFunContainer](#) &settings)
- **LensFunFilter** (*QObject* \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override
- bool [registerSettingsToXmp](#) ([MetaEngineData](#) &data) const

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) (*DImg* \*const orgImage, *QObject* \*const parent, const *QString* &name=*QString*())
- [DImgThreadedFilter](#) (*QObject* \*const parent=nullptr, const *QString* &name=*QString*())
- const *QString* & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- *QList*< int > [multithreadedSteps](#) (int [stop](#), int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual *QString* [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const *QString* &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual *QList*< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (*QObject* \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- *QThread::Priority* [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (*QThread::Priority* priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static *QString* [DisplayableName](#) ()
- static *QString* [FilterIdentifier](#) ()
- static *QList*< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.1002.1 Member Function Documentation

### 6.1002.1.1 filterAction()

`FilterAction` Digikam::LensFunFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1002.1.2 filterIdentifier()

`QString` Digikam::LensFunFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1002.1.3 readParameters()

```
void Digikam::LensFunFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1003 Digikam::LensFunface Class Reference

### Public Types

- typedef const IfCamera \* **DevicePtr**
- typedef QList< LensPtr > **LensList**
- typedef const IfLens \* **LensPtr**
- enum **MetadataMatch** { **MetadataUnavailable** = -2 , **MetadataNoMatch** = -1 , **MetadataPartialMatch** = 0 , **MetadataExactMatch** = 1 }

### Public Member Functions

- DevicePtr **findCamera** (const QString &make, const QString &model) const
- MetadataMatch **findFromMetadata** (const [DMetadata](#) \*const meta)
- LensPtr **findLens** (const QString &model) const
- QString **lensDescription** ( ) const
- const IfCamera \*const \* **lensFunCameras** ( ) const
- IfDatabase \* **lensFunDataBase** ( ) const
- QString **makeDescription** ( ) const
- QString **modelDescription** ( ) const
- void **setFilterSettings** (const [LensFunContainer](#) &other)
- void **setSettings** (const [LensFunContainer](#) &other)
- [LensFunContainer](#) **settings** ( ) const
- void **setUsedCamera** (DevicePtr cam)
- void **setUsedLens** (LensPtr lens)
- bool **supportsCCA** ( ) const
- bool **supportsDistortion** ( ) const
- bool **supportsGeometry** ( ) const
- bool **supportsVig** ( ) const
- DevicePtr **usedCamera** ( ) const
- LensPtr **usedLens** ( ) const



### Static Public Member Functions

- static QString **lensFunVersion** ()

## 6.1003.1 Member Function Documentation

### 6.1003.1.1 lensDescription()

```
QString Digikam::LensFunIface::lensDescription ( ) const
```

Return Lens string description found in metadata

### 6.1003.1.2 makeDescription()

```
QString Digikam::LensFunIface::makeDescription ( ) const
```

Return Camera maker string description found in metadata

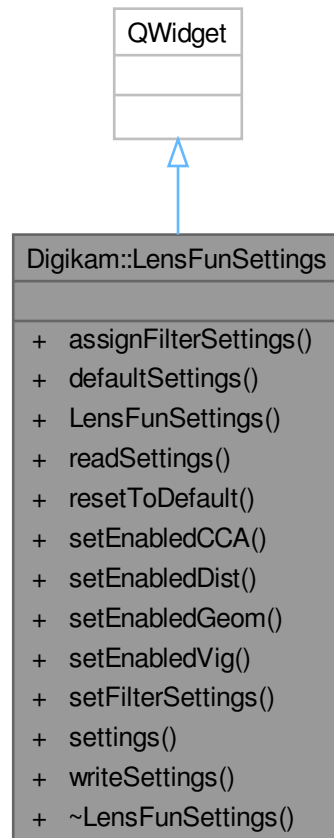
### 6.1003.1.3 modelDescription()

```
QString Digikam::LensFunIface::modelDescription ( ) const
```

Return Camera model string description found in metadata

## 6.1004 Digikam::LensFunSettings Class Reference

Inheritance diagram for Digikam::LensFunSettings:



### Signals

- void `signalSettingsChanged ()`

### Public Member Functions

- void `assignFilterSettings (LensFunContainer &prm)`
- `LensFunContainer defaultSettings ()` const
- `LensFunSettings (QWidget *const parent=nullptr)`
- void `readSettings (const KConfigGroup &group)`
- void `resetToDefault ()`
- void `setEnabledCCA (bool b)`
- void `setEnabledDist (bool b)`
- void `setEnabledGeom (bool b)`
- void `setEnabledVig (bool b)`
- void `setFilterSettings (const LensFunContainer &settings)`
- `LensFunContainer settings ()` const
- void `writeSettings (KConfigGroup &group)`

## 6.1005 Digikam::LessThanByProximityToSubject Class Reference

### Public Member Functions

- **LessThanByProximityToSubject** (const [ItemInfo](#) &subject)
- **bool operator()** (const [ItemInfo](#) &a, const [ItemInfo](#) &b)

### Public Attributes

- [ItemInfo](#) **subject**

## 6.1006 Digikam::LevelsContainer Class Reference

### Public Attributes

- double **gamma** [5] = { 1.0 }
- int **hInput** [5] = { 65535 }
- int **hOutput** [5] = { 65535 }
- int **lInput** [5] = { 0 }
- int **lOutput** [5] = { 0 }

## 6.1007 Digikam::LevelsFilter Class Reference

Inheritance diagram for Digikam::LevelsFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **LevelsFilter** (const [LevelsContainer](#) &settings, [DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, [DImg](#) &destImage, int progressBegin=0, int progressEnd=100)
- **LevelsFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [LevelsContainer](#) &settings=[LevelsContainer](#)())
- **LevelsFilter** (QObject \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int [CurrentVersion](#) ()
- static QString [DisplayableName](#) ()
- static QString [FilterIdentifier](#) ()
- static QList< int > [SupportedVersions](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.1007.1 Member Function Documentation

### 6.1007.1.1 filterAction()

`FilterAction` Digikam::LevelsFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1007.1.2 filterIdentifier()

`QString` Digikam::LevelsFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

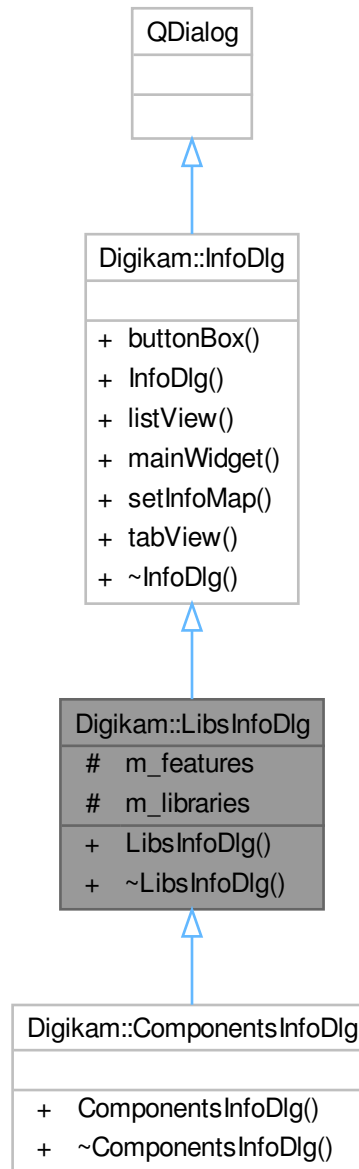
### 6.1007.1.3 readParameters()

```
void Digikam::LevelsFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1008 Digikam::LibsInfoDlg Class Reference

Inheritance diagram for Digikam::LibsInfoDlg:



### Public Member Functions

- [LibsInfoDlg](#) (`QWidget *const parent`)

### Public Member Functions inherited from [Digikam::InfoDlg](#)

- `QDialogButtonBox * buttonBox () const`



- **InfoDlg** (QWidget \*const parent)
- QTreeWidget \* **listView** () const
- QWidget \* **mainWidget** () const
- virtual void **setInfoMap** (const QMap< QString, QString > &list)
- QTabWidget \* **tabView** () const

### Protected Attributes

- QTreeWidgetItem \* **m\_features** = nullptr
- QTreeWidgetItem \* **m\_libraries** = nullptr

## 6.1008.1 Constructor & Destructor Documentation

### 6.1008.1.1 LibsInfoDlg()

```
Digikam::LibsInfoDlg::LibsInfoDlg (  
    QWidget *const parent ) [explicit]
```

NOTE: MANIFEST.txt is a text file generated with the bundles and listing all git revisions of rolling release components. One section title start with '+'. All component revisions are listed below line by line with the name and the revision separated by ':'. More than one section can be listed in manifest.

## 6.1009 Digikam::LightTablePreview Class Reference

Inheritance diagram for Digikam::LightTablePreview:



### Signals

- void **signalDroppedItems** (const [ItemInfoList](#) &)

### Signals inherited from [Digikam::ItemPreviewView](#)

- void **signalAddToExistingQueue** (int)
- void **signalDeleteItem** ()
- void **signalEscapePreview** ()
- void **signalGotoAlbumAndItem** (const [ItemInfo](#) &)
- void **signalGotoDateAndItem** (const [ItemInfo](#) &)
- void **signalGotoTagAndItem** (int)
- void **signalNextItem** ()
- void **signalPopupTagsView** ()
- void **signalPreviewLoaded** (bool success)
- void **signalPrevItem** ()
- void **signalSlideShowCurrent** ()

### Signals inherited from [Digikam::GraphicsDImgView](#)

- void **activated** ()
- void **contentsMoved** (bool panningFinished)
- void **contentsMoving** (int, int)
- void **leftButtonClicked** ()
- void **leftButtonDoubleClicked** ()
- void **resized** ()
- void **rightButtonClicked** ()
- void **toNextImage** ()
- void **toPreviousImage** ()
- void **viewportRectChanged** (const QRectF &viewportRect)

### Public Member Functions

- **LightTablePreview** (QWidget \*const parent=nullptr)
- void **setDragAndDropEnabled** (bool b)
- void **showDragAndDropMessage** ()

### Public Member Functions inherited from [Digikam::ItemPreviewView](#)

- [ItemInfo](#) **getItemInfo** () const
- **ItemPreviewView** (QWidget \*const parent, Mode mode=IconViewPreview, [Album](#) \*const currAlbum=nullptr)
- void **reload** ()
- void **setImagePath** (const QString &path=QString())
- void **setItemInfo** (const [ItemInfo](#) &info=[ItemInfo](#)(), const [ItemInfo](#) &previous=[ItemInfo](#)(), const [ItemInfo](#) &next=[ItemInfo](#)())
- void **setPreviousNextPaths** (const QString &previous, const QString &next)

### Public Member Functions inherited from [Digikam::GraphicsDImgView](#)

- int **contentsX** () const
- int **contentsY** () const
- void **drawText** (QPainter \*p, const QRectF &rect, const QString &text)
- void **fitToWindow** ()
- **GraphicsDImgView** (QWidget \*const parent=nullptr)
- [GraphicsDImgItem](#) \* **item** () const
- [SinglePhotoPreviewLayout](#) \* **layout** () const
- [DImgPreviewItem](#) \* **previewItem** () const
- void **scrollPointOnPoint** (const QPointF &scenePos, const QPoint &viewportPos)
- void **setContentPos** (int x, int y)
- void **setItem** ([GraphicsDImgItem](#) \*const item)
- void **toggleFullScreen** (bool set)
- QRect **visibleArea** () const

### Additional Inherited Members

### Public Types inherited from [Digikam::ItemPreviewView](#)

- enum **Mode** { [IconViewPreview](#) , [LightTablePreview](#) }

### Protected Slots inherited from [Digikam::GraphicsDImgView](#)

- void **slotContentsMoved** ()
- void **slotCornerButtonPressed** ()
- void **slotPanIconHidden** ()
- virtual void **slotPanIconSelectionMoved** (const QRect &, bool)

### Protected Member Functions inherited from [Digikam::ItemPreviewView](#)

- bool **acceptsMouseClicked** (QMouseEvent \*e) override
- void **dragEnterEvent** (QDragEnterEvent \*e) override
- void **dragMoveEvent** (QDragMoveEvent \*e) override
- void **dropEvent** (QDropEvent \*e) override
- void **enterEvent** (QEnterEvent \*) override
- void **leaveEvent** (QEvent \*e) override
- void **mousePressEvent** (QMouseEvent \*e) override
- void **showEvent** (QShowEvent \*e) override

### Protected Member Functions inherited from [Digikam::GraphicsDImgView](#)

- void **continuePanning** (const QPoint &pos)
- void **drawForeground** (QPainter \*painter, const QRectF &rect) override
- void **finishPanning** ()
- void **installPanIcon** ()
- void **mouseDoubleClickEvent** (QMouseEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **resizeEvent** (QResizeEvent \*) override
- void **scrollContentsBy** (int dx, int dy) override
- void **setScaleFitToWindow** (bool value)
- void **setShowText** (bool value)
- void **startPanning** (const QPoint &pos)
- void **wheelEvent** (QWheelEvent \*) override



- void **slotAssignRating** (int)
- void **slotColorLabelChanged** (const QUrl &, int)
- void **slotDockLocationChanged** (Qt::DockWidgetArea area)
- void **slotPickLabelChanged** (const QUrl &, int)
- void **slotRatingChanged** (const QUrl &, int)
- void **slotToggleTag** (const QUrl &, int)

### Public Slots inherited from [Digikam::ItemThumbnailBar](#)

- void **assignRating** (const QList< QModelIndex > &index, int rating)
- void **slotDockLocationChanged** (Qt::DockWidgetArea area)

### Public Slots inherited from [Digikam::ItemCategorizedView](#)

- void **hintAt** (const [ItemInfo](#) &info)
- void **openAlbum** (const QList< [Album](#) \* > &album)
- void **setCurrentInfo** (const [ItemInfo](#) &info)
- void **setCurrentUrl** (const QUrl &url)
- void **setCurrentUrlWhenAvailable** (const QUrl &url)
- void **setCurrentWhenAvailable** (qulonglong imageld)
- void **setSelectedItemInfos** (const QList< [ItemInfo](#) > &infos)
- void **setSelectedUrls** (const QList< QUrl > &urlList)
- void **setThumbnailSize** (int size)

### Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void **copy** () override
- void **cut** () override
- void **hideIndexNotification** ()
- void **paste** () override
- void **showIndexNotification** (const QModelIndex &index, const QString &message)

### Public Slots inherited from [Digikam::DCategorizedView](#)

- void **reset** () override

### Signals

- void **signalClearAll** ()
- void **signalContentChanged** ()
- void **signalDroppedItems** (const QList< [ItemInfo](#) > &)
- void **signalEditItem** (const [ItemInfo](#) &)
- void **signalRemoveItem** (const [ItemInfo](#) &)
- void **signalSetItemOnLeftPanel** (const [ItemInfo](#) &)
- void **signalSetItemOnRightPanel** (const [ItemInfo](#) &)

## Signals inherited from [Digikam::ItemCategorizedView](#)

- void **currentChanged** (const [ItemInfo](#) &info)
- void **deselected** (const QList< [ItemInfo](#) > &nowDeselectedInfos)
 

*Emitted when items are deselected. There may be other selected infos left. This signal is not emitted when the model is reset; then only selectionCleared is emitted.*
- void **imageActivated** (const [ItemInfo](#) &info)
 

*Emitted when the given image is activated. Info is never null.*
- void **modelChanged** ()
 

*Emitted when a new model is set.*
- void **selected** (const QList< [ItemInfo](#) > &newSelectedInfos)
 

*Emitted when new items are selected. The parameter includes only the newly selected infos, there may be other already selected infos.*

## Signals inherited from [Digikam::ItemViewCategorized](#)

- void **clicked** (const QMouseEvent \*e, const QModelIndex &index)
- void **entered** (const QMouseEvent \*e, const QModelIndex &index)
- void **keyPressed** (QKeyEvent \*e)
- void **selectionChanged** ()
- void **selectionCleared** ()
- void **viewportClicked** (const QMouseEvent \*e)
- void **zoomInStep** ()
- void **zoomOutStep** ()

## Public Member Functions

- void **clear** ()
- int **countItems** () const
- [ItemInfo](#) **findItemByIndex** (const QModelIndex &index) const
- QModelIndex **findItemByInfo** (const [ItemInfo](#) &info) const
- bool **isOnLeftPanel** (const [ItemInfo](#) &info) const
- bool **isOnRightPanel** (const [ItemInfo](#) &info) const
- [LightTableThumbBar](#) (QWidget \*const parent)
- void **removeItemByInfo** (const [ItemInfo](#) &info)
- void **setItems** (const [ItemInfoList](#) &list)
- void **setNavigateByPair** (bool b)
- void **setOnLeftPanel** (const [ItemInfo](#) &info)
- void **setOnRightPanel** (const [ItemInfo](#) &info)
- void **toggleTag** (int tagID)

## Public Member Functions inherited from [Digikam::ItemThumbnailBar](#)

- QModelIndex **firstIndex** () const
- void **installOverlays** ()
- [ItemThumbnailBar](#) (QWidget \*const parent=nullptr)
- QModelIndex **lastIndex** () const
- QModelIndex **nextIndex** (const QModelIndex &index) const
- QModelIndex **previousIndex** (const QModelIndex &index) const
- void **setFlow** (QListView::Flow newFlow)
- void **setModelsFiltered** ([ItemModel](#) \*model, [ImageSortFilterModel](#) \*filterModel)
- void **setScrollBarPolicy** (Qt::ScrollBarPolicy policy)

## Public Member Functions inherited from [Digikam::ItemCategorizedView](#)

- void **addOverlay** ([ItemDelegateOverlay](#) \*overlay, [ItemDelegate](#) \*delegate=nullptr)
 

*Add and remove an overlay. It will as well be removed automatically when destroyed. Unless you pass a different delegate, the current delegate will be used.*
- void **addSelectionOverlay** ([ItemDelegate](#) \*delegate=nullptr)
- [Album](#) \* **albumAt** (const [QPoint](#) &pos) const
- [ItemInfoList](#) **allItemInfos** () const
- [QList](#)< [QUrl](#) > **allUrls** () const
- [Album](#) \* **currentAlbum** () const
- [ItemInfo](#) **currentInfo** () const
- [QUrl](#) **currentUrl** () const
- [ItemDelegate](#) \* **delegate** () const
- [QItemSelectionModel](#) \* **getSelectionModel** () const
- [ItemAlbumFilterModel](#) \* **imageAlbumFilterModel** () const
- [ItemAlbumModel](#) \* **imageAlbumModel** () const
 

*Returns 0 if the [ItemModel](#) is not an [ItemAlbumModel](#).*
- [ItemFilterModel](#) \* **imageFilterModel** () const
 

*Returns any [ItemFilterMode](#) in chain. May not be [sourceModel\(\)](#)*
- [ItemModel](#) \* **imageModel** () const
- [ImageSortFilterModel](#) \* **imageSortFilterModel** () const
- [ItemThumbnailModel](#) \* **imageThumbnailModel** () const
 

*Returns 0 if the [ItemModel](#) is not an [ItemThumbnailModel](#).*
- [QModelIndex](#) **indexForInfo** (const [ItemInfo](#) &info) const
- [ItemCategorizedView](#) ([QWidget](#) \*const parent=nullptr)
- [ItemInfo](#) **nextInfo** (const [ItemInfo](#) &info)
- [ItemInfo](#) **nextInOrder** (const [ItemInfo](#) &startingPoint, int nth)
- [ItemInfo](#) **previousInfo** (const [ItemInfo](#) &info)
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemInfoList](#) **selectedItemInfos** () const
- [ItemInfoList](#) **selectedItemInfosCurrentFirst** () const
- void **setModels** ([ItemModel](#) \*model, [ImageSortFilterModel](#) \*filterModel)
- virtual void **setThumbnailSize** (const [ThumbnailSize](#) &size)
- [ThumbnailSize](#) **thumbnailSize** () const
- void **toIndex** (const [QUrl](#) &url)

## Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **awayFromSelection** ()
- [DItemDelegate](#) \* **delegate** () const
- void **invertSelection** ()
- bool **isToolTipEnabled** () const
- [ItemViewCategorized](#) ([QWidget](#) \*const parent=nullptr)
- int **numberOfSelectedIndexes** () const
- void **scrollTo** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible) override
- void **scrollToRelaxed** (const [QModelIndex](#) &index, [ScrollHint](#) hint=EnsureVisible)
- void **setInitialSelectedItem** (bool enabled)
- void **setScrollCurrentToCenter** (bool enabled)
- void **setScrollStepGranularity** (int factor)
- void **setSelectedIndexes** (const [QList](#)< [QModelIndex](#) > &indexes)
- void **setSpacing** (int spacing)
- void **setToolTipEnabled** (bool enabled)
- void **setUsePointingHandCursor** (bool useCursor)
- void **toFirstIndex** ()
- void **toIndex** (const [QModelIndex](#) &index)
- void **toLastIndex** ()
- void **toNextIndex** ()
- void **toPreviousIndex** ()



## Public Member Functions inherited from Digikam::DCategorizedView

- virtual QModelIndexList [categorizedIndexesIn](#) (const QRect &rect) const
- virtual QModelIndex [categoryAt](#) (const QPoint &point) const
- [DCategoryDrawer](#) \* [categoryDrawer](#) () const
- virtual QItemSelectionRange [categoryRange](#) (const QModelIndex &index) const
- virtual QRect [categoryVisualRect](#) (const QModelIndex &index) const
- [DCategorizedView](#) (QWidget \*const parent=nullptr)
- QModelIndex [indexAt](#) (const QPoint &point) const override
- void [setCategoryDrawer](#) ([DCategoryDrawer](#) \*categoryDrawer)
- void [setDrawDraggedItems](#) (bool drawDraggedItems)
- void [setGridSize](#) (const QSize &size)
- void [setModel](#) (QAbstractItemModel \*model) override
- QRect [visualRect](#) (const QModelIndex &index) const override

## Public Member Functions inherited from Digikam::DragDropViewImplementation

- virtual void [copy](#) ()
- virtual void [cut](#) ()
- virtual void [paste](#) ()

## Public Member Functions inherited from Digikam::GroupingViewImplementation

- [ItemInfoList](#) [getHiddenGroupedInfos](#) (const [ItemInfoList](#) &infos) const
- bool [needGroupResolving](#) ([OperationType](#) type, const [ItemInfoList](#) &infos) const
- [ItemInfoList](#) [resolveGrouping](#) (const [ItemInfoList](#) &infos) const

## Additional Inherited Members

## Protected Slots inherited from Digikam::ItemCategorizedView

- void [slotCurrentUriTimer](#) ()
- void [slotItemInfosAdded](#) ()

## Protected Slots inherited from Digikam::ItemViewCategorized

- void [layoutAboutToBeChanged](#) ()
- void [layoutWasChanged](#) ()
- void [slotActivated](#) (const QModelIndex &index)
- void [slotClicked](#) (const QModelIndex &index)
- void [slotEntered](#) (const QModelIndex &index)
- virtual void [slotThemeChanged](#) ()

## Protected Slots inherited from Digikam::DCategorizedView

- void [currentChanged](#) (const QModelIndex &current, const QModelIndex &previous) override
- void [rowsInserted](#) (const QModelIndex &parent, int start, int end) override
- virtual void [rowsInsertedArtificial](#) (const QModelIndex &parent, int start, int end)
- virtual void [slotLayoutChanged](#) ()
- void [updateGeometries](#) () override

## Protected Member Functions inherited from [Digikam::ItemThumbnailBar](#)

- bool **event** (QEvent \*) override
- bool **hasHiddenGroupedImages** (const [ItemInfo](#) &info) const override  
*must be implemented by parent view*
- void **slotSetupChanged** () override

## Protected Member Functions inherited from [Digikam::ItemCategorizedView](#)

- virtual void **activated** (const [ItemInfo](#) &info, Qt::KeyboardModifiers modifiers)  
*Reimplement these in a subclass.*
- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override
- [ItemInfo](#) **imageInfo** (const QModelIndex &index) const
- [ItemInfoList](#) **imageInfos** (const QList< QModelIndex > &indexes) const
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- void **installDefaultModels** ()  
*install default [ItemAlbumModel](#) and filter model, ready for use*
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([ItemDelegate](#) \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override  
*Reimplement these in a subclass.*
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([DItemDelegate](#) \*delegate)
- void **setToolTip** ([ItemViewToolTip](#) \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

### Protected Member Functions inherited from [Digikam::DCategorizedView](#)

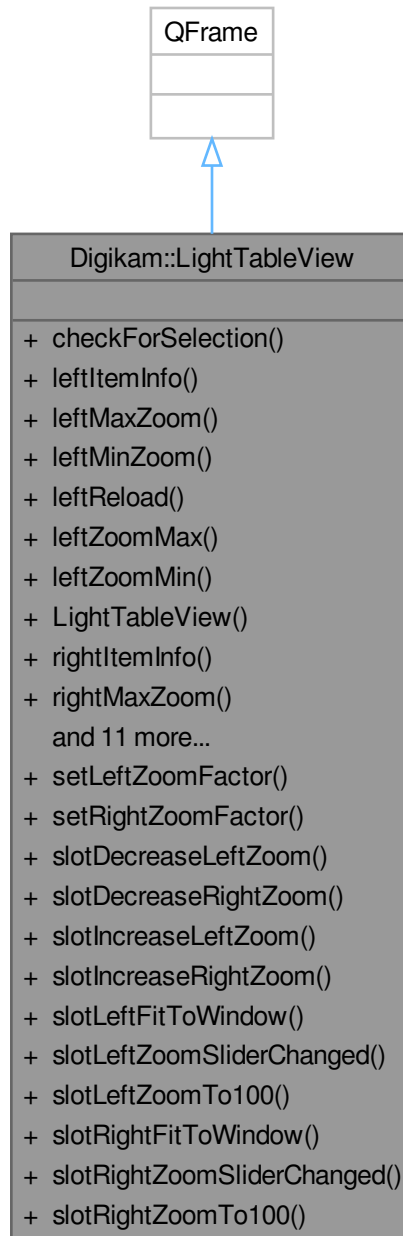
- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

### Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

## 6.1011 Digikam::LightTableView Class Reference

Inheritance diagram for Digikam::LightTableView:



### Public Slots

- void **setLeftZoomFactor** (double z)
- void **setRightZoomFactor** (double z)
- void **slotDecreaseLeftZoom** ()

- void **slotDecreaseRightZoom** ()
- void **slotIncreaseLeftZoom** ()
- void **slotIncreaseRightZoom** ()
- void **slotLeftFitToWindow** ()
- void **slotLeftZoomSliderChanged** (int)
- void **slotLeftZoomTo100** ()
- void **slotRightFitToWindow** ()
- void **slotRightZoomSliderChanged** (int)
- void **slotRightZoomTo100** ()

## Signals

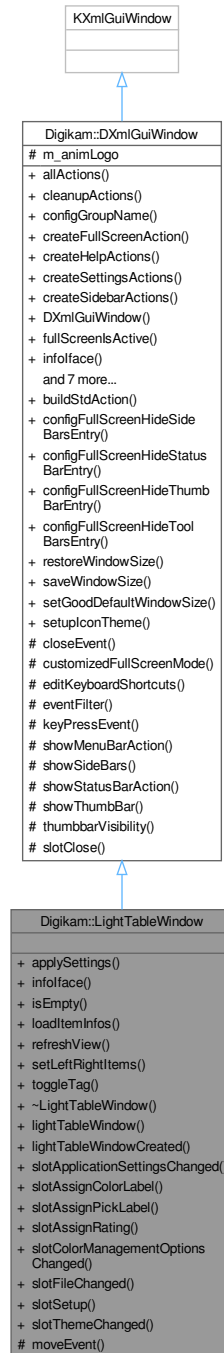
- void **signalDeleteItem** (const [ItemInfo](#) &)
- void **signalEditItem** (const [ItemInfo](#) &)
- void **signalLeftDroppedItems** (const [ItemInfoList](#) &)
- void **signalLeftPanelLeftButtonClicked** ()
- void **signalLeftPopupTagsView** ()
- void **signalLeftPreviewLoaded** (bool)
- void **signalLeftPreviewSelected** (bool)
- void **signalLeftSlideShowCurrent** ()
- void **signalLeftZoomFactorChanged** (double)
- void **signalRightDroppedItems** (const [ItemInfoList](#) &)
- void **signalRightPanelLeftButtonClicked** ()
- void **signalRightPopupTagsView** ()
- void **signalRightPreviewLoaded** (bool)
- void **signalRightPreviewSelected** (bool)
- void **signalRightSlideShowCurrent** ()
- void **signalRightZoomFactorChanged** (double)
- void **signalToggleOnSyncPreview** (bool)

## Public Member Functions

- void **checkForSelection** (const [ItemInfo](#) &info)
- [ItemInfo](#) **leftItemInfo** () const
- bool **leftMaxZoom** () const
- bool **leftMinZoom** () const
- void **leftReload** ()
- double **leftZoomMax** () const
- double **leftZoomMin** () const
- **LightTableView** (QWidget \*const parent=nullptr)
- [ItemInfo](#) **rightItemInfo** () const
- bool **rightMaxZoom** () const
- bool **rightMinZoom** () const
- void **rightReload** ()
- double **rightZoomMax** () const
- double **rightZoomMin** () const
- void **setLeftItemInfo** (const [ItemInfo](#) &info=[ItemInfo](#)())
- void **setNavigateByPair** (bool b)
- void **setPreviewSettings** (const [PreviewSettings](#) &settings)
- void **setRightItemInfo** (const [ItemInfo](#) &info=[ItemInfo](#)())
- void **setSyncPreview** (bool sync)
- void **toggleFullScreen** (bool set)

## 6.1012 Digikam::LightTableWindow Class Reference

Inheritance diagram for Digikam::LightTableWindow:



### Classes

- class [Private](#)

## Public Slots

- void [slotApplicationSettingsChanged](#) ()
- void **slotAssignColorLabel** (int colorId)
- void **slotAssignPickLabel** (int pickId)
- void **slotAssignRating** (int rating)
- void **slotColorManagementOptionsChanged** ()
- void **slotFileChanged** (const QString &filePath)
- void **slotSetup** () override
- void **slotThemeChanged** ()

## Signals

- void **signalWindowHasMoved** ()

## Public Member Functions

- void **applySettings** ()
- [DInfoInterface](#) \* [infoface](#) ([DPluginAction](#) \*const ac) override
- bool **isEmpty** () const
- void **loadItemInfos** (const [ItemInfoList](#) &list, const [ItemInfo](#) &imageInfoCurrent, bool addTo)
- void **refreshView** ()
- void **setLeftRightItems** (const [ItemInfoList](#) &list, bool addTo)
- void **toggleTag** (int tagID)

## Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- QList< QAction \* > **allActions** () const
- void **cleanupActions** ()
- QString **configGroupName** () const
- void **createFullScreenAction** (const QString &name)
- void **createHelpActions** (const QString &handbookSection, bool coreOptions=true)
- void **createSettingsActions** ()
- void **createSidebarActions** ()
- **DXmlGuiWindow** (QWidget \*const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())
- bool **fullScreenIsActive** () const
- void **readFullScreenSettings** (const KConfigGroup &group)
- virtual void **registerExtraPluginsActions** (QString &)
- void **registerPluginsActions** ()
- void **setConfigGroupName** (const QString &name)
- void **setFullScreenOptions** (int options)
- void **unminimizeAndActivateWindow** ()

## Static Public Member Functions

- static [LightTableWindow](#) \* **lightTableWindow** ()
- static bool **lightTableWindowCreated** ()

## Static Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- static QAction \* **buildStdAction** (StdActionType type, const QObject \*const recvr, const char \*const slot, QObject \*const parent)
- static QString **configFullScreenHideSideBarsEntry** ()
- static QString **configFullScreenHideStatusBarEntry** ()
- static QString **configFullScreenHideThumbBarEntry** ()
- static QString **configFullScreenHideToolBarsEntry** ()
- static void **restoreWindowSize** (QWindow \*const win, const KConfigGroup &group)
- static void **saveWindowSize** (QWindow \*const win, KConfigGroup &group)
- static void **setGoodDefaultWindowSize** (QWindow \*const win)
- static void **setupIconTheme** ()

## Protected Member Functions

- void **moveEvent** (QMoveEvent \*e) override

## Protected Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- void **closeEvent** (QCloseEvent \*e) override
- void **editKeyboardShortcuts** (KActionCollection \*const extraac=nullptr, const QString &actitle=QString())
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override
- void **keyPressEvent** (QKeyEvent \*e) override
- QAction \* **showMenuBarAction** () const
- QAction \* **showStatusBarAction** () const
- virtual void **showThumbBar** (bool visible)
- virtual bool **thumbbarVisibility** () const

## Additional Inherited Members

## Protected Slots inherited from [Digikam::DXmlGuiWindow](#)

- bool **slotClose** ()

## Protected Attributes inherited from [Digikam::DXmlGuiWindow](#)

- [DLogoAction](#) \* **m\_animLogo** = nullptr

## 6.1012.1 Member Function Documentation

### 6.1012.1.1 infoface()

```
DInfoInterface * Digikam::LightTableWindow::infoIface (
    DPluginAction *const ac ) [override], [virtual]
```

Return the interface instance to access to items information.

Implements [Digikam::DXmlGuiWindow](#).



### 6.1012.1.2 loadItemInfos()

```
void Digikam::LightTableWindow::loadItemInfos (
    const ItemInfoList & list,
    const ItemInfo & givenItemInfoCurrent,
    bool addTo )
```

We get here either

- via CTRL+L (from the albumview) a) digikamapp.cpp: CTRL+key\_L leads to slotImageLightTable() b) digikamview.cpp: void ItemIconView::slotImageLightTable() calls d->iconView->insertToLightTable(list, info); c) albumiconview.cpp: AlbumIconView::insertToLightTable calls lview->loadItemInfos(list, current);
- via drag&drop, i.e. calls issued by the ...Dropped... routines

### 6.1012.1.3 setLeftRightItems()

```
void Digikam::LightTableWindow::setLeftRightItems (
    const ItemInfoList & list,
    bool addTo )
```

Set the images for the left and right panel.

### 6.1012.1.4 slotApplicationSettingsChanged

```
void Digikam::LightTableWindow::slotApplicationSettingsChanged ( ) [slot]
```

## 6.1013 Digikam::LightTableWindow::Private Class Reference

### Public Member Functions

- void **addPageUpDownActions** (const [LightTableWindow](#) \*const q, QWidget \*const w)

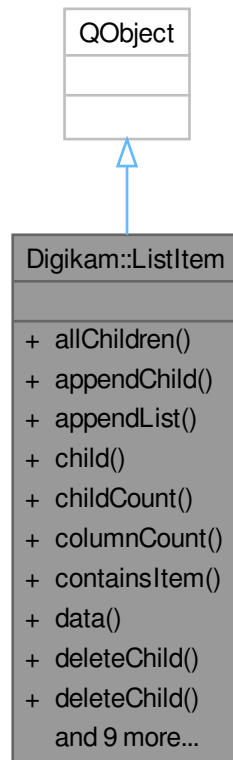
### Public Attributes

- bool **autoLoadOnRightPanel** = true
- bool **autoSyncPreview** = true
- QAction \* **backwardAction** = nullptr
- [ThumbBarDock](#) \* **barViewDock** = nullptr
- QAction \* **clearListAction** = nullptr
- QAction \* **clearOnCloseAction** = nullptr
- QMainWindow \* **dockArea** = nullptr
- QAction \* **editItemAction** = nullptr
- QAction \* **fileDeleteAction** = nullptr
- QAction \* **fileDeleteFinalAction** = nullptr
- QAction \* **firstAction** = nullptr
- QAction \* **forwardAction** = nullptr
- bool **fromLeftPreview** = true
- [SidebarSplitter](#) \* **hSplitter** = nullptr

- QAction \* **lastAction** = nullptr
- DAdjustableLabel \* **leftFileName** = nullptr  
*File name loaded in left preview and show on status bar.*
- ItemPropertiesSideBarDB \* **leftSideBar** = nullptr  
*Side bar to display properties of the left preview and show on the left side..*
- DZoomBar \* **leftZoomBar** = nullptr  
*Zoom bar to control the left preview and show on status bar.*
- QAction \* **leftZoomFitToWindowAction** = nullptr
- QAction \* **leftZoomMinusAction** = nullptr
- QAction \* **leftZoomPlusAction** = nullptr
- QAction \* **leftZoomTo100percents** = nullptr
- QAction \* **navigateByPairAction** = nullptr
- LightTableView \* **previewView** = nullptr  
*Central view including left and right pannels.*
- QAction \* **removeItemAction** = nullptr
- DAdjustableLabel \* **rightFileName** = nullptr  
*File name loaded in right preview and show on status bar.*
- ItemPropertiesSideBarDB \* **rightSideBar** = nullptr  
*Side bar to display properties of the right preview and show on the right side..*
- DZoomBar \* **rightZoomBar** = nullptr  
*Zoom bar to control the right preview and show on status bar.*
- QAction \* **rightZoomFitToWindowAction** = nullptr
- QAction \* **rightZoomMinusAction** = nullptr
- QAction \* **rightZoomPlusAction** = nullptr
- QAction \* **rightZoomTo100percents** = nullptr
- QAction \* **setItemLeftAction** = nullptr
- QAction \* **setItemRightAction** = nullptr
- QAction \* **showBarAction** = nullptr
- StatusProgressBar \* **statusProgressBar** = nullptr
- QAction \* **syncPreviewAction** = nullptr
- LightTableThumbBar \* **thumbView** = nullptr  
*Top level view to host thumbbar.*
- QAction \* **viewCMViewAction** = nullptr

## 6.1014 Digikam::ListItem Class Reference

Inheritance diagram for Digikam::ListItem:



### Public Member Functions

- `QList< ListItem * > allChildren ()` const
- void **appendChild** (`ListItem *const child`)
- void **appendList** (`const QList< ListItem * > &items`)
- `ListItem * child` (`int row`) const
- int **childCount** () const
- int **columnCount** () const
- `ListItem * containsItem` (`ListItem *const item`) const  
*containsItem - search child items if contains a ListItem with the same data as item*
- `QVariant data` (`int column`) const
- void **deleteChild** (`int row`)
- void **deleteChild** (`ListItem *const item`)
- bool **equal** (`ListItem *const item`) const
- `QList< int > getTagIds` () const
- `ListItem` (`QList< QVariant > &data`, `ListItem *const parent=nullptr`)
- `ListItem * parent` () const
- void **removeAll** ()
- void **removeTagId** (`int tagId`)
- int **row** () const
- void **setData** (`const QList< QVariant > &data`)

## 6.1014.1 Member Function Documentation

### 6.1014.1.1 containsItem()

```
ListItem * Digikam::ListItem::containsItem (  
    ListItem *const item ) const
```

#### Parameters

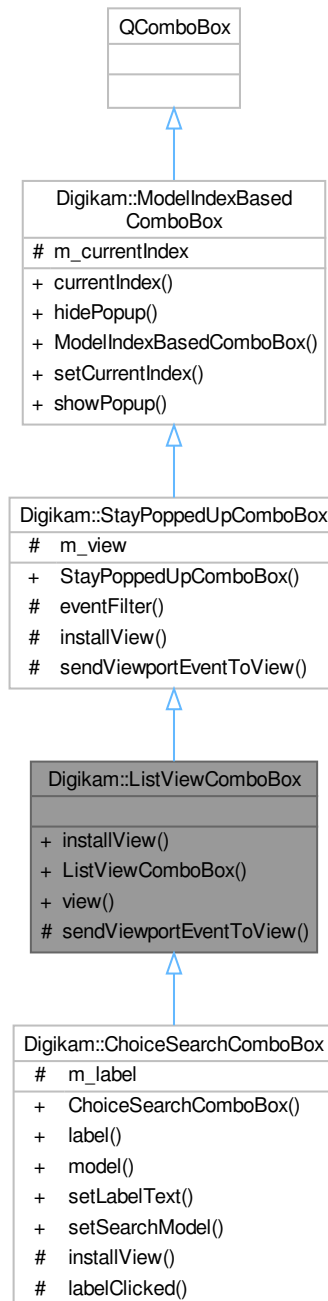
<i>item</i>	- <a href="#">ListItem</a> pointer for which we should search if there is a similar item
-------------	--

#### Returns

- NULL if no similar item was found and a valid [ListItem](#) if a [ListItem](#) with the same data was found

## 6.1015 Digikam::ListViewComboBox Class Reference

Inheritance diagram for Digikam::ListViewComboBox:



### Public Member Functions

- virtual void `installView` (QAbstractItemView \*`view`=nullptr)
- `ListViewComboBox` (QWidget \*`parent`=nullptr)
- QListView \* `view` () const

## Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex **currentIndex** () const
- void **hidePopup** () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

## Protected Member Functions

- void [sendViewportEventToView](#) (QEvent \*e) override

## Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- void [installView](#) (QAbstractItemView \*view)

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

- QAbstractItemView \* **m\_view** = nullptr

## Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex **m\_currentIndex**

## 6.1015.1 Constructor & Destructor Documentation

### 6.1015.1.1 [ListViewComboBox\(\)](#)

```
Digikam::ListViewComboBox::ListViewComboBox (
    QWidget * parent = nullptr ) [explicit]
```

This class provides an implementation of a [StayPoppedUpComboBox](#) with a QListView. This is the standard view of a QComboBox, but in conjunction with [StayPoppedUpComboBox](#) some extra steps are needed. You need three steps: Construct the object, call `setModel()` with an appropriate QAbstractItemModel, then call [installView\(\)](#).

## 6.1015.2 Member Function Documentation

### 6.1015.2.1 installView()

```
void Digikam::ListViewComboBox::installView (
    QAbstractItemView * view = nullptr ) [virtual]
```

Replace the standard combo box list view with a QTreeView. Call this after installing an appropriate model.

Reimplemented in [Digikam::ChoiceSearchComboBox](#).

### 6.1015.2.2 sendViewportEventToView()

```
void Digikam::ListViewComboBox::sendViewportEventToView (
    QEvent * e ) [override], [protected], [virtual]
```

Implement in subclass: Send the given event to the viewportEvent() method of m\_view. This method is protected for a usual QAbstractItemView. You can override, pass a view, and call parent implementation. The existing view will be used. You must then also reimplement sendViewportEventToView.

Implements [Digikam::StayPoppedUpComboBox](#).

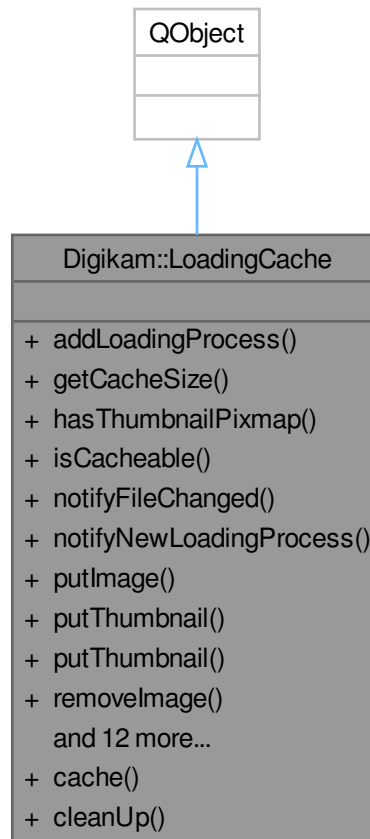
### 6.1015.2.3 view()

```
QListView * Digikam::ListViewComboBox::view ( ) const
```

Returns the QTreeView of this class. Valid after [installView\(\)](#) has been called.

## 6.1016 Digikam::LoadingCache Class Reference

Inheritance diagram for Digikam::LoadingCache:



### Classes

- class [CacheLock](#)

### Signals

- void [fileChanged](#) (const QString &filePath)

### Public Member Functions

- void [addLoadingProcess](#) ([LoadingProcess](#) \*const process)
- quint64 [getCacheSize](#) () const
- bool [hasThumbnailPixmap](#) (const QString &cacheKey) const
- bool [isCacheable](#) (const [DImg](#) &img) const
- void [notifyFileChanged](#) (const QString &filePath, bool notify=true)



- void [notifyNewLoadingProcess](#) ([LoadingProcess](#) \*const process, const [LoadingDescription](#) &description)
- bool [putImage](#) (const QString &cacheKey, const [DImg](#) &img, const QString &filePath) const
- void [putThumbnail](#) (const QString &cacheKey, const QImage &thumb, const QString &filePath)
- void [putThumbnail](#) (const QString &cacheKey, const QPixmap &thumb, const QString &filePath)
- void [removeImage](#) (const QString &cacheKey)
- void [removeImages](#) ()
- void [removeLoadingProcess](#) ([LoadingProcess](#) \*const process)
- void [removeThumbnail](#) (const QString &cacheKey)
- void [removeThumbnails](#) ()
- const QPixmap \* [retrieveBufferedTPixmap](#) (const QString &cacheKey) const
- [DImg](#) \* [retrieveImage](#) (const QString &cacheKey) const
- [LoadingProcess](#) \* [retrieveLoadingProcess](#) (const QString &cacheKey) const
- const QImage \* [retrieveThumbnail](#) (const QString &cacheKey) const
- const QPixmap \* [retrieveThumbnailPixmap](#) (const QString &cacheKey) const
- void [setCacheSize](#) (int megabytes)
- void [setFileWatch](#) ([LoadingCacheFileWatch](#) \*const watch)
- void [setThumbnailCacheSize](#) (int numberOfQImages, int numberOfQPixmaps)

### Static Public Member Functions

- static [LoadingCache](#) \* [cache](#) ()
- static void [cleanUp](#) ()

### Friends

- class [CacheLock](#)
- class [LoadingCacheFileWatch](#)

## 6.1016.1 Member Function Documentation

### 6.1016.1.1 addLoadingProcess()

```
void Digikam::LoadingCache::addLoadingProcess (
    LoadingProcess *const process )
```

Add a loading process to the list. Only one loading process for the same cache key is registered at a time.

### 6.1016.1.2 fileChanged

```
void Digikam::LoadingCache::fileChanged (
    const QString & filePath ) [signal]
```

This signal is emitted when the cache is notified that a file was changed. There is no information in this signal if the file was ever contained in the cache. The signal may be emitted under [CacheLock](#). Strongly consider a queued connection.

### 6.1016.1.3 getCacheSize()

```
quint64 Digikam::LoadingCache::getCacheSize ( ) const
```

Get the cache size in bytes.

#### 6.1016.1.4 isCacheable()

```
bool Digikam::LoadingCache::isCacheable (
    const DImg & img ) const
```

Returns whether the given [DImg](#) fits in the cache.

#### 6.1016.1.5 notifyFileChanged()

```
void Digikam::LoadingCache::notifyFileChanged (
    const QString & filePath,
    bool notify = true )
```

Remove all entries from cache that were loaded from filePath. Emits relevant signals if notify = true.

#### 6.1016.1.6 notifyNewLoadingProcess()

```
void Digikam::LoadingCache::notifyNewLoadingProcess (
    LoadingProcess *const process,
    const LoadingDescription & description )
```

Notify all currently registered loading processes

#### 6.1016.1.7 putImage()

```
bool Digikam::LoadingCache::putImage (
    const QString & cacheKey,
    const DImg & img,
    const QString & filePath ) const
```

Put image into for given string into the cache. Returns true if image has been put in the cache, false otherwise. Ownership of the [DImg](#) instance is passed to the cache. When it cannot be put in the cache it is deleted. The third parameter specifies a file path that will be watched. If this file changes, the object will be removed from the cache.

#### 6.1016.1.8 putThumbnail()

```
void Digikam::LoadingCache::putThumbnail (
    const QString & cacheKey,
    const QImage & thumb,
    const QString & filePath )
```

Puts a thumbnail into the thumbnail cache.

#### 6.1016.1.9 removeImage()

```
void Digikam::LoadingCache::removeImage (
    const QString & cacheKey )
```

Remove entries for the given cacheKey from the cache

### 6.1016.1.10 removeImages()

```
void Digikam::LoadingCache::removeImages ( )
```

Remove all entries from the cache

### 6.1016.1.11 removeLoadingProcess()

```
void Digikam::LoadingCache::removeLoadingProcess (
    LoadingProcess *const process )
```

Remove loading process for given cache key

### 6.1016.1.12 removeThumbnail()

```
void Digikam::LoadingCache::removeThumbnail (
    const QString & cacheKey )
```

Remove the thumbnail for the given file path from the thumbnail cache

### 6.1016.1.13 removeThumbnails()

```
void Digikam::LoadingCache::removeThumbnails ( )
```

Remove all thumbnails

### 6.1016.1.14 retrieveImage()

```
DImg * Digikam::LoadingCache::retrieveImage (
    const QString & cacheKey ) const
```

Retrieves an image for the given string from the cache, or 0 if no image is found.

### 6.1016.1.15 retrieveLoadingProcess()

```
LoadingProcess * Digikam::LoadingCache::retrieveLoadingProcess (
    const QString & cacheKey ) const
```

Find the loading process for given cacheKey, or 0 if not found

### 6.1016.1.16 retrieveThumbnail()

```
const QImage * Digikam::LoadingCache::retrieveThumbnail (
    const QString & cacheKey ) const
```

The [LoadingCache](#) support both the caching of QImage and QPixmap objects. QPixmaps can only be accessed from the main thread, so the tasks cannot access this cache. Retrieves a thumbnail for the given filePath from the thumbnail cache, or a 0 if the thumbnail is not found.

**6.1016.1.17 setCacheSize()**

```
void Digikam::LoadingCache::setCacheSize (
    int megabytes )
```

Sets the cache size in megabytes. The thumbnail cache is not affected and `setThumbnailCacheSize` takes the maximum number.

**6.1016.1.18 setFileWatch()**

```
void Digikam::LoadingCache::setFileWatch (
    LoadingCacheFileWatch *const watch )
```

Sets a [LoadingCacheFileWatch](#) to watch the files contained in this cache. Ownership of this object is transferred to the cache.

**6.1016.1.19 setThumbnailCacheSize()**

```
void Digikam::LoadingCache::setThumbnailCacheSize (
    int numberOfQImages,
    int numberOfQPixmaps )
```

Sets the size of the thumbnail cache

**Parameters**

<i>numberOfQImages</i>	The maximum number of thumbnails of max possible size in QImage format that will be cached. If the size of the images is smaller, a larger number will be cached.
<i>numberOfQPixmaps</i>	The maximum number of thumbnails of max possible size in QPixmap format that will be cached. If the size of the images is smaller, a larger number will be cached. Note: The main cache is unaffected by this method, and <code>setCacheSize</code> takes megabytes as parameter. Note: A good caching strategy will be to set one of the numbers to 0 Default values: (0, 100)

**6.1017 Digikam::LoadingCache::CacheLock Class Reference****Public Member Functions**

- **CacheLock** ([LoadingCache](#) \*const cache)
- void **timedWait** ()
- void **wakeAll** ()

**6.1017.1 Detailed Description****Warning**

All methods of [LoadingCache](#) shall only be called when a [CacheLock](#) is held

## 6.1018 Digikam::LoadingCacheFileWatch Class Reference

Inheritance diagram for Digikam::LoadingCacheFileWatch:



### Public Member Functions

- void **addedImage** (const QString &filePath)
- void **checkFileWatch** (const QString &filePath)
- void **removeImage** (const QString &filePath)

### Protected Member Functions

- void [notifyFileChanged](#) (const QString &filePath)

### Protected Attributes

- class [LoadingCache](#) \* **m\_cache** = nullptr
- QHash< QString, QPair< quint64, QDateTime > > **m\_watchHash**

## Friends

- class **LoadingCache**

## 6.1018.1 Member Function Documentation

### 6.1018.1.1 notifyFileChanged()

```
void Digikam::LoadingCacheFileWatch::notifyFileChanged (
    const QString & filePath ) [protected]
```

Convenience method. Call this to tell the cache to remove stored images for filePath from the cache. Calling this method is fast, you do not need to check if the file is contained in the cache. Do not hold the CacheLock when calling this method.

## 6.1019 Digikam::LoadingCacheInterface Class Reference

### Static Public Member Functions

- static void [cleanCache](#) ()
- static void [cleanThumbnailCache](#) ()
- static void [cleanUp](#) ()
- static void [connectToSignalFileChanged](#) (QObject \*const object, const char \*slot)
- static void [fileChanged](#) (const QString &filePath, bool notify=true)
- static void [initialize](#) ()
- static void [putImage](#) (const QString &filePath, const [DImg](#) &img)
- static void [setCacheOptions](#) (int cacheSize)

## 6.1019.1 Member Function Documentation

### 6.1019.1.1 cleanCache()

```
void Digikam::LoadingCacheInterface::cleanCache ( ) [static]
```

remove all images from the cache (e.g. when loading settings changed) Does not affect thumbnails.

### 6.1019.1.2 cleanThumbnailCache()

```
void Digikam::LoadingCacheInterface::cleanThumbnailCache ( ) [static]
```

Remove all thumbnails from the thumbnail cache. Does not affect main image cache.

### 6.1019.1.3 cleanUp()

```
void Digikam::LoadingCacheInterface::cleanUp ( ) [static]
```

clean up cache at shutdown

#### 6.1019.1.4 connectToSignalFileChanged()

```
void Digikam::LoadingCacheInterface::connectToSignalFileChanged (
    QObject *const object,
    const char * slot ) [static]
```

Connect the given object/slot to the signal void fileChanged(const QString& filePath); which is emitted when the cache gains knowledge about a possible change of this file on disk.

#### 6.1019.1.5 fileChanged()

```
void Digikam::LoadingCacheInterface::fileChanged (
    const QString & filePath,
    bool notify = true ) [static]
```

Remove an image from the cache because it may have changed on disk

#### 6.1019.1.6 putImage()

```
void Digikam::LoadingCacheInterface::putImage (
    const QString & filePath,
    const QImage & img ) [static]
```

add a copy of the image to cache

#### 6.1019.1.7 setCacheOptions()

```
void Digikam::LoadingCacheInterface::setCacheOptions (
    int cacheSize ) [static]
```

Set cache size in Megabytes. Set to 0 to disable caching.

## 6.1020 Digikam::LoadingDescription Class Reference

### Classes

- class [PostProcessingParameters](#)
- class [PreviewParameters](#)

### Public Types

- enum [ColorManagementSettings](#) { [NoColorConversion](#) , [ApplyTransform](#) , [ConvertForEditor](#) , [ConvertToSRGB](#) , [ConvertForDisplay](#) , [ConvertForOutput](#) }
- enum [RawDecodingHint](#) { [RawDecodingDefaultSettings](#) , [RawDecodingGlobalSettings](#) , [RawDecodingCustomSettings](#) , [RawDecodingTimeOptimized](#) }

## Public Member Functions

- QString [cacheKey](#) () const
- bool [equalsIgnoreReducedVersion](#) (const [LoadingDescription](#) &other) const
- bool [equalsOrBetterThan](#) (const [LoadingDescription](#) &other) const
- bool [isPreviewImage](#) () const
- bool [isReducedVersion](#) () const
- bool [isThumbnail](#) () const
- [LoadingDescription](#) ()
- [LoadingDescription](#) (const QString &filePath, [ColorManagementSettings](#)=NoColorConversion)
- [LoadingDescription](#) (const QString &filePath, const [DRawDecoding](#) &settings, [RawDecodingHint](#) rawDecodingHint=[RawDecodingCustomSettings](#), [ColorManagementSettings](#)=NoColorConversion)
- [LoadingDescription](#) (const QString &filePath, const [PreviewSettings](#) &settings, int size, [ColorManagementSettings](#)=NoColorConversion, [PreviewParameters::PreviewType](#)=[PreviewParameters::PreviewImage](#))
- QStringList [lookupCacheKeys](#) () const
- bool [needCheckRawDecoding](#) () const
- bool **operator!=** (const [LoadingDescription](#) &other) const
- bool **operator==** (const [LoadingDescription](#) &other) const
- [ThumbnailIdentifier](#) [thumbnailIdentifier](#) () const

## Static Public Member Functions

- static QStringList [possibleCacheKeys](#) (const QString &filePath)
- static QStringList [possibleThumbnailCacheKeys](#) (const QString &filePath)

## Public Attributes

- QString [filePath](#)
- [PostProcessingParameters](#) [postProcessingParameters](#)
- [PreviewParameters](#) [previewParameters](#)
- [RawDecodingHint](#) [rawDecodingHint](#) = [RawDecodingDefaultSettings](#)
- [DRawDecoding](#) [rawDecodingSettings](#)

## 6.1020.1 Member Enumeration Documentation

### 6.1020.1.1 ColorManagementSettings

```
enum Digikam::LoadingDescription::ColorManagementSettings
```

#### Enumerator

ApplyTransform	IccData is an <a href="#">IccTransform</a> .
ConvertForDisplay	IccData can be the output profile.
ConvertForOutput	IccData is the output profile.

### 6.1020.1.2 RawDecodingHint

```
enum Digikam::LoadingDescription::RawDecodingHint
```



## Enumerator

RawDecodingDefaultSettings	The raw decoding options passed are taken from default, hardcoded settings
RawDecodingGlobalSettings	The raw decoding options passed are taken from global settings
RawDecodingCustomSettings	The raw decoding options may be customly edited by the user
RawDecodingTimeOptimized	The raw decoding options are hardcoded settings optimized for loading time The halfSizeColorImage and 16bit settings can be adjusted separately

## 6.1020.2 Constructor & Destructor Documentation

### 6.1020.2.1 LoadingDescription() [1/4]

```
Digikam::LoadingDescription::LoadingDescription ( )
```

An invalid [LoadingDescription](#)

### 6.1020.2.2 LoadingDescription() [2/4]

```
Digikam::LoadingDescription::LoadingDescription (
    const QString & filePath,
    ColorManagementSettings cm = NoColorConversion ) [explicit]
```

Use this for full loading of non-raw files

### 6.1020.2.3 LoadingDescription() [3/4]

```
Digikam::LoadingDescription::LoadingDescription (
    const QString & filePath,
    const DRawDecoding & settings,
    RawDecodingHint rawDecodingHint = RawDecodingCustomSettings,
    ColorManagementSettings cm = NoColorConversion )
```

Use this for full loading of raw files

### 6.1020.2.4 LoadingDescription() [4/4]

```
Digikam::LoadingDescription::LoadingDescription (
    const QString & filePath,
    const PreviewSettings & settings,
    int size,
    ColorManagementSettings cm = NoColorConversion,
    PreviewParameters::PreviewType type = PreviewParameters::PreviewImage )
```

For preview and thumbnail jobs: Stores preview max size and Exif rotation. Raw files / preview jobs: If size is not 0, the embedded preview will be loaded if available. If size is 0, [DImg](#) based loading will be used with default raw decoding settings. You can also adjust raw decoding settings and hint in this case.

### 6.1020.3 Member Function Documentation

#### 6.1020.3.1 cacheKey()

```
QString Digikam::LoadingDescription::cacheKey ( ) const
```

Return the cache key for this description

#### 6.1020.3.2 equalsIgnoreReducedVersion()

```
bool Digikam::LoadingDescription::equalsIgnoreReducedVersion (
    const LoadingDescription & other ) const
```

Returns whether the other loading task equals this one ignoring parameters used to specify a reduced version.

#### 6.1020.3.3 equalsOrBetterThan()

```
bool Digikam::LoadingDescription::equalsOrBetterThan (
    const LoadingDescription & other ) const
```

Returns whether this loading task equals the other one or is superior to it, if the other one is a reduced version

#### 6.1020.3.4 isPreviewImage()

```
bool Digikam::LoadingDescription::isPreviewImage ( ) const
```

Returns if this description will load a preview

#### 6.1020.3.5 isReducedVersion()

```
bool Digikam::LoadingDescription::isReducedVersion ( ) const
```

Returns whether this description describes a loading operation which loads the image in a reduced version (quality, size etc.)

#### 6.1020.3.6 isThumbnail()

```
bool Digikam::LoadingDescription::isThumbnail ( ) const
```

Returns if this description will load a thumbnail

#### 6.1020.3.7 lookupCacheKeys()

```
QStringList Digikam::LoadingDescription::lookupCacheKeys ( ) const
```

Return all possible cache keys, starting with the best choice, for which a result may be found in the cache for this description. Included in the list are better quality versions, if this description is reduced.

### 6.1020.3.8 needCheckRawDecoding()

```
bool Digikam::LoadingDescription::needCheckRawDecoding ( ) const
```

For some RAW images, the same cache key is not enough to say it is the correct result. You must check the raw decoding settings in this case.

### 6.1020.3.9 operator==( )

```
bool Digikam::LoadingDescription::operator== (
    const LoadingDescription & other ) const
```

Returns whether the other loading task equals this one

### 6.1020.3.10 possibleCacheKeys()

```
QStringList Digikam::LoadingDescription::possibleCacheKeys (
    const QString & filePath ) [static]
```

Returns all possible cacheKeys for the given file path (all cache keys under which the given file could be stored in the cache).

### 6.1020.3.11 thumbnailIdentifier()

```
ThumbnailIdentifier Digikam::LoadingDescription::thumbnailIdentifier ( ) const
```

If this referenced a thumbnail, recreate the identifier

## 6.1021 Digikam::LoadingDescription::PostProcessingParameters Class Reference

### Public Member Functions

- bool **hasProfile** ( ) const
- bool **hasTransform** ( ) const
- bool **needsProcessing** ( ) const
- bool **operator==** (const [PostProcessingParameters](#) &other) const
- [IccProfile](#) **profile** ( ) const
- void **setProfile** (const [IccProfile](#) &profile)
- void **setTransform** (const [IccTransform](#) &transform)
- [IccTransform](#) **transform** ( ) const

### Public Attributes

- [ColorManagementSettings](#) **colorManagement** = NoColorConversion
- QVariant **iccData**

## 6.1022 Digikam::LoadingDescription::PreviewParameters Class Reference

### Public Types

- enum **PreviewFlag** { **NoFlags** = 0 , **OnlyPregenerate** = 1 << 0 , **OnlyFromStorage** = 1 << 1 }
- enum **PreviewType** { **NoPreview** , **PreviewImage** , **Thumbnail** , **DetailThumbnail** }

### Public Member Functions

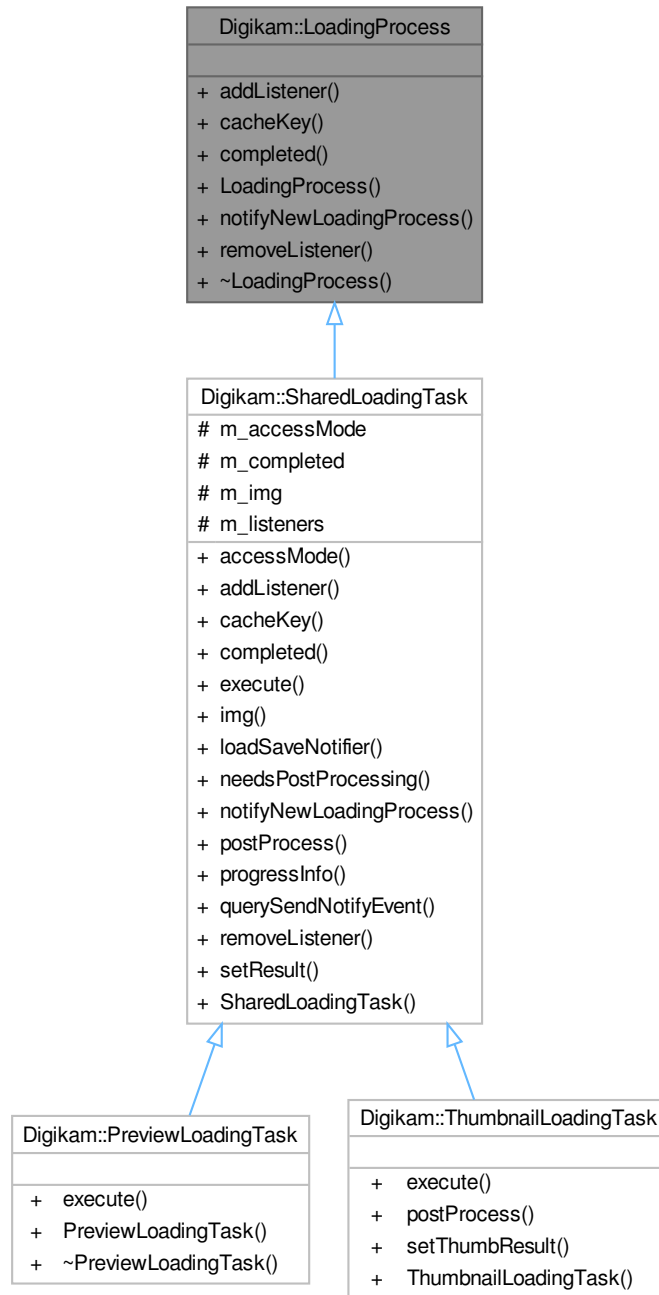
- bool **onlyFromStorage** () const
- bool **onlyPregenerate** () const
- bool **operator==** (const [PreviewParameters](#) &other) const

### Public Attributes

- QVariant **extraParameter**
- PreviewFlags **flags** = NoFlags
- [PreviewSettings](#) **previewSettings**
- int **size** = 0
- QVariant **storageReference**
- PreviewType **type** = NoPreview

## 6.1023 Digikam::LoadingProcess Class Reference

Inheritance diagram for Digikam::LoadingProcess:



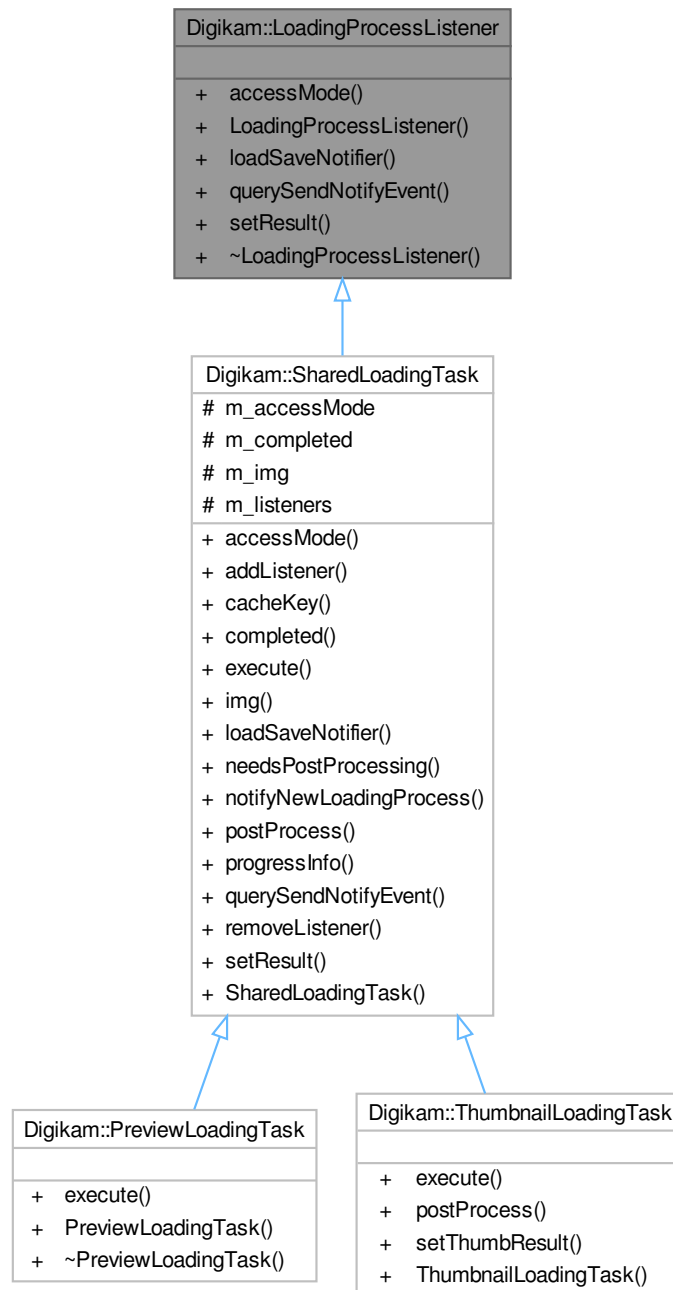
### Public Member Functions

- virtual void **addListener** ([LoadingProcessListener](#) \*const listener)=0
- virtual QString **cacheKey** () const =0

- virtual bool **completed** () const =0
- virtual void **notifyNewLoadingProcess** ([LoadingProcess](#) \*const process, const [LoadingDescription](#) &description)=0
- virtual void **removeListener** ([LoadingProcessListener](#) \*const listener)=0

## 6.1024 Digikam::LoadingProcessListener Class Reference

Inheritance diagram for Digikam::LoadingProcessListener:

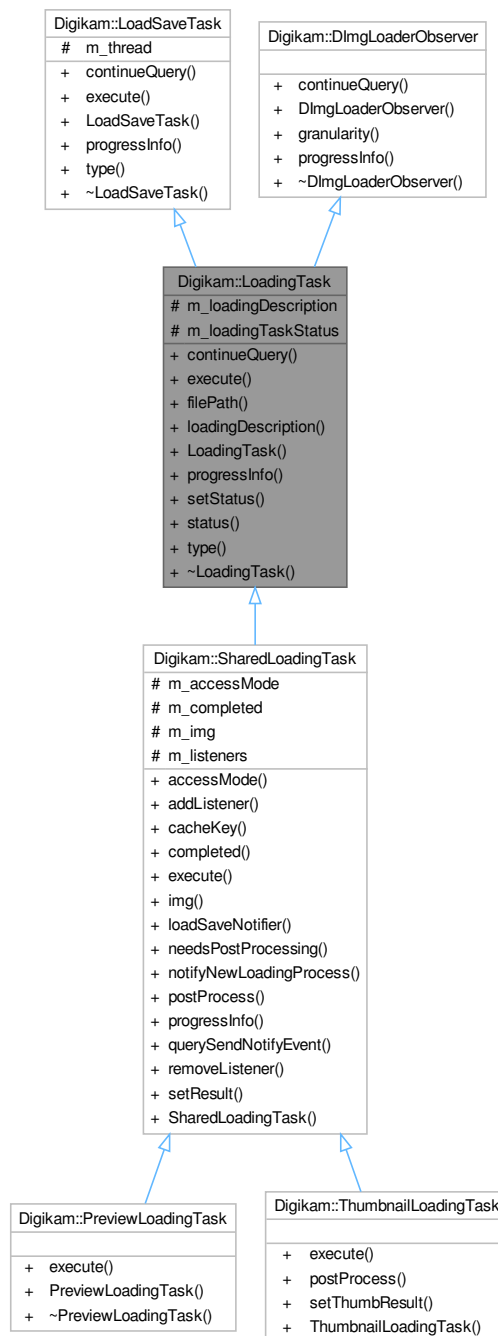


## Public Member Functions

- virtual [LoadSaveThread::AccessMode](#) **accessMode** () const =0
- virtual [LoadSaveNotifier](#) \* **loadSaveNotifier** () const =0
- virtual bool **querySendNotifyEvent** () const =0
- virtual void **setResult** (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)=0

## 6.1025 Digikam::LoadingTask Class Reference

Inheritance diagram for Digikam::LoadingTask:



## Public Types

- enum **LoadingTaskStatus** { **LoadingTaskStatusLoading** , **LoadingTaskStatusPreloading** , **LoadingTaskStatusStopping** }

## Public Types inherited from [Digikam::LoadSaveTask](#)

- enum **TaskType** { **TaskTypeLoading** , **TaskTypeSaving** }

## Public Member Functions

- bool [continueQuery](#) () override
- void [execute](#) () override
- QString [filePath](#) () const
- const [LoadingDescription](#) & [loadingDescription](#) () const
- **LoadingTask** ([LoadSaveThread](#) \*const thread, const [LoadingDescription](#) &description, LoadingTaskStatus loadingTaskStatus=**LoadingTaskStatusLoading**)
- void [progressInfo](#) (float progress) override
- void [setStatus](#) (LoadingTaskStatus status)
- LoadingTaskStatus **status** () const
- TaskType [type](#) () override

## Public Member Functions inherited from [Digikam::LoadSaveTask](#)

- **LoadSaveTask** ([LoadSaveThread](#) \*const thread)

## Public Member Functions inherited from [Digikam::DImgLoaderObserver](#)

- virtual float [granularity](#) ()

## Protected Attributes

- [LoadingDescription](#) **m\_loadingDescription**
- volatile LoadingTaskStatus **m\_loadingTaskStatus** = LoadingTaskStatusLoading

## Protected Attributes inherited from [Digikam::LoadSaveTask](#)

- [LoadSaveThread](#) \* **m\_thread** = nullptr

## 6.1025.1 Member Function Documentation

### 6.1025.1.1 [continueQuery\(\)](#)

bool [Digikam::LoadingTask::continueQuery](#) ( ) [override], [virtual]

Implements [Digikam::LoadSaveTask](#).



### 6.1025.1.2 execute()

```
void Digikam::LoadingTask::execute ( ) [override], [virtual]
```

Implements [Digikam::LoadSaveTask](#).

### 6.1025.1.3 progressInfo()

```
void Digikam::LoadingTask::progressInfo (
    float progress ) [override], [virtual]
```

Implements [Digikam::LoadSaveTask](#).

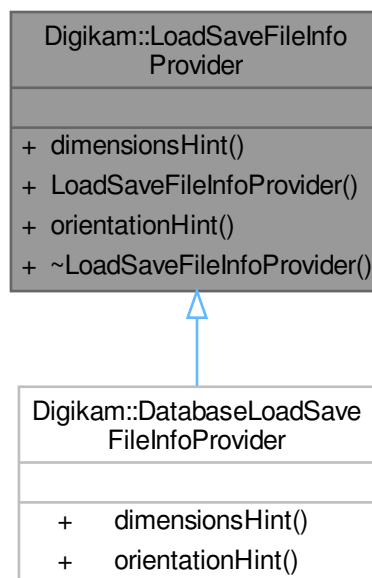
### 6.1025.1.4 type()

```
LoadingTask::TaskType Digikam::LoadingTask::type ( ) [override], [virtual]
```

Implements [Digikam::LoadSaveTask](#).

## 6.1026 Digikam::LoadSaveFileInfoProvider Class Reference

Inheritance diagram for Digikam::LoadSaveFileInfoProvider:



## Public Member Functions

- virtual QSize [dimensionsHint](#) (const QString &path)=0
- virtual int [orientationHint](#) (const QString &path)=0

### 6.1026.1 Member Function Documentation

#### 6.1026.1.1 dimensionsHint()

```
virtual QSize Digikam::LoadSaveFileInfoProvider::dimensionsHint (  
    const QString & path ) [pure virtual]
```

Gives a hint at the size of the image. This can be used to supersede the Exif information in the file.

Implemented in [Digikam::DatabaseLoadSaveFileInfoProvider](#).

#### 6.1026.1.2 orientationHint()

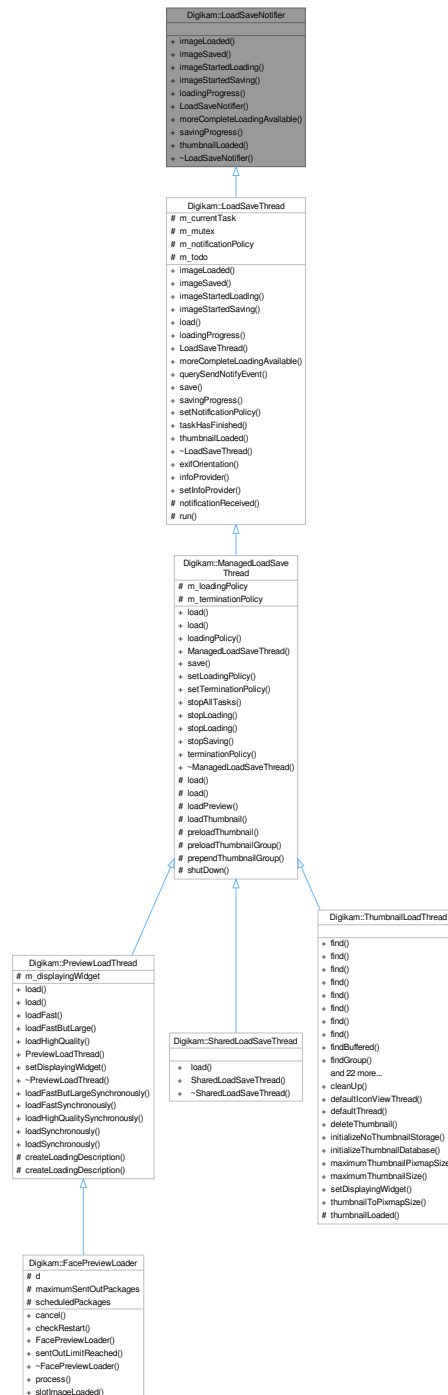
```
virtual int Digikam::LoadSaveFileInfoProvider::orientationHint (  
    const QString & path ) [pure virtual]
```

Gives a hint at the orientation of the image. This can be used to supersede the Exif information in the file. Will not be used if DMetadata::ORIENTATION\_UNSPECIFIED (default value)

Implemented in [Digikam::DatabaseLoadSaveFileInfoProvider](#).

## 6.1027 Digikam::LoadSaveNotifier Class Reference

Inheritance diagram for Digikam::LoadSaveNotifier:



### Public Member Functions

- virtual void **imageLoaded** (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)=0
- virtual void **imageSaved** (const QString &filePath, bool success)=0

- virtual void **imageStartedLoading** (const [LoadingDescription](#) &loadingDescription)=0
- virtual void **imageStartedSaving** (const QString &filePath)=0
- virtual void **loadingProgress** (const [LoadingDescription](#) &loadingDescription, float progress)=0
- virtual void **moreCompleteLoadingAvailable** (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription)=0
- virtual void **savingProgress** (const QString &filePath, float progress)=0
- virtual void **thumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QImage &img)=0

## 6.1027.1 Member Function Documentation

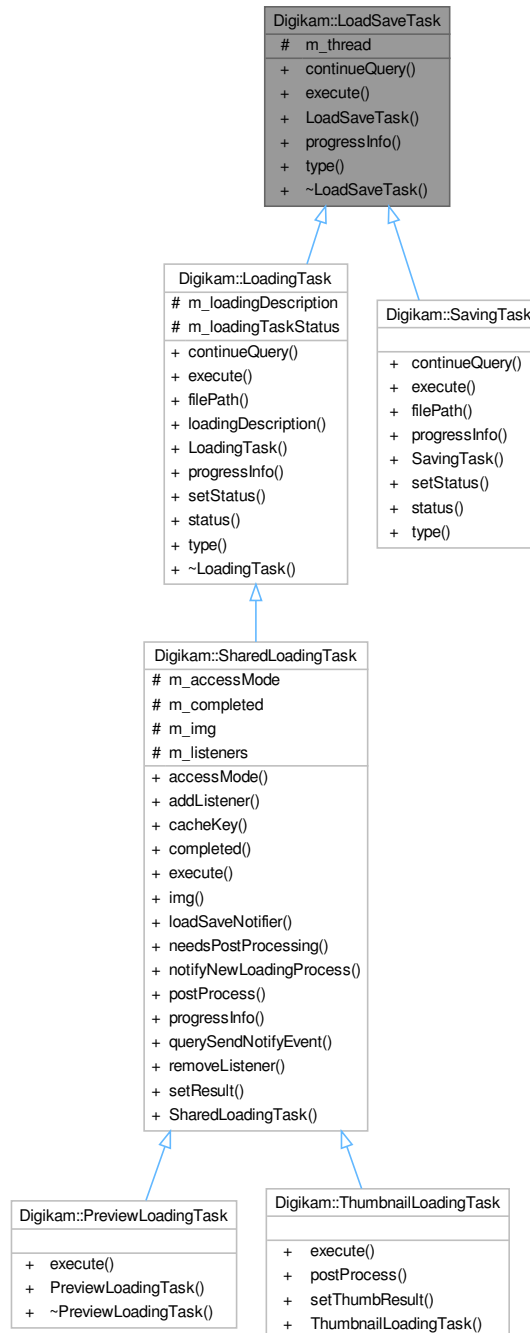
### 6.1027.1.1 thumbnailLoaded()

```
virtual void Digikam::LoadSaveNotifier::thumbnailLoaded (  
    const LoadingDescription & loadingDescription,  
    const QImage & img ) [pure virtual]
```

Implemented in [Digikam::ThumbnailLoadThread](#).

## 6.1028 Digikam::LoadSaveTask Class Reference

Inheritance diagram for Digikam::LoadSaveTask:



### Public Types

- enum **TaskType** { **TaskTypeLoading** , **TaskTypeSaving** }

**Public Member Functions**

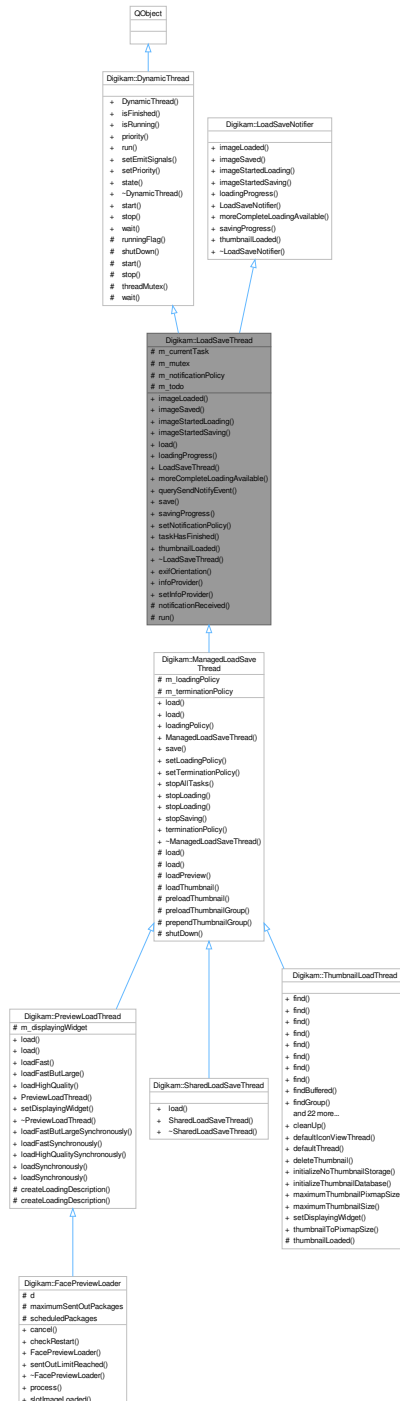
- virtual bool **continueQuery** ()=0
- virtual void **execute** ()=0
- **LoadSaveTask** ([LoadSaveThread](#) \*const thread)
- virtual void **progressInfo** (float progress)=0
- virtual TaskType **type** ()=0

**Protected Attributes**

- [LoadSaveThread](#) \* **m\_thread** = nullptr

## 6.1029 Digikam::LoadSaveThread Class Reference

Inheritance diagram for Digikam::LoadSaveThread:



### Public Types

- enum [AccessMode](#) { [AccessModeRead](#) , [AccessModeReadWrite](#) }
- enum [NotificationPolicy](#) { [NotificationPolicyDirect](#) , [NotificationPolicyTimeLimited](#) }

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Signals

- void [signalImageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)
- void **signalImageSaved** (const QString &filePath, bool success)
- void [signalImageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription)
- void **signalImageStartedSaving** (const QString &filePath)
- void [signalLoadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress)
- void [signalMoreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription)
- void **signalSavingProgress** (const QString &filePath, float progress)
- void **signalThumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QImage &img)

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void [starting](#) ()

## Public Member Functions

- void [imageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img) override
- void [imageSaved](#) (const QString &filePath, bool success) override
- void [imageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription) override
- void [imageStartedSaving](#) (const QString &filePath) override
- void [load](#) (const [LoadingDescription](#) &description)
- void [loadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress) override
- **LoadSaveThread** (QObject \*const parent=nullptr)
- void [moreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription) override
- virtual bool **querySendNotifyEvent** () const
- void [save](#) (const [DImg](#) &image, const QString &filePath, const QString &format)
- void [savingProgress](#) (const QString &filePath, float progress) override
- void **setNotificationPolicy** ([NotificationPolicy](#) notificationPolicy)
- virtual void **taskHasFinished** ()
- void [thumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const QImage &img) override
- [~LoadSaveThread](#) () override

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override



### Static Public Member Functions

- static int [exifOrientation](#) (const QString &filePath, const [DMetadata](#) &metadata, bool isRaw, bool fromRaw↔ EmbeddedPreview)
- static [LoadSaveFileInfoProvider](#) \* [infoProvider](#) ()
- static void [setInfoProvider](#) ([LoadSaveFileInfoProvider](#) \*const infoProvider)

### Protected Member Functions

- void [notificationReceived](#) ()
- void [run](#) () override

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

### Protected Attributes

- [LoadSaveTask](#) \* [m\\_currentTask](#) = nullptr
- QMutex [m\\_mutex](#)
- [NotificationPolicy](#) [m\\_notificationPolicy](#) = [NotificationPolicyTimeLimited](#)
- QList< [LoadSaveTask](#) \* > [m\\_todo](#)

### Additional Inherited Members

### Public Slots inherited from [Digikam::DynamicThread](#)

- void [start](#) ()
- void [stop](#) ()
- void [wait](#) ()

## 6.1029.1 Member Enumeration Documentation

### 6.1029.1.1 AccessMode

enum [Digikam::LoadSaveThread::AccessMode](#)

used by [SharedLoadSaveThread](#) only

#### Enumerator

<a href="#">AccessModeRead</a>	image will only be used for reading
<a href="#">AccessModeReadWrite</a>	image data will possibly be changed

### 6.1029.1.2 NotificationPolicy

enum `Digikam::LoadSaveThread::NotificationPolicy`

#### Enumerator

<code>NotificationPolicyDirect</code>	Always send notification, unless the last event is still in the event queue
<code>NotificationPolicyTimeLimited</code>	Always wait for a certain amount of time after the last event sent. In particular, the first event will be sent only after waiting for this time span. (Or no event will be sent, when the loading has finished before) This is the default.

## 6.1029.2 Constructor & Destructor Documentation

### 6.1029.2.1 ~LoadSaveThread()

`Digikam::LoadSaveThread::~~LoadSaveThread ( ) [override]`

Destructor: The thread will execute all pending tasks and wait for this upon destruction

## 6.1029.3 Member Function Documentation

### 6.1029.3.1 exifOrientation()

```
int Digikam::LoadSaveThread::exifOrientation (
    const QString & filePath,
    const DMetadata & metadata,
    bool isRaw,
    bool fromRawEmbeddedPreview ) [static]
```

Retrieves the Exif orientation, either from the info provider if available, or from the metadata

### 6.1029.3.2 imageLoaded()

```
void Digikam::LoadSaveThread::imageLoaded (
    const LoadingDescription & loadingDescription,
    const DImg & img ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

### 6.1029.3.3 imageSaved()

```
void Digikam::LoadSaveThread::imageSaved (
    const QString & filePath,
    bool success ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

#### 6.1029.3.4 imageStartedLoading()

```
void Digikam::LoadSaveThread::imageStartedLoading (
    const LoadingDescription & loadingDescription ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

#### 6.1029.3.5 imageStartedSaving()

```
void Digikam::LoadSaveThread::imageStartedSaving (
    const QString & filePath ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

#### 6.1029.3.6 load()

```
void Digikam::LoadSaveThread::load (
    const LoadingDescription & description )
```

Append a task to load the given file to the task list

#### 6.1029.3.7 loadingProgress()

```
void Digikam::LoadSaveThread::loadingProgress (
    const LoadingDescription & loadingDescription,
    float progress ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

#### 6.1029.3.8 moreCompleteLoadingAvailable()

```
void Digikam::LoadSaveThread::moreCompleteLoadingAvailable (
    const LoadingDescription & oldLoadingDescription,
    const LoadingDescription & newLoadingDescription ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

#### 6.1029.3.9 run()

```
void Digikam::LoadSaveThread::run ( ) [override], [protected], [virtual]
```

Implement this pure virtual function in your subclass.

Implements [Digikam::DynamicThread](#).

### 6.1029.3.10 save()

```
void Digikam::LoadSaveThread::save (
    const QImage & image,
    const QString & filePath,
    const QString & format )
```

Append a task to save the image to the task list

### 6.1029.3.11 savingProgress()

```
void Digikam::LoadSaveThread::savingProgress (
    const QString & filePath,
    float progress ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

### 6.1029.3.12 signalImageLoaded

```
void Digikam::LoadSaveThread::signalImageLoaded (
    const LoadingDescription & loadingDescription,
    const QImage & img ) [signal]
```

This signal is emitted when the loading process has finished. If the process failed, img is null.

### 6.1029.3.13 signalImageStartedLoading

```
void Digikam::LoadSaveThread::signalImageStartedLoading (
    const LoadingDescription & loadingDescription ) [signal]
```

All signals are delivered to the thread from where the [LoadSaveThread](#) object has been created. This thread must use its event loop to get the signals. You must connect to these signals with `Qt::AutoConnection` (default) or `Qt::QueuedConnection`. This signal is emitted when the loading process begins.

### 6.1029.3.14 signalLoadingProgress

```
void Digikam::LoadSaveThread::signalLoadingProgress (
    const LoadingDescription & loadingDescription,
    float progress ) [signal]
```

This signal is emitted whenever new progress info is available and the notification policy allows emitting the signal. No progress info will be sent for preloaded images ([ManagedLoadSaveThread](#)).

### 6.1029.3.15 signalMoreCompleteLoadingAvailable

```
void Digikam::LoadSaveThread::signalMoreCompleteLoadingAvailable (
    const LoadingDescription & oldLoadingDescription,
    const LoadingDescription & newLoadingDescription ) [signal]
```

This signal is emitted if

- you are doing shared loading ([SharedLoadSaveThread](#))
- you started a loading operation with a [LoadingDescription](#) for a reduced version of the image
- another thread started a loading operation for a more complete version You may want to cancel the current operation and start with the given loadingDescription

### 6.1029.3.16 thumbnailLoaded()

```
void Digikam::LoadSaveThread::thumbnailLoaded (
    const LoadingDescription & loadingDescription,
    const QImage & img ) [override], [virtual]
```

Implements [Digikam::LoadSaveNotifier](#).

Reimplemented in [Digikam::ThumbnailLoadThread](#).

## 6.1030 Digikam::LocalContrastContainer Class Reference

### Public Member Functions

- double **getBlur** (int nstage) const
- double **getPower** (int nstage) const

### Public Attributes

- int **functionId** = 0
- int **highSaturation** = 100
- int **lowSaturation** = 100
- struct {
  - double **blur** = 80.0
  - bool **enabled** = false
  - double **power** = 30.0 } **stage** [TONEMAPPING\_MAX\_STAGES]
- bool **stretchContrast** = true

## 6.1031 Digikam::LocalContrastFilter Class Reference

Inheritance diagram for Digikam::LocalContrastFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **LocalContrastFilter** (*DImg* \*const image, *QObject* \*const parent=nullptr, const [LocalContrastContainer](#) &par=[LocalContrastContainer](#)())
- **LocalContrastFilter** (*QObject* \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) (*DImg* \*const orgImage, *QObject* \*const parent, const *QString* &name=*QString*())
- [DImgThreadedFilter](#) (*QObject* \*const parent=nullptr, const *QString* &name=*QString*())
- const *QString* & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- *QList*< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual *QString* **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const *QString* &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual *QList*< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (*QObject* \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- *QThread::Priority* **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (*QThread::Priority* priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static *QString* **DisplayableName** ()
- static *QString* **FilterIdentifier** ()
- static *QList*< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false



## 6.1031.1 Member Function Documentation

### 6.1031.1.1 filterAction()

`FilterAction` Digikam::LocalContrastFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1031.1.2 filterIdentifier()

`QString` Digikam::LocalContrastFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

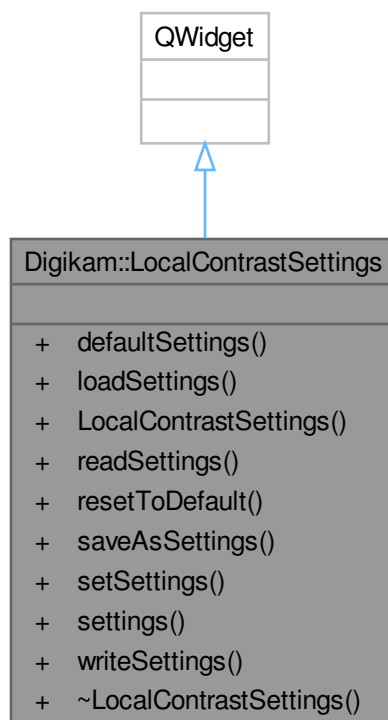
### 6.1031.1.3 readParameters()

`void` Digikam::LocalContrastFilter::readParameters (   
 const `FilterAction` & *action* ) [override], [virtual]

Implements [Digikam::DImgThreadedFilter](#).

## 6.1032 Digikam::LocalContrastSettings Class Reference

Inheritance diagram for Digikam::LocalContrastSettings:



## Signals

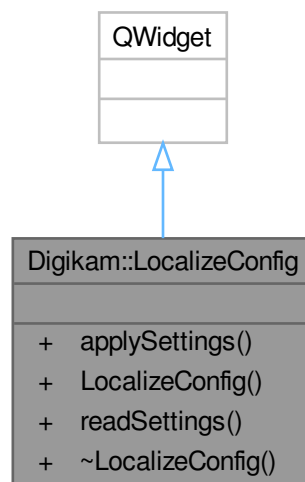
- void **signalSettingsChanged** ()

## Public Member Functions

- [LocalContrastContainer](#) **defaultSettings** () const
- void **loadSettings** ()
- **LocalContrastSettings** (QWidget \*const parent)
- void **readSettings** (KConfigGroup &group)
- void **resetToDefault** ()
- void **saveAsSettings** ()
- void **setSettings** (const [LocalContrastContainer](#) &settings)
- [LocalContrastContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.1033 Digikam::LocalizeConfig Class Reference

Inheritance diagram for Digikam::LocalizeConfig:



## Public Member Functions

- void **applySettings** ()
- **LocalizeConfig** (QWidget \*const parent=nullptr)
- void **readSettings** ()

## 6.1034 Digikam::LocalizeContainer Class Reference

### Public Member Functions

- void **readFromConfig** (const KConfigGroup &group)
- void **writeToConfig** (KConfigGroup &group) const

### Public Attributes

- QStringList **alternativeLang**  
*List of langges to use with Alternative Languages Text editor.*
- QString **defaultLanguage**
- bool **enableSpellCheck** = false  
*Enable spell-checking feature.*
- QStringList **ignoredWords**  
*Default language code to use with x-default (empty for auto-detection).*
- [DOnlineTranslator::Engine](#) **translatorEngine** = [DOnlineTranslator::Google](#)  
*Online translator to use.*
- QStringList **translatorLang**  
*List of langues to use with Online translator.*

### 6.1034.1 Detailed Description

The class [LocalizeContainer](#) encapsulates all spell-check and localize related settings.

### 6.1034.2 Member Data Documentation

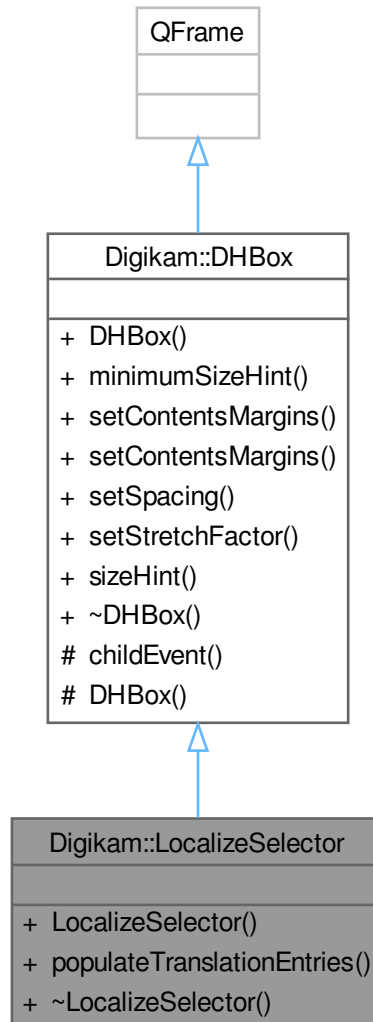
#### 6.1034.2.1 ignoredWords

```
QStringList Digikam::LocalizeContainer::ignoredWords
```

Words to ignore with spell-checking.

## 6.1035 Digikam::LocalizeSelector Class Reference

Inheritance diagram for Digikam::LocalizeSelector:



### Signals

- void **signalTranslate** (const QString &lang)

### Public Member Functions

- **LocalizeSelector** (QWidget \*const parent)
- void **populateTranslationEntries** ()

## Public Member Functions inherited from Digikam::DHBox

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentMargins** (const QMargins &argins)
- void **setContentMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

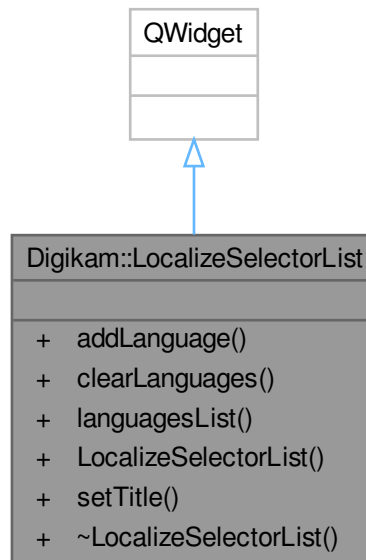
## Additional Inherited Members

## Protected Member Functions inherited from Digikam::DHBox

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1036 Digikam::LocalizeSelectorList Class Reference

Inheritance diagram for Digikam::LocalizeSelectorList:



## Signals

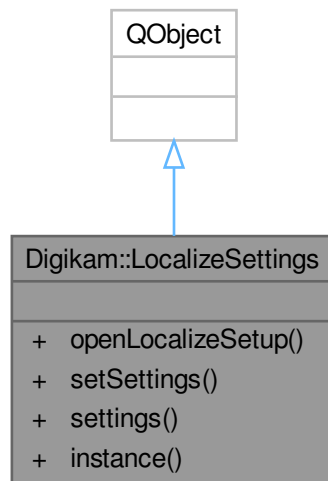
- void **signalSettingsChanged** ()

### Public Member Functions

- void **addLanguage** (const QString &code)
- void **clearLanguages** ()
- QStringList **languagesList** () const
- **LocalizeSelectorList** (QWidget \*const parent)
- void **setTitle** (const QString &title)

## 6.1037 Digikam::LocalizeSettings Class Reference

Inheritance diagram for Digikam::LocalizeSettings:



### Public Types

- enum **ConfigPart** { **LocalizeConfig** , **SpellCheckConfig** , **AllConfig** }

### Signals

- void **signalOpenLocalizeSetup** ()
- void **signalSettingsChanged** ()

### Public Member Functions

- void **openLocalizeSetup** ()
- void **setSettings** (const [LocalizeContainer](#) &settings, ConfigPart config)
- [LocalizeContainer](#) **settings** () const

## Static Public Member Functions

- static [LocalizeSettings](#) \* [instance](#) ()

## Friends

- class [LocalizeSettingsCreator](#)

## 6.1037.1 Member Function Documentation

### 6.1037.1.1 [instance\(\)](#)

```
LocalizeSettings * Digikam::LocalizeSettings::instance ( ) [static]
```

Global container for spell-check and localize settings. All accessor methods are thread-safe.

### 6.1037.1.2 [setSettings\(\)](#)

```
void Digikam::LocalizeSettings::setSettings (
    const LocalizeContainer & settings,
    ConfigPart config )
```

Sets the current Metadata settings and writes them to config.

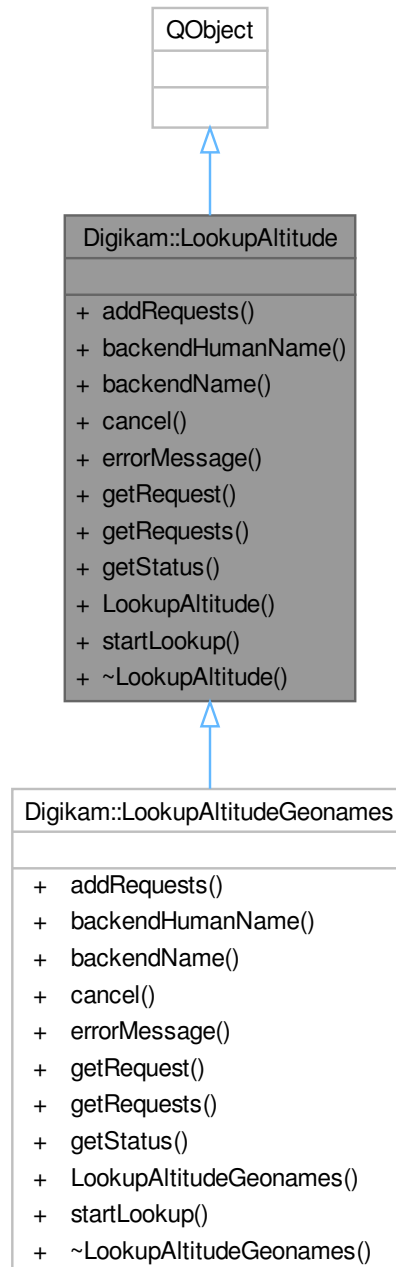
### 6.1037.1.3 [settings\(\)](#)

```
LocalizeContainer Digikam::LocalizeSettings::settings ( ) const
```

Returns the current Metadata settings.

## 6.1038 Digikam::LookupAltitude Class Reference

Inheritance diagram for Digikam::LookupAltitude:



### Classes

- class [Request](#)



## Public Types

- enum **StatusEnum** { **StatusInProgress** = 0 , **StatusSuccess** = 1 , **StatusCanceled** = 2 , **StatusError** = 3 }

## Signals

- void **signalDone** ()
- void **signalRequestsReady** (const QList< int > &readyRequests)

## Public Member Functions

- virtual void **addRequests** (const Request::List &requests)=0
- virtual QString **backendHumanName** () const =0
- virtual QString **backendName** () const =0
- virtual void **cancel** ()=0
- virtual QString **errorMessage** () const =0
- virtual [Request](#) **getRequest** (const int index) const =0
- virtual Request::List **getRequests** () const =0
- virtual StatusAltitude **getStatus** () const =0
- **LookupAltitude** (QObject \*const parent)
- virtual void **startLookup** ()=0

## 6.1039 Digikam::LookupAltitude::Request Class Reference

### Public Types

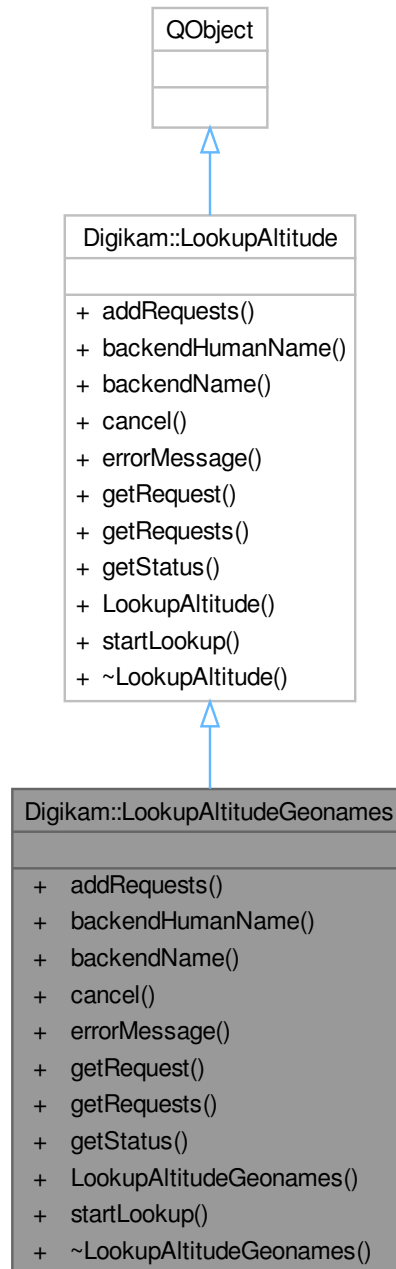
- typedef QList< [Request](#) > **List**

### Public Attributes

- [GeoCoordinates](#) **coordinates**
- QVariant **data**
- bool **success** = false

## 6.1040 Digikam::LookupAltitudeGeonames Class Reference

Inheritance diagram for Digikam::LookupAltitudeGeonames:



### Public Member Functions

- void **addRequests** (const Request::List &requests) override
- QString **backendHumanName** () const override

- QString [backendName](#) () const override
- void [cancel](#) () override
- QString [errorMessage](#) () const override
- Request [getRequest](#) (const int index) const override
- Request::List [getRequests](#) () const override
- StatusAltitude [getStatus](#) () const override
- **LookupAltitudeGeonames** (QObject \*const parent)
- void [startLookup](#) () override

## Public Member Functions inherited from [Digikam::LookupAltitude](#)

- virtual void [addRequests](#) (const Request::List &requests)=0
- **LookupAltitude** (QObject \*const parent)

## Additional Inherited Members

## Public Types inherited from [Digikam::LookupAltitude](#)

- enum **StatusEnum** { **StatusInProgress** = 0 , **StatusSuccess** = 1 , **StatusCanceled** = 2 , **StatusError** = 3 }

## Signals inherited from [Digikam::LookupAltitude](#)

- void [signalDone](#) ()
- void [signalRequestsReady](#) (const QList< int > &readyRequests)

## 6.1040.1 Member Function Documentation

### 6.1040.1.1 backendHumanName()

QString Digikam::LookupAltitudeGeonames::backendHumanName ( ) const [override], [virtual]

Implements [Digikam::LookupAltitude](#).

### 6.1040.1.2 backendName()

QString Digikam::LookupAltitudeGeonames::backendName ( ) const [override], [virtual]

Implements [Digikam::LookupAltitude](#).

### 6.1040.1.3 cancel()

void Digikam::LookupAltitudeGeonames::cancel ( ) [override], [virtual]

Implements [Digikam::LookupAltitude](#).

#### 6.1040.1.4 errorMessage()

```
QString Digikam::LookupAltitudeGeonames::errorMessage ( ) const [override], [virtual]
```

Implements [Digikam::LookupAltitude](#).

#### 6.1040.1.5 getRequest()

```
LookupAltitude::Request Digikam::LookupAltitudeGeonames::getRequest (
    const int index ) const [override], [virtual]
```

Implements [Digikam::LookupAltitude](#).

#### 6.1040.1.6 getRequests()

```
LookupAltitude::Request::List Digikam::LookupAltitudeGeonames::getRequests ( ) const [override],
[virtual]
```

Implements [Digikam::LookupAltitude](#).

#### 6.1040.1.7 getStatus()

```
LookupAltitude::StatusAltitude Digikam::LookupAltitudeGeonames::getStatus ( ) const [override],
[virtual]
```

Implements [Digikam::LookupAltitude](#).

#### 6.1040.1.8 startLookup()

```
void Digikam::LookupAltitudeGeonames::startLookup ( ) [override], [virtual]
```

Implements [Digikam::LookupAltitude](#).

## 6.1041 Digikam::LookupFactory Class Reference

### Static Public Member Functions

- static [LookupAltitude](#) \* **getAltitudeLookup** (const QString &backendName, QObject \*const parent)

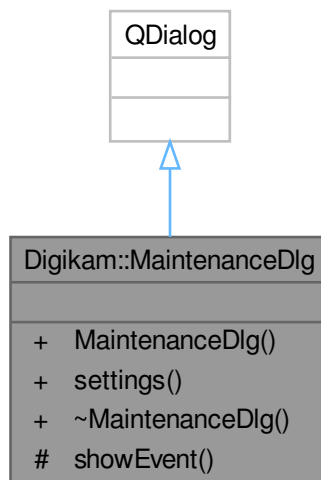
## 6.1042 Digikam::MaintenanceData Class Reference

### Public Member Functions

- [Identity](#) `getIdentity ()` const
- `qulonglong` `getImageld ()` const
- `QString` `getImagePath ()` const
- [ItemInfo](#) `getItemInfo ()` const
- `qulonglong` `getSimilarityImageld ()` const
- `int` `getThumbnailId ()` const
- `void` `setIdentities (const QList< Identity > &identities)`
- `void` `setImagelds (const QList< qulonglong > &ids)`
- `void` `setImagePaths (const QList< QString > &paths)`
- `void` `setItemInfos (const QList< ItemInfo > &infos)`
- `void` `setSimilarityImagelds (const QList< qulonglong > &ids)`
- `void` `setThumbnailIds (const QList< int > &ids)`

## 6.1043 Digikam::MaintenanceDlg Class Reference

Inheritance diagram for Digikam::MaintenanceDlg:



### Classes

- class [Private](#)

### Public Member Functions

- `MaintenanceDlg` (`QWidget *const parent=nullptr`)
- `MaintenanceSettings` `settings ()` const

## Protected Member Functions

- void **showEvent** (QShowEvent \*) override

## 6.1044 Digikam::MaintenanceDlg::Private Class Reference

### Public Types

- enum **Operation** {  
**Options** = 0 , **NewItems** , **DbCleanup** , **Thumbnails** ,  
**FingerPrints** , **Duplicates** , **FaceManagement** , **AutotagsAssignment** ,  
**ImageQualitySorter** , **MetadataSync** , **Stretch** }

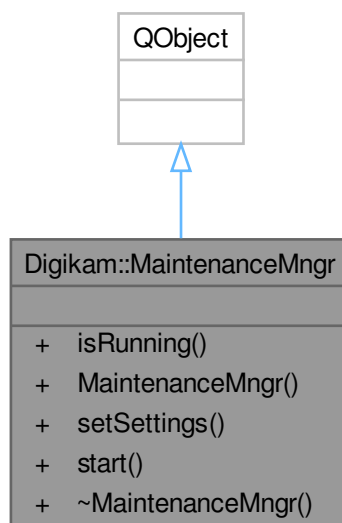
### Public Attributes

- [AlbumSelectors](#) \* **albumSelectors** = nullptr
- QComboBox \* **autotaggingScanMode** = nullptr
- QDialogButtonBox \* **buttons** = nullptr
- QCheckBox \* **cleanFacesDb** = nullptr
- QCheckBox \* **cleanSimilarityDb** = nullptr
- QCheckBox \* **cleanThumbsDb** = nullptr
- const QString **configAutotaggingScanMode** = QLatin1String("AutotaggingScanMode")
- const QString **configAutotagsAssignment** = QLatin1String("AutotagsAssignment")
- const QString **configAutotagsLanguages** = QLatin1String("AutotagsLanguages")
- const QString **configCleanupDatabase** = QLatin1String("CleanupDatabase")
- const QString **configCleanupFacesDatabase** = QLatin1String("CleanupFacesDatabase")
- const QString **configCleanupSimilarityDatabase** = QLatin1String("CleanupSimilarityDatabase")
- const QString **configCleanupThumbDatabase** = QLatin1String("CleanupThumbDatabase")
- const QString **configDuplicates** = QLatin1String("Duplicates")
- const QString **configDuplicatesRestriction** = QLatin1String("duplicatesRestriction")
- const QString **configFaceManagement** = QLatin1String("FaceManagement")
- const QString **configFaceScannedHandling** = QLatin1String("FaceScannedHandling")
- const QString **configFingerPrints** = QLatin1String("FingerPrints")
- const QString **configGroupName** = QLatin1String("MaintenanceDlg Settings")
- const QString **configImageQualitySorter** = QLatin1String("ImageQualitySorter")
- const QString **configMaxSimilarity** = QLatin1String("maxSimilarity")
- const QString **configMetadataSync** = QLatin1String("MetadataSync")
- const QString **configMinSimilarity** = QLatin1String("minSimilarity")
- const QString **configModelSelectionMode** = QLatin1String("ModelSelectionMode")
- const QString **configNewItems** = QLatin1String("NewItems")
- const QString **configQualityScanMode** = QLatin1String("QualityScanMode")
- const QString **configQualitySettingsSelected** = QLatin1String("QualitySettingsSelected")
- const QString **configScanFingerPrints** = QLatin1String("ScanFingerPrints")
- const QString **configScanThumbs** = QLatin1String("ScanThumbs")
- const QString **configShrinkDatabases** = QLatin1String("ShrinkDatabases")
- const QString **configSyncDirection** = QLatin1String("SyncDirection")
- const QString **configThumbnails** = QLatin1String("Thumbnails")
- const QString **configUseLastSettings** = QLatin1String("UseLastSettings")
- const QString **configUseMutiCoreCPU** = QLatin1String("UseMutiCoreCPU")
- [DHBox](#) \* **dupeRestrictionBox** = nullptr
- [DVBox](#) \* **duplicatesBox** = nullptr
- [DExpanderBox](#) \* **expanderBox** = nullptr

- QComboBox \* **faceScannedHandling** = nullptr
- QLabel \* **logo** = nullptr
- QPushButton \* **metadataSetup** = nullptr
- QComboBox \* **modelSelectionMode** = nullptr
- QComboBox \* **qualityScanMode** = nullptr
- [ImageQualityConfSelector](#) \* **qualitySelector** = nullptr
- QCheckBox \* **resetFaceDb** = nullptr
- QCheckBox \* **retrainAllFaces** = nullptr
- QCheckBox \* **scanFingerPrints** = nullptr
- QCheckBox \* **scanThumbs** = nullptr
- QComboBox \* **searchResultRestriction** = nullptr
- QCheckBox \* **shrinkDatabases** = nullptr
- [DIntRangeBox](#) \* **similarityRange** = nullptr
- [DHBox](#) \* **similarityRangeBox** = nullptr
- QComboBox \* **syncDirection** = nullptr
- QLabel \* **title** = nullptr
- [LocalizeSelectorList](#) \* **trSelectorList** = nullptr
- QCheckBox \* **useLastSettings** = nullptr
- QCheckBox \* **useMutiCoreCPU** = nullptr
- [DVBox](#) \* **vbox** = nullptr
- [DVBox](#) \* **vbox2** = nullptr
- [DVBox](#) \* **vbox3** = nullptr
- [DVBox](#) \* **vbox4** = nullptr
- [DVBox](#) \* **vbox5** = nullptr

## 6.1045 Digikam::MaintenanceMngr Class Reference

Inheritance diagram for Digikam::MaintenanceMngr:



## Signals

- void **signalComplete** ()

## Public Member Functions

- bool **isRunning** () const
- **MaintenanceMngr** (QObject \*const parent)
- void **setSettings** (const [MaintenanceSettings](#) &settings)
- void **start** ()

## 6.1046 Digikam::MaintenanceSettings Class Reference

### Public Attributes

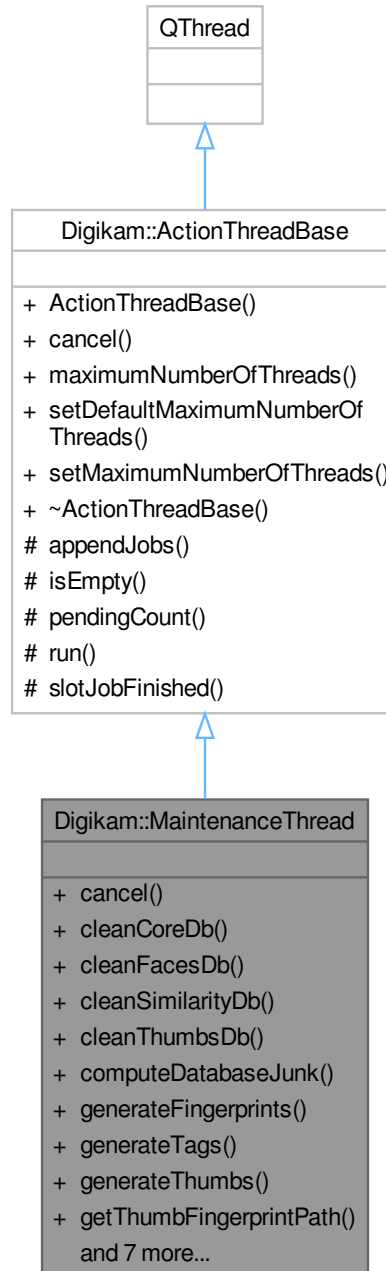
- AlbumList **albums**
- int **autotaggingScanMode** = [AutotagsAssignment::AllItems](#)  
*autotagging scan mode*
- bool **autotagsAssignment** = false  
*Autotags assignment.*
- QStringList **autotagsLanguages**  
*Autotags languages.*
- bool **cleanFacesDb** = false
- bool **cleanSimilarityDb** = false
- bool **cleanThumbDb** = false
- bool **databaseCleanup** = false  
*Perform database cleanup.*
- bool **duplicates** = false  
*Scan for new items.*
- Haarlfacce::DuplicatesSearchRestrictions **duplicatesRestriction** = Haarlfacce::DuplicatesSearchRestrictions←  
::None  
*The type of restrictions to apply on duplicates search results.*
- bool **faceManagement** = false  
*Scan for faces.*
- [FaceScanSettings](#) **faceSettings**  
*Face detection settings.*
- bool **fingerPrints** = false  
*Generate finger-prints.*
- int **maxSimilarity** = 100  
*Maximal similarity between items to compare, in percents.*
- bool **metadataSync** = false  
*Sync metadata and DB.*
- int **minSimilarity** = 90  
*Minimal similarity between items to compare, in percents.*
- int **modelSelectionMode** = [DetectorModel::YOLOV5NANO](#)  
*model selection mode*
- bool **newItems** = false  
*Find new items on whole collection.*
- [ImageQualityContainer](#) **quality**  
*Image Quality Sorting Settings.*



- int **qualityScanMode** = [ImageQualitySorter::AllItems](#)  
*Mode to assign Pick Labels to items. NOTE: turn all items by default to prevent clearing whole Pick Labels from Collection.*
- int **qualitySettingsSelected** = [ImageQualityConfSelector::GlobalSettings](#)  
*Type of quality settings selected.*
- bool **qualitySort** = false  
*Perform Image Quality Sorting.*
- bool **scanFingerPrints** = false  
*Rebuild all fingerprints or only scan missing items.*
- bool **scanThumbs** = false  
*Rebuild all thumbnails or only scan missing items.*
- bool **shrinkDatabases** = false
- int **syncDirection** = [MetadataSynchronizer::WriteFromDatabaseToFile](#)  
*Sync direction (image metadata <-> DB).*
- AlbumList **tags**
- bool **thumbnails** = false  
*Generate thumbnails.*
- bool **useMutiCoreCPU** = false  
*Use Multi-core CPU to process items.*
- bool **wholeAlbums** = true
- bool **wholeTags** = true

## 6.1047 Digikam::MaintenanceThread Class Reference

Inheritance diagram for Digikam::MaintenanceThread:



### Signals

- void [signalAddItemsToProcess](#) (int count)
- void [signalAdvance](#) ()

- void **signalAdvance** (const [ItemInfo](#) &, const QImage &)
- void **signalAdvance** (const [ItemInfo](#) &, const QImage &, const QStringList &)
- void [signalAdvance](#) (const [ItemInfo](#) &, const QImage &, int)
- void **signalAdvance** (const QImage &)
- void [signalCanceled](#) ()
- void [signalCompleted](#) ()
- void [signalData](#) (const QList< qlonglong > &staleImagelds, const QList< int > &staleThumblds, const QList< [Identity](#) > &staleIdentities, const QList< qlonglong > &staleSimilarityImagelds)
- void [signalFinished](#) (bool done, bool errorFree)
- void [signalRemovePending](#) (const [ItemInfo](#) &info)
- void [signalStarted](#) ()

### Public Member Functions

- void **cancel** ()
- void **cleanCoreDb** (const QList< qlonglong > &imagelds)
- void **cleanFacesDb** (const QList< [Identity](#) > &staleIdentities)
- void **cleanSimilarityDb** (const QList< qlonglong > &imagelds)
- void **cleanThumbsDb** (const QList< int > &thumbnaillds)
- void **computeDatabaseJunk** (bool thumbsDb=false, bool facesDb=false, bool similarityDb=false)
- void **generateFingerprints** (const QList< qlonglong > &itemIds, bool rebuildAll)
- void **generateTags** (const QStringList &paths, int modelType, const QStringList &langs)
- void **generateThumbs** (const QStringList &paths)
- QString **getThumbFingerprintPath** ()
- **MaintenanceThread** (QObject \*const parent)
- void **removeMetadata** (const [ItemInfoList](#) &items, MetadataRemover::RemoveAction action)
- void **setUseMultiCore** (const bool b)
- void **shrinkDatabases** ()
- void **sortByImageQuality** (const QStringList &paths, const [ImageQualityContainer](#) &quality)
- void **syncMetadata** (const [ItemInfoList](#) &items, MetadataSynchronizer::SyncDirection dir, bool tagsOnly)

### Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void [cancel](#) (bool isCancel=true)
- int [maximumNumberOfThreads](#) () const
- void [setDefaultMaximumNumberOfThreads](#) ()
- void [setMaximumNumberOfThreads](#) (int n)

### Additional Inherited Members

### Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void [slotJobFinished](#) ()

### Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void [appendJobs](#) (const [ActionJobCollection](#) &jobs)
- bool [isEmpty](#) () const
- int [pendingCount](#) () const
- void [run](#) () override

## 6.1047.1 Member Function Documentation

### 6.1047.1.1 signalAddItemsToProcess

```
void Digikam::MaintenanceThread::signalAddItemsToProcess (
    int count ) [signal]
```

Signal to emit the count of additional items to process.

### 6.1047.1.2 signalAdvance [1/2]

```
void Digikam::MaintenanceThread::signalAdvance ( ) [signal]
```

Emit when an item was processed and on additional information is necessary.

### 6.1047.1.3 signalAdvance [2/2]

```
void Digikam::MaintenanceThread::signalAdvance (
    const ItemInfo & ,
    const QImage & ,
    int ) [signal]
```

Emit when an item have been processed. QImage can be used to pass item thumbnail processed.

### 6.1047.1.4 signalCanceled

```
void Digikam::MaintenanceThread::signalCanceled ( ) [signal]
```

Signal to emit to sub-tasks to cancel processing.

### 6.1047.1.5 signalCompleted

```
void Digikam::MaintenanceThread::signalCompleted ( ) [signal]
```

Emit when a items list have been fully processed.

### 6.1047.1.6 signalData

```
void Digikam::MaintenanceThread::signalData (
    const QList< qlonglong > & staleImageIds,
    const QList< int > & staleThumbIds,
    const QList< Identity > & staleIdentities,
    const QList< qlonglong > & staleSimilarityImageIds ) [signal]
```

Signal to emit junk data for db cleaner.

### 6.1047.1.7 signalFinished

```
void Digikam::MaintenanceThread::signalFinished (
    bool done,
    bool errorFree ) [signal]
```

Signal to emit after processing with info if the processing was done and if yes, without errors.

### 6.1047.1.8 signalRemovePending

```
void Digikam::MaintenanceThread::signalRemovePending (
    const ItemInfo & info ) [signal]
```

Signal to remove pending item from lazy sync.

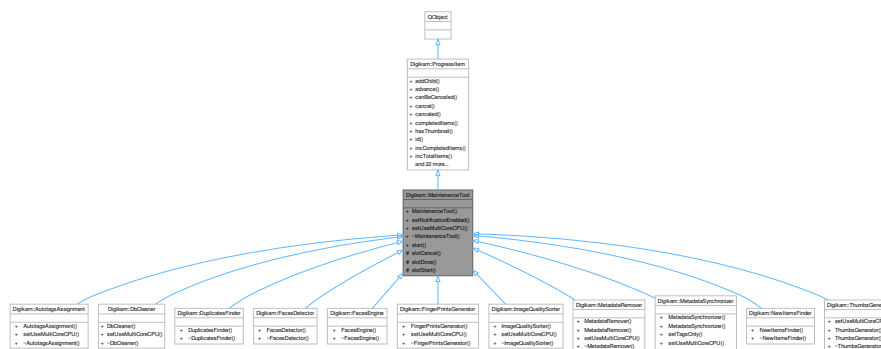
### 6.1047.1.9 signalStarted

```
void Digikam::MaintenanceThread::signalStarted ( ) [signal]
```

Emit when the task has started it's work.

## 6.1048 Digikam::MaintenanceTool Class Reference

Inheritance diagram for Digikam::MaintenanceTool:



### Public Slots

- void **start** ()

### Signals

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

## Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void [progressItemCanceledById](#) (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- [MaintenanceTool](#) (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)
- virtual void [setUseMultiCoreCPU](#) (bool)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void [addChild](#) ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)  
*Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void [cancel](#) ()
- bool [canceled](#) () const
- unsigned int [completedItems](#) () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool [incCompletedItems](#) (unsigned int v=1)
- void [incTotalItems](#) (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool canBeCanceled, bool hasThumb)
- void [removeChild](#) ([ProgressItem](#) \*const kiddo)
- void [reset](#) ()

*Reset the progress value of this item to 0 and the status string to the empty string.*

- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)
- void **setProgress** (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void **setShowAtStart** (bool showAtStart)
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void **setStatus** (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void **setThumbnail** (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void **setUsesBusyIndicator** (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool **showAtStart** () const
- const QString & **status** () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool **usesBusyIndicator** () const

### Protected Slots

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.1048.1 Member Function Documentation

### 6.1048.1.1 setNotificationEnabled()

```
void Digikam::MaintenanceTool::setNotificationEnabled (
    bool b )
```

If true, show a notification message on desktop notification manager with time elapsed to run process.

### 6.1048.1.2 setUseMultiCoreCPU()

```
virtual void Digikam::MaintenanceTool::setUseMultiCoreCPU (
    bool ) [inline], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented in [Digikam::AutotagsAssignment](#), [Digikam::DbCleaner](#), [Digikam::FingerPrintsGenerator](#), [Digikam::ImageQualitySorter](#), [Digikam::MetadataRemover](#), [Digikam::MetadataSynchronizer](#), and [Digikam::ThumbsGenerator](#).

**6.1048.1.3 signalCanceled**

```
void Digikam::MaintenanceTool::signalCanceled ( ) [signal]
```

Emit when process is canceled.

**6.1048.1.4 signalComplete**

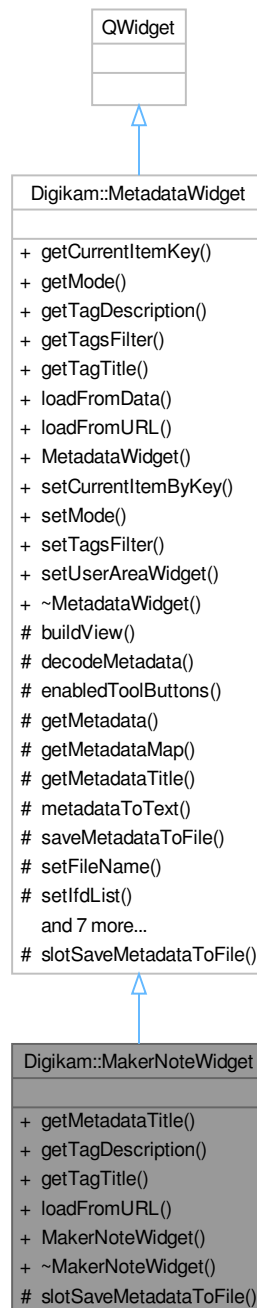
```
void Digikam::MaintenanceTool::signalComplete ( ) [signal]
```

Emit when process is done (not canceled).



## 6.1049 Digikam::MakerNoteWidget Class Reference

Inheritance diagram for Digikam::MakerNoteWidget:



### Public Member Functions

- QString [getMetadataTitle](#) () const override
- QString [getTagDescription](#) (const QString &key) override

- QString [getTagTitle](#) (const QString &key) override
- bool [loadFromURL](#) (const QUrl &url) override
- **MakerNoteWidget** (QWidget \*const parent, const QString &name=QString())

### Public Member Functions inherited from [Digikam::MetadataWidget](#)

- QString [getCurrentItemKey](#) () const
- int [getMode](#) () const
- QStringList [getTagsFilter](#) () const
- virtual bool [loadFromData](#) (const QString &fileName, const [DMetadata](#) &data=[DMetadata](#)())
- **MetadataWidget** (QWidget \*const parent, const QString &name=QString())
- void [setCurrentItemByKey](#) (const QString &itemKey)
- void [setMode](#) (int mode)
- void [setTagsFilter](#) (const QStringList &list)
- void [setUserAreaWidget](#) (QWidget \*const w)

### Protected Slots

- void [slotSaveMetadataToFile](#) () override

### Protected Slots inherited from [Digikam::MetadataWidget](#)

- virtual void [slotSaveMetadataToFile](#) ()=0

### Additional Inherited Members

### Public Types inherited from [Digikam::MetadataWidget](#)

- enum [TagFilters](#) { NONE = 0 , PHOTO , CUSTOM }

### Signals inherited from [Digikam::MetadataWidget](#)

- void [signalSetupMetadataFilters](#) ()

### Protected Member Functions inherited from [Digikam::MetadataWidget](#)

- void [enabledToolButtons](#) (bool)
- [DMetadata](#) \* [getMetadata](#) () const
- const [DMetadata::MetaDataMap](#) & [getMetadataMap](#) ()
- QString [metadataToText](#) () const
- QUrl [saveMetadataToFile](#) (const QString &caption, const QString &fileFilter)
- void [setFileName](#) (const QString &fileName)
- void [setIfdList](#) (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &keysFilter, const QStringList &tagsFilter)
- void [setIfdList](#) (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &tagsFilter=QStringList())
- bool [setMetadata](#) (const [DMetadata](#) &data=[DMetadata](#)())
- virtual void [setMetadataEmpty](#) ()
- void [setMetadataMap](#) (const [DMetadata::MetaDataMap](#) &data=[DMetadata::MetaDataMap](#)())
- void [setup](#) ()
- bool [storeMetadataToFile](#) (const QUrl &url, const QByteArray &metaData)
- [MetadataListView](#) \* [view](#) () const

## 6.1049.1 Member Function Documentation

### 6.1049.1.1 getMetadataTitle()

```
QString Digikam::MakerNoteWidget::getMetadataTitle ( ) const [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

### 6.1049.1.2 getTagDescription()

```
QString Digikam::MakerNoteWidget::getTagDescription (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

### 6.1049.1.3 getTagTitle()

```
QString Digikam::MakerNoteWidget::getTagTitle (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

### 6.1049.1.4 loadFromURL()

```
bool Digikam::MakerNoteWidget::loadFromURL (
    const QUrl & url ) [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

## 6.1050 Digikam::ManagedLoadSaveThread Class Reference

Inheritance diagram for Digikam::ManagedLoadSaveThread:



### Public Types

- enum [LoadingMode](#) { [LoadingModeNormal](#) , [LoadingModeShared](#) }

- enum [LoadingPolicy](#) { [LoadingPolicyFirstRemovePrevious](#) , [LoadingPolicyPrepend](#) , [LoadingPolicySimplePrepend](#) , [LoadingPolicyAppend](#) , [LoadingPolicySimpleAppend](#) , [LoadingPolicyPreload](#) }
- enum [LoadingTaskFilter](#) { [LoadingTaskFilterAll](#) , [LoadingTaskFilterPreloading](#) }
- enum [TerminationPolicy](#) { [TerminationPolicyTerminateLoading](#) , [TerminationPolicyTerminatePreloading](#) , [TerminationPolicyWait](#) , [TerminationPolicyTerminateAll](#) }

### Public Types inherited from [Digikam::LoadSaveThread](#)

- enum [AccessMode](#) { [AccessModeRead](#) , [AccessModeReadWrite](#) }
- enum [NotificationPolicy](#) { [NotificationPolicyDirect](#) , [NotificationPolicyTimeLimited](#) }

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

### Public Member Functions

- void [load](#) (const [LoadingDescription](#) &description)
- void [load](#) (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- [LoadingPolicy](#) [loadingPolicy](#) () const
- [ManagedLoadSaveThread](#) (QObject \*const parent=nullptr)
- void [save](#) (const [DImg](#) &image, const QString &filePath, const QString &format)
- void [setLoadingPolicy](#) ([LoadingPolicy](#) policy)
- void [setTerminationPolicy](#) ([TerminationPolicy](#) terminationPolicy)
- void [stopAllTasks](#) ()
- void [stopLoading](#) (const [LoadingDescription](#) &desc, [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopLoading](#) (const QString &filePath=QString(), [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopSaving](#) (const QString &filePath=QString())
- [TerminationPolicy](#) [terminationPolicy](#) () const

### Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- void [imageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img) override
- void [imageSaved](#) (const QString &filePath, bool success) override
- void [imageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription) override
- void [imageStartedSaving](#) (const QString &filePath) override
- void [load](#) (const [LoadingDescription](#) &description)
- void [loadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress) override
- [LoadSaveThread](#) (QObject \*const parent=nullptr)
- void [moreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription) override
- virtual bool [querySendNotifyEvent](#) () const
- void [save](#) (const [DImg](#) &image, const QString &filePath, const QString &format)
- void [savingProgress](#) (const QString &filePath, float progress) override
- void [setNotificationPolicy](#) ([NotificationPolicy](#) notificationPolicy)
- virtual void [taskHasFinished](#) ()
- void [thumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const QImage &img) override
- [~LoadSaveThread](#) () override

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Protected Member Functions

- void **load** (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void **load** (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [LoadingPolicy](#) policy, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void **loadPreview** (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- void **loadThumbnail** (const [LoadingDescription](#) &description)
- void **preloadThumbnail** (const [LoadingDescription](#) &description)
- void **preloadThumbnailGroup** (const QList< [LoadingDescription](#) > &descriptions)
- void **prependThumbnailGroup** (const QList< [LoadingDescription](#) > &descriptions)
- void **shutDown** ()

## Protected Member Functions inherited from [Digikam::LoadSaveThread](#)

- void **notificationReceived** ()
- void **run** () override

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes

- [LoadingPolicy](#) **m\_loadingPolicy** = [LoadingPolicyAppend](#)
- [TerminationPolicy](#) **m\_terminationPolicy** = [TerminationPolicyTerminateLoading](#)

## Protected Attributes inherited from [Digikam::LoadSaveThread](#)

- [LoadSaveTask](#) \* **m\_currentTask** = nullptr
- QMutex **m\_mutex**
- [NotificationPolicy](#) **m\_notificationPolicy** = [NotificationPolicyTimeLimited](#)
- QList< [LoadSaveTask](#) \* > **m\_todo**

## Additional Inherited Members

### Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

### Signals inherited from [Digikam::LoadSaveThread](#)

- void **signalImageLoaded** (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)
- void **signalImageSaved** (const QString &filePath, bool success)
- void **signalImageStartedLoading** (const [LoadingDescription](#) &loadingDescription)
- void **signalImageStartedSaving** (const QString &filePath)
- void **signalLoadingProgress** (const [LoadingDescription](#) &loadingDescription, float progress)
- void **signalMoreCompleteLoadingAvailable** (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription)
- void **signalSavingProgress** (const QString &filePath, float progress)
- void **signalThumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QImage &img)

### Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

### Static Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- static int **exifOrientation** (const QString &filePath, const [DMetadata](#) &metadata, bool isRaw, bool fromRaw↔EmbeddedPreview)
- static [LoadSaveFileInfoProvider](#) \* **infoProvider** ()
- static void **setInfoProvider** ([LoadSaveFileInfoProvider](#) \*const infoProvider)

## 6.1050.1 Member Enumeration Documentation

### 6.1050.1.1 LoadingMode

enum [Digikam::ManagedLoadSaveThread::LoadingMode](#)

used by [SharedLoadSaveThread](#) only

#### Enumerator

LoadingModeNormal	no sharing of loading process, no caching of image
LoadingModeShared	loading process is shared, image is cached

### 6.1050.1.2 LoadingPolicy

enum `Digikam::ManagedLoadSaveThread::LoadingPolicy`

#### Enumerator

<code>LoadingPolicyFirstRemovePrevious</code>	Load image immediately, remove and stop all previous loading tasks.
<code>LoadingPolicyPrepend</code>	Prepend loading in front of all other tasks, but wait for the current task to finish. No other tasks will be removed, preloading tasks will be stopped and postponed.
<code>LoadingPolicySimplePrepend</code>	Prepend in front of all other tasks (not touching the current task). Do not check for duplicate tasks, do not check for preloading tasks.
<code>LoadingPolicyAppend</code>	Append loading task to the end of the list, but in front of all preloading tasks. No other tasks will be removed, preloading tasks will be stopped and postponed. This is similar to the simple <code>load()</code> operation from <code>LoadSaveThread</code> , except for the special care taken for preloading.
<code>LoadingPolicySimpleAppend</code>	Append to the lists of tasks. Do not check for duplicate tasks, do not check for preloading tasks.
<code>LoadingPolicyPreload</code>	Preload image, i.e. load it with low priority when no other tasks are scheduled. All other tasks will take precedence, and preloading tasks will be stopped and postponed when another task is added. No progress info will be sent for preloaded images

### 6.1050.1.3 LoadingTaskFilter

enum `Digikam::ManagedLoadSaveThread::LoadingTaskFilter`

#### Enumerator

<code>LoadingTaskFilterAll</code>	filter all loading tasks
<code>LoadingTaskFilterPreloading</code>	filter only tasks with preloading policy

### 6.1050.1.4 TerminationPolicy

enum `Digikam::ManagedLoadSaveThread::TerminationPolicy`

#### Enumerator

<code>TerminationPolicyTerminateLoading</code>	Wait for saving tasks, stop and remove loading tasks This is the default.
<code>TerminationPolicyTerminatePreloading</code>	Wait for loading and saving tasks, stop and remove preloading tasks
<code>TerminationPolicyWait</code>	Wait for all pending tasks
<code>TerminationPolicyTerminateAll</code>	Stop all pending tasks



## 6.1050.2 Constructor & Destructor Documentation

### 6.1050.2.1 ManagedLoadSaveThread()

```
Digikam::ManagedLoadSaveThread::ManagedLoadSaveThread (
    QObject *const parent = nullptr ) [explicit]
```

Termination is controlled by setting the TerminationPolicy Default is TerminationPolicyTerminateLoading

## 6.1050.3 Member Function Documentation

### 6.1050.3.1 load()

```
void Digikam::ManagedLoadSaveThread::load (
    const LoadingDescription & description )
```

Append a task to load the given file to the task list. If there is already a task for the given file, it will possibly be rescheduled, but no second task will be added. Only loading tasks will - if required by the policy - be stopped or removed, saving tasks will not be touched.

### 6.1050.3.2 save()

```
void Digikam::ManagedLoadSaveThread::save (
    const DImg & image,
    const QString & filePath,
    const QString & format )
```

Append a task to save the image to the task list

### 6.1050.3.3 setLoadingPolicy()

```
void Digikam::ManagedLoadSaveThread::setLoadingPolicy (
    LoadingPolicy policy )
```

Set the loading policy. Default is LoadingPolicyAppend. You can override the default value for each operation.

### 6.1050.3.4 stopLoading() [1/2]

```
void Digikam::ManagedLoadSaveThread::stopLoading (
    const LoadingDescription & desc,
    LoadingTaskFilter filter = LoadingTaskFilterAll )
```

Same than previous method, but Stop and remove tasks filtered by LoadingDescription.

### 6.1050.3.5 stopLoading() [2/2]

```
void Digikam::ManagedLoadSaveThread::stopLoading (
    const QString & filePath = QString(),
    LoadingTaskFilter filter = LoadingTaskFilterAll )
```

Stop and remove tasks filtered by filePath and policy. If filePath isNull, applies to all file paths.

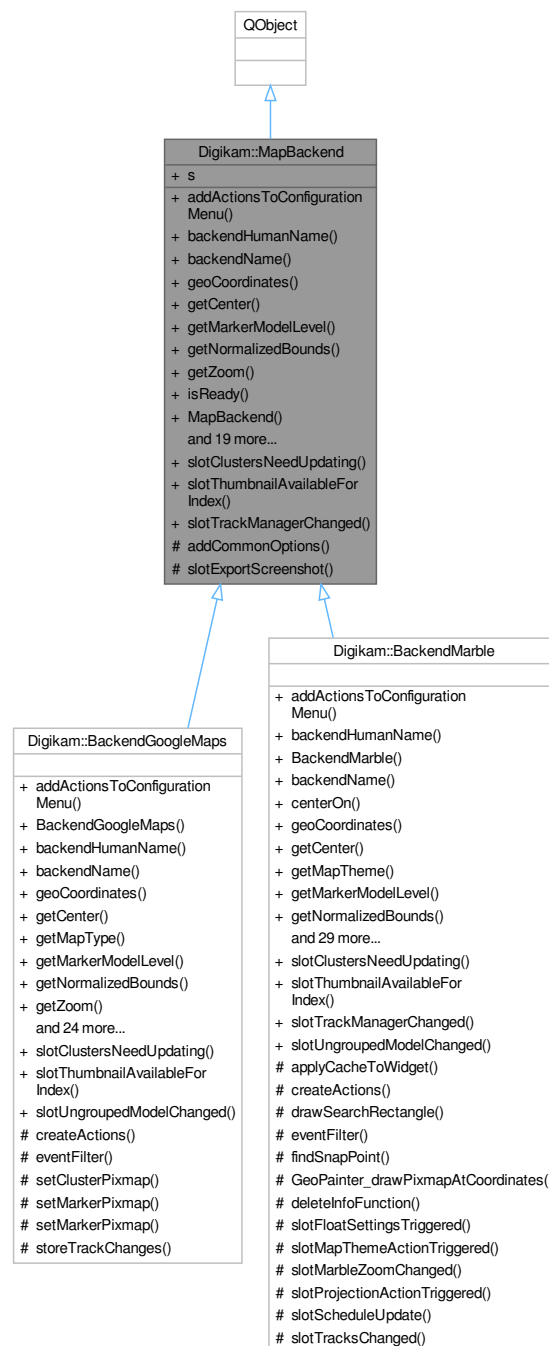
### 6.1050.3.6 stopSaving()

```
void Digikam::ManagedLoadSaveThread::stopSaving (
    const QString & filePath = QString() )
```

Stop and remove saving tasks filtered by filePath. If filePath isNull, applies to all file paths.

## 6.1051 Digikam::MapBackend Class Reference

Inheritance diagram for Digikam::MapBackend:



## Public Slots

- virtual void **slotClustersNeedUpdating** ()=0
- virtual void **slotThumbnailAvailableForIndex** (const QVariant &index, const QPixmap &pixmap)
- virtual void **slotTrackManagerChanged** ()

## Signals

- void **signalBackendReadyChanged** (const QString &backendName)
- void **signalClustersClicked** (const QList &clusterIndices)
- void **signalClustersMoved** (const QList &clusterIndices, const QPair< int, QModelIndex > &snapTarget)
- void **signalMarkersMoved** (const QList &markerIndices)
- void **signalSelectionHasBeenMade** (const Digikam::GeoCoordinates::Pair &coordinates)
- void **signalZoomChanged** (const QString &newZoom)

## Public Member Functions

- virtual void **addActionsToConfigurationMenu** (QMenu \*const configurationMenu)=0
- virtual QString **backendHumanName** () const =0
- virtual QString **backendName** () const =0
- virtual bool **geoCoordinates** (const QPoint &point, [GeoCoordinates](#) \*const coordinates) const =0
- virtual [GeoCoordinates](#) **getCenter** () const =0
- virtual int **getMarkerModelLevel** ()=0
- virtual [GeoCoordinates::PairList](#) **getNormalizedBounds** ()=0
- virtual QString **getZoom** () const =0
- virtual bool **isReady** () const =0
- **MapBackend** (const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > &sharedData, QObject \*const parent)
- virtual QSize **mapSize** () const =0
- virtual QWidget \* **mapWidget** ()=0
- virtual void **mapWidgetDocked** (const bool state)=0
- virtual void **mouseModeChanged** ()=0
- virtual void **readSettingsFromGroup** (const KConfigGroup \*const group)=0
- virtual void **regionSelectionChanged** ()=0
- virtual void **releaseWidget** ([GeofaceInternalWidgetInfo](#) \*const info)=0
- virtual void **reload** ()=0
- virtual void **saveSettingsToGroup** (KConfigGroup \*const group)=0
- virtual bool **screenCoordinates** (const [GeoCoordinates](#) &coordinates, QPoint \*const point)=0
- virtual void **setActive** (const bool state)=0
- virtual void **setCenter** (const [GeoCoordinates](#) &coordinate)=0
- virtual void **setZoom** (const QString &newZoom)=0
- virtual void **updateActionAvailability** ()=0
- virtual void **updateClusters** ()=0
- virtual void **updateMarkers** ()=0
- virtual void **zoomIn** ()=0
- virtual void **zoomOut** ()=0

## Public Attributes

- const QExplicitlySharedDataPointer< [GeofaceSharedData](#) > **s**

**Protected Slots**

- void **slotExportScreenshot** ()

**Protected Member Functions**

- void **addCommonOptions** (QMenu \*const configurationMenu)

**6.1051.1 Member Function Documentation****6.1051.1.1 mapWidget()**

```
virtual QWidget * Digikam::MapBackend::mapWidget ( ) [pure virtual]
```

Implemented in [Digikam::BackendMarble](#).

**6.1051.1.2 mouseModeChanged()**

```
virtual void Digikam::MapBackend::mouseModeChanged ( ) [pure virtual]
```

Implemented in [Digikam::BackendGoogleMaps](#).

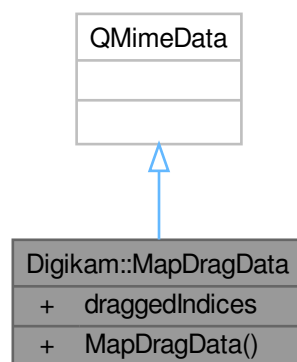
**6.1051.1.3 setActive()**

```
virtual void Digikam::MapBackend::setActive (
    const bool state ) [pure virtual]
```

Implemented in [Digikam::BackendGoogleMaps](#).

**6.1052 Digikam::MapDragData Class Reference**

Inheritance diagram for Digikam::MapDragData:

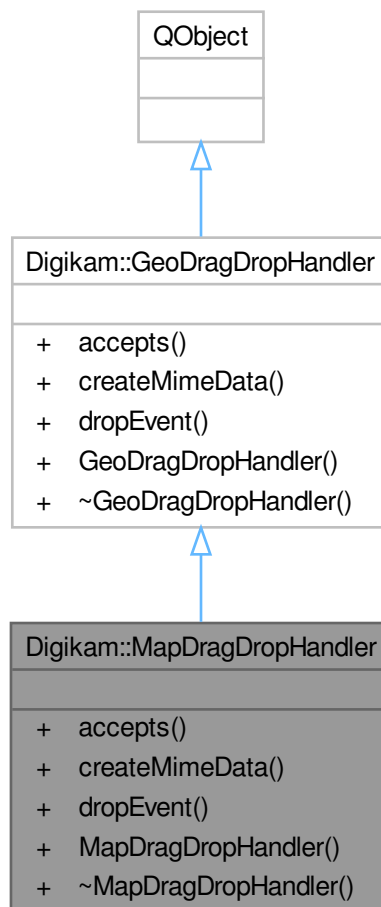


**Public Attributes**

- `QList< QPersistentModelIndex >` **draggedIndices**

**6.1053 Digikam::MapDragDropHandler Class Reference**

Inheritance diagram for Digikam::MapDragDropHandler:

**Public Member Functions**

- `Qt::DropAction` [accepts](#) (const `QDropEvent *e`) override
- `QMimeData *` [createMimeData](#) (const `QList< QPersistentModelIndex > &modelIndices`) override
- `bool` [dropEvent](#) (const `QDropEvent *e`, const `GeoCoordinates &dropCoordinates`) override
- **MapDragDropHandler** (`QAbstractItemModel *const`, `GPSGeofaceModelHelper *const parent`)

**Public Member Functions inherited from Digikam::GeoDragDropHandler**

- **GeoDragDropHandler** (`QObject *const parent=nullptr`)

## 6.1053.1 Member Function Documentation

### 6.1053.1.1 accepts()

```
Qt::DropAction Digikam::MapDragDropHandler::accepts (
    const QDropEvent * e ) [override], [virtual]
```

Implements [Digikam::GeoDragDropHandler](#).

### 6.1053.1.2 createMimeData()

```
QMimeData * Digikam::MapDragDropHandler::createMimeData (
    const QList< QPersistentModelIndex > & modelIndices ) [override], [virtual]
```

Implements [Digikam::GeoDragDropHandler](#).

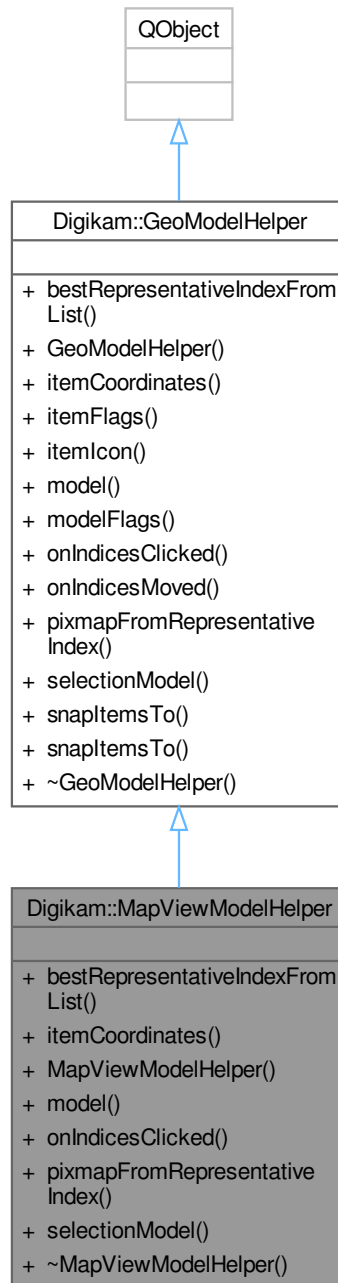
### 6.1053.1.3 dropEvent()

```
bool Digikam::MapDragDropHandler::dropEvent (
    const QDropEvent * e,
    const GeoCoordinates & dropCoordinates ) [override], [virtual]
```

Implements [Digikam::GeoDragDropHandler](#).

## 6.1054 Digikam::MapViewModelHelper Class Reference

Inheritance diagram for Digikam::MapViewModelHelper:



### Signals

- void **signalFilteredImages** (const QList< qlonglong > &idList)

## Signals inherited from [Digikam::GeoModelHelper](#)

- void **signalModelChangedDrastically** ()
- void **signalThumbnailAvailableForIndex** (const QPersistentModelIndex &index, const QPixmap &pixmap)
- void **signalVisibilityChanged** ()

## Public Member Functions

- QPersistentModelIndex **bestRepresentativeIndexFromList** (const QList< QPersistentModelIndex > &list, const int sortKey) override  
*This function finds the best representative marker from a group of markers. This is needed to display a thumbnail for a marker group.*
- bool **itemCoordinates** (const QModelIndex &index, [GeoCoordinates](#) \*const coordinates) const override  
*Gets the coordinates of a marker found at current model index.*
- **MapViewModelHelper** (QItemSelectionModel \*const selection, [DCategorizedSortFilterProxyModel](#) \*const filterModel, QObject \*const parent, const MapWidgetView::Application application)
- QAbstractItemModel \* **model** () const override
- void **onIndicesClicked** (const QList< QPersistentModelIndex > &clickedIndices) override  
*This functions is called when one clicks on a thumbnail.*
- QPixmap **pixmapFromRepresentativeIndex** (const QPersistentModelIndex &index, const QSize &size) override  
*This function retrieves the thumbnail for an index.*
- QItemSelectionModel \* **selectionModel** () const override
- **~MapViewModelHelper** () override  
*Destructor.*

## Public Member Functions inherited from [Digikam::GeoModelHelper](#)

- **GeoModelHelper** (QObject \*const parent=nullptr)
- virtual PropertyFlags **itemFlags** (const QModelIndex &index) const
- virtual bool **itemIcon** (const QModelIndex &index, QPoint \*const offset, QSize \*const size, QPixmap \*const pixmap, QUrl \*const url) const  
*these are necessary for ungrouped models*
- virtual PropertyFlags **modelFlags** () const
- virtual void **onIndicesMoved** (const QList< QPersistentModelIndex > &movedIndices, const [GeoCoordinates](#) &targetCoordinates, const QPersistentModelIndex &targetSnapIndex)
- virtual void **snapItemsTo** (const QModelIndex &targetIndex, const QList< QModelIndex > &snappedIndices)
- void **snapItemsTo** (const QModelIndex &targetIndex, const QList< QPersistentModelIndex > &snappedIndices)

## Additional Inherited Members

## Public Types inherited from [Digikam::GeoModelHelper](#)

- enum **PropertyFlag** { **FlagNull** = 0 , **FlagVisible** = 1 , **FlagMovable** = 2 , **FlagSnaps** = 4 }

## 6.1054.1 Member Function Documentation

### 6.1054.1.1 [bestRepresentativeIndexFromList\(\)](#)

```
QPersistentModelIndex Digikam::MapViewModelHelper::bestRepresentativeIndexFromList (
    const QList< QPersistentModelIndex > & list,
    const int sortKey ) [override], [virtual]
```



## Parameters

<i>list</i>	A list containing markers.
<i>sortKey</i>	Determines the sorting options and is actually of type <code>GPSItemInfoSorter::SortOptions</code>

## Returns

Returns the index of the marker.

Reimplemented from [Digikam::GeoModelHelper](#).

**6.1054.1.2 itemCoordinates()**

```
bool Digikam::MapViewModelHelper::itemCoordinates (
    const QModelIndex & index,
    GeoCoordinates *const coordinates ) const [override], [virtual]
```

## Parameters

<i>index</i>	Current model index.
<i>coordinates</i>	Here will be returned the coordinates of the current marker.

## Returns

True, if the marker has coordinates.

Implements [Digikam::GeoModelHelper](#).

**6.1054.1.3 model()**

```
QAbstractItemModel * Digikam::MapViewModelHelper::model ( ) const [override], [virtual]
```

## Returns

Returns digiKam's filter model.

Implements [Digikam::GeoModelHelper](#).

**6.1054.1.4 onIndicesClicked()**

```
void Digikam::MapViewModelHelper::onIndicesClicked (
    const QList< QPersistentModelIndex > & clickedIndices ) [override], [virtual]
```

## Parameters

<i>clickedIndices</i>	A list containing the marker indices belonging to the group whose thumbnail has been clicked.
-----------------------	---

Reimplemented from [Digikam::GeoModelHelper](#).

#### 6.1054.1.5 pixmapFromRepresentativeIndex()

```
QPixmap Digikam::MapViewModelHelper::pixmapFromRepresentativeIndex (
    const QPersistentModelIndex & index,
    const QSize & size ) [override], [virtual]
```

##### Parameters

<i>index</i>	The marker's index.
<i>size</i>	The size of the thumbnail.

##### Returns

If the thumbnail has been loaded in the [ThumbnailLoadThread](#) instance, it is returned. If not, a QPixmap is returned and [ThumbnailLoadThread](#)'s signal named `signalThumbnailLoaded` is emitted when the thumbnail becomes available.

Reimplemented from [Digikam::GeoModelHelper](#).

#### 6.1054.1.6 selectionModel()

```
QItemSelectionModel * Digikam::MapViewModelHelper::selectionModel ( ) const [override], [virtual]
```

##### Returns

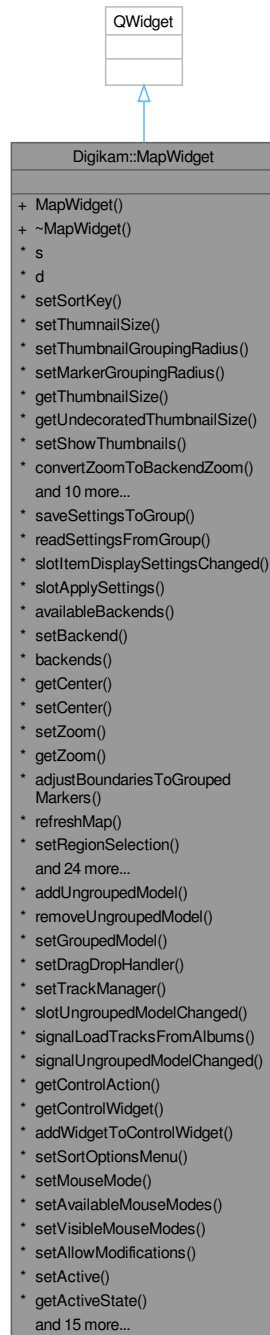
Returns digiKam's selection model.

Implements [Digikam::GeoModelHelper](#).

## 6.1055 Digikam::MapWidget Class Reference

The central map view class of geolocation interface.

Inheritance diagram for Digikam::MapWidget:



## Classes

- class [Private](#)

## Public Member Functions

- **MapWidget** (QWidget \*const parent=nullptr)
- **~MapWidget** () override

## Appearance

- void [setSortKey](#) (const int sortKey)
- void [setThumbnailSize](#) (const int newThumbnailSize)
- void **setThumbnailGroupingRadius** (const int newGroupingRadius)
- void **setMarkerGroupingRadius** (const int newGroupingRadius)
- int **getThumbnailSize** () const
- int **getUndecoratedThumbnailSize** () const
- void **setShowThumbnails** (const bool state)
- QString [convertZoomToBackendZoom](#) (const QString &someZoom, const QString &targetBackend) const
- void [getColorInfos](#) (const int clusterIndex, QColor \*fillColor, QColor \*strokeColor, Qt::PenStyle \*strokeStyle, QString \*labelText, QColor \*labelColor, const GeoGroupState \*const overrideSelection=nullptr, const int \*const overrideCount=nullptr) const  
*Return color and style information for rendering the cluster.*
- void [getColorInfos](#) (const GeoGroupState groupState, const int nMarkers, QColor \*fillColor, QColor \*strokeColor, Qt::PenStyle \*strokeStyle, QString \*labelText, QColor \*labelColor) const
- void **slotShowThumbnailsChanged** ()
- void **slotZoomIn** ()
- void **slotZoomOut** ()
- void **slotDecreaseThumbnailSize** ()
- void **slotIncreaseThumbnailSize** ()
- void **stopThumbnailTimer** ()
- void **signalRemoveCurrentFilter** ()
- void **signalStickyModeChanged** ()

## Settings Management related functions

- void **saveSettingsToGroup** (KConfigGroup \*const group)
- void **readSettingsFromGroup** (const KConfigGroup \*const group)
- void [slotItemDisplaySettingsChanged](#) ()

## Map related functions

- QStringList **availableBackends** () const
- bool [setBackend](#) (const QString &backendName)
- QList< [MapBackend](#) \* > **backends** () const
- [GeoCoordinates](#) **getCenter** () const
- void **setCenter** (const [GeoCoordinates](#) &coordinate)
- void **setZoom** (const QString &newZoom)
- QString **getZoom** ()
- void [adjustBoundariesToGroupedMarkers](#) (const bool useSaneZoomLevel=true)  
*Adjusts the visible map area such that all grouped markers are visible.*
- void **refreshMap** ()
- void **setRegionSelection** (const [GeoCoordinates::Pair](#) &region)
- [GeoCoordinates::Pair](#) **getRegionSelection** ()
- void **clearRegionSelection** ()
- void **updateMarkers** ()
- void [updateClusters](#) ()
- void **markClustersAsDirty** ()
- QPixmap [getDecoratedPixmapForCluster](#) (const int clusterId, const [GeoGroupState](#) \*const selectedStateOverride, const int \*const countOverride, QPoint \*const centerPoint)
- QVariant **getClusterRepresentativeMarker** (const int clusterIndex, const int sortKey)
- void **slotBackendReadyChanged** (const QString &backendName)

- void **slotChangeBackend** (QAction \*action)
- void **slotBackendZoomChanged** (const QString &newZoom)
- void **slotClustersMoved** (const QList &clusterIndices, const QPair< int, QModelIndex > &snapTarget)
- void **slotClustersClicked** (const QList &clusterIndices)
- void **slotLazyReclusteringRequestCallBack** ()
  - Helper function to buffer reclustering.*
- void **slotRequestLazyReclustering** ()
  - Request reclustering, repeated calls should generate only one actual update of the clusters.*
- void **slotRemoveCurrentRegionSelection** ()
- void **slotNewSelectionFromMap** (const Digikam::GeoCoordinates::Pair &sel)
- bool **currentBackendReady** () const
- void **applyCacheToBackend** ()
- void **saveBackendToCache** ()
- void **setShowPlaceholderWidget** (const bool state)
- void **setMapWidgetInFrame** (QWidget \*const widgetForFrame)
  - Set widgetForFrame as the widget in the frame, but does not show it.*
- void **removeMapWidgetFromFrame** ()
- void **slotClustersNeedUpdating** ()
- void **signalRegionSelectionChanged** ()

### Data Management

- void **addUngroupedModel** (GeoModelHelper \*const modelHelper)
- void **removeUngroupedModel** (GeoModelHelper \*const modelHelper)
- void **setGroupedModel** (AbstractMarkerTiler \*const markerModel)
- void **setDragDropHandler** (GeoDragDropHandler \*const dragDropHandler)
- void **setTrackManager** (TrackManager \*const trackManager)
- void **slotUngroupedModelChanged** ()
- void **signalLoadTracksFromAlbums** ()
- void **signalUngroupedModelChanged** (const int index)

### UI setup

- QAction \* **getControlAction** (const QString &actionName)
- QWidget \* **getControlWidget** ()
  - Returns the control widget instance.*
- void **addWidgetToControlWidget** (QWidget \*const newWidget)
- void **setSortOptionsMenu** (QMenu \*const sortMenu)
- void **setMouseMode** (const GeoMouseModes mouseMode)
- void **setAvailableMouseModes** (const GeoMouseModes mouseModes)
- void **setVisibleMouseModes** (const GeoMouseModes mouseModes)
- void **setAllowModifications** (const bool state)
- void **setActive** (const bool state)
- bool **getActiveState** ()
- bool **getStickyModeState** () const
- void **setStickyModeState** (const bool state)
- void **setVisibleExtraActions** (const GeoExtraActions actions)
- void **setEnabledExtraActions** (const GeoExtraActions actions)
- void **slotMouseModeChanged** (QAction \*triggeredAction)
- void **rebuildConfigurationMenu** ()
- void **createActions** ()
- void **createActionsForBackendSelection** ()

- void **dropEvent** (QDropEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dragEnterEvent** (QDragEnterEvent \*event) override
- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **slotUpdateActionsEnabled** ()
- void **slotStickyModeChanged** ()
- void **signalMouseModeChanged** (const Digikam::GeoMouseModes &currentMouseMode)

### 6.1055.1 Detailed Description

The [MapWidget](#) class is the central widget of geolocation interface. It provides a widget which can display maps using either the Marble or Google Maps backend. Using a model, items can be displayed on the map. For models containing only a small number of items, the items can be shown directly, but for models with a larger number of items, the items can also be grouped. Currently, any number of ungrouped models can be shown, but only one grouped model. Item selection models can also be used along with the models, to interact with the selection states of the items on the map. In order to use a model with geolocation interface, however, a model helper has to be implemented, which extracts data from the model that is not provided by the Qt part of a model's API.

Now, a brief introduction on how to get geolocation interface working is provided:

- First, an instance of [MapWidget](#) has to be created.
- Next, [GeoModelHelper](#) has to be subclassed and at least the pure virtual functions have to be implemented.
- To show the model's data ungrouped, the model helper has to be added to [MapWidget](#) instance using `addUngroupedModel`.
- To show the model's data grouped, an instance of [AbstractMarkerTiler](#) has to be created and the model helper has to be set to it using `setMarkerGeoModelHelper`. The [AbstractMarkerTiler](#) has then to be given to [MapWidget](#) using `setGroupedModel`. If the items to be displayed do not reside in a model, a subclass of [AbstractMarkerTiler](#) can be created which returns just the number of items in a particular area, and picks representative items for thumbnails.
- To handle dropping of items from the host applications UI onto the map, `DragDropHandler` has to be subclassed as well and added to the model using `setDragDropHandler`.
- Finally, `setActive()` has to be called to tell the widget that it should start displaying things.

### 6.1055.2 Constructor & Destructor Documentation

#### 6.1055.2.1 ~MapWidget()

```
Digikam::MapWidget::~MapWidget ( ) [override]
```

### 6.1055.3 Member Function Documentation

#### 6.1055.3.1 addUngroupedModel()

```
void Digikam::MapWidget::addUngroupedModel (
    GeoModelHelper *const modelHelper )
```

#### 6.1055.3.2 adjustBoundariesToGroupedMarkers()

```
void Digikam::MapWidget::adjustBoundariesToGroupedMarkers (
    const bool useSaneZoomLevel = true )
```

Note that a call to this function currently has no effect if the widget has been set inactive via `setActive()` or the backend is not yet ready.

## Parameters

<i>useSaneZoomLevel</i>	Stop zooming at a sane level, if markers are too close together.
-------------------------	--

**6.1055.3.3 applyCacheToBackend()**

```
void Digikam::MapWidget::applyCacheToBackend ( ) [protected]
```

**6.1055.3.4 convertZoomToBackendZoom()**

```
QString Digikam::MapWidget::convertZoomToBackendZoom (
    const QString & someZoom,
    const QString & targetBackend ) const
```

**6.1055.3.5 dragEnterEvent()**

```
void Digikam::MapWidget::dragEnterEvent (
    QDragEnterEvent * event ) [override], [protected]
```

**6.1055.3.6 getColorInfos() [1/2]**

```
void Digikam::MapWidget::getColorInfos (
    const GeoGroupState groupState,
    const int nMarkers,
    QColor * fillColor,
    QColor * strokeColor,
    Qt::PenStyle * strokeStyle,
    QString * labelText,
    QColor * labelColor ) const
```

**6.1055.3.7 getColorInfos() [2/2]**

```
void Digikam::MapWidget::getColorInfos (
    const int clusterIndex,
    QColor * fillColor,
    QColor * strokeColor,
    Qt::PenStyle * strokeStyle,
    QString * labelText,
    QColor * labelColor,
    const GeoGroupState *const overrideSelection = nullptr,
    const int *const overrideCount = nullptr ) const
```

## Parameters

<i>clusterIndex</i>	Index of the cluster
<i>fillColor</i>	Color used to fill the circle
<i>strokeColor</i>	Color used for the stroke around the circle
<i>strokeStyle</i>	Style used to draw the stroke around the circle
<i>labelText</i>	Text for the label
<i>labelColor</i>	Color for the label text
<i>overrideSelection</i>	Get the colors for a different selection state
<i>overrideCount</i>	Get the colors for a different amount of markers

### 6.1055.3.8 getDecoratedPixmapForCluster()

```
QPixmap Digikam::MapWidget::getDecoratedPixmapForCluster (
    const int clusterId,
    const GeoGroupState *const selectedStateOverride,
    const int *const countOverride,
    QPoint *const centerPoint )
```

### 6.1055.3.9 removeUngroupedModel()

```
void Digikam::MapWidget::removeUngroupedModel (
    GeoModelHelper *const modelHelper )
```

### 6.1055.3.10 setBackend()

```
bool Digikam::MapWidget::setBackend (
    const QString & backendName )
```

### 6.1055.3.11 setGroupedModel()

```
void Digikam::MapWidget::setGroupedModel (
    AbstractMarkerTiler *const markerModel )
```

### 6.1055.3.12 setSortKey()

```
void Digikam::MapWidget::setSortKey (
    const int sortKey )
```

### 6.1055.3.13 setThumbnailSize()

```
void Digikam::MapWidget::setThumbnailSize (
    const int newThumbnailSize )
```

### 6.1055.3.14 slotClustersClicked

```
void Digikam::MapWidget::slotClustersClicked (
    const QList & clusterIndices ) [protected], [slot]
```

### 6.1055.3.15 slotClustersMoved

```
void Digikam::MapWidget::slotClustersMoved (
    const QList & clusterIndices,
    const QPair< int, QModelIndex > & snapTarget ) [protected], [slot]
```



**6.1055.3.16 slotItemDisplaySettingsChanged**

```
void Digikam::MapWidget::slotItemDisplaySettingsChanged ( ) [protected], [slot]
```

**6.1055.3.17 slotMouseMoveModeChanged**

```
void Digikam::MapWidget::slotMouseMoveModeChanged (
    QAction * triggeredAction ) [protected], [slot]
```

**6.1055.3.18 slotNewSelectionFromMap**

```
void Digikam::MapWidget::slotNewSelectionFromMap (
    const Digikam::GeoCoordinates::Pair & sel ) [protected], [slot]
```

**6.1055.3.19 slotUpdateActionsEnabled**

```
void Digikam::MapWidget::slotUpdateActionsEnabled ( ) [slot]
```

**6.1055.3.20 updateClusters()**

```
void Digikam::MapWidget::updateClusters ( )
```

**6.1056 Digikam::MapWidget::Private Class Reference****Public Attributes**

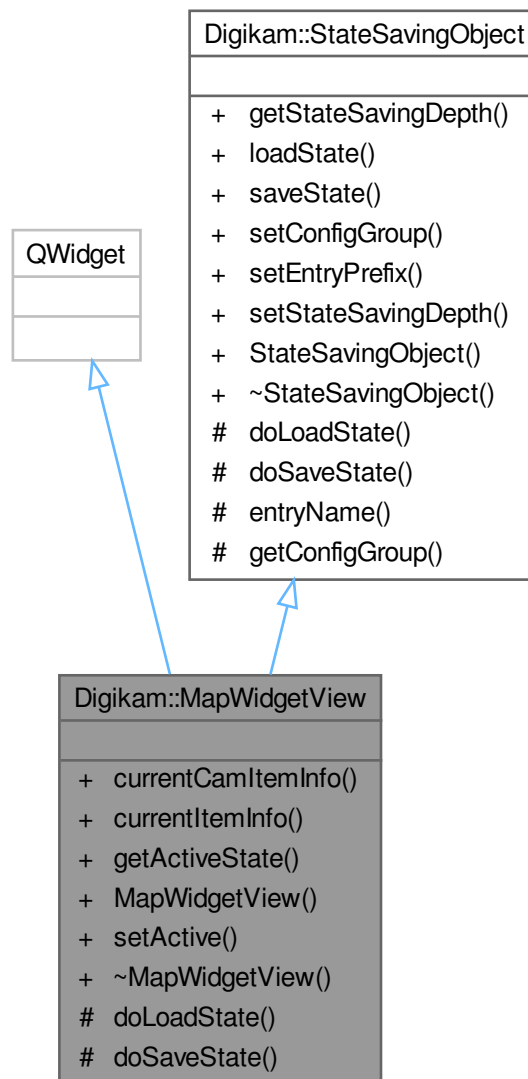
- QAction \* **actionDecreaseThumbnailSize** = nullptr
- QActionGroup \* **actionGroupBackendSelection** = nullptr
- QAction \* **actionIncreaseThumbnailSize** = nullptr
- QAction \* **actionLoadTracksFromAlbums** = nullptr
- QAction \* **actionPreviewGroupedItems** = nullptr
- QAction \* **actionPreviewSingleItems** = nullptr
- QAction \* **actionRemoveCurrentRegionSelection** = nullptr
- QAction \* **actionRemoveFilter** = nullptr
- QAction \* **actionSetFilterMode** = nullptr
- QAction \* **actionSetPanMode** = nullptr
- QAction \* **actionSetRegionSelectionFromIconMode** = nullptr
- QAction \* **actionSetRegionSelectionMode** = nullptr
- QAction \* **actionSetSelectThumbnailMode** = nullptr
- QAction \* **actionSetZoomIntoGroupMode** = nullptr
- QAction \* **actionShowNumbersOnItems** = nullptr
- QAction \* **actionShowThumbnails** = nullptr
- QAction \* **actionStickyMode** = nullptr
- QAction \* **actionZoomIn** = nullptr
- QAction \* **actionZoomOut** = nullptr
- GeoExtraActions **availableExtraActions**
- QToolButton \* **buttonStickyMode** = nullptr

- [GeoCoordinates](#) **cacheCenterCoordinate** = [GeoCoordinates](#)(52.0, 6.0)
  - NOTE: these values are cached in case the backend is not ready:*
- QString **cacheZoom** = QLatin1String("marble:900")
- QMenu \* **configurationMenu** = nullptr
  - actions for controlling the widget*
- QPointer< QWidget > **controlWidget**
- [MapBackend](#) \* **currentBackend** = nullptr
- QString **currentBackendName**
- [GeoDragDropHandler](#) \* **dragDropHandler** = nullptr
- QWidget \* **hBoxForAdditionalControlWidgetItems** = nullptr
- bool **lazyReclusteringRequested** = false
- QList< [MapBackend](#) \* > **loadedBackends**
- QToolButton \* **loadTracksFromAlbums** = nullptr
- QActionGroup \* **mouseModeActionGroup** = nullptr
- QWidget \* **mouseModesHolder** = nullptr
- [PlaceholderWidget](#) \* **placeholderWidget** = nullptr
  - NOTE: to be sorted later.*
- QToolButton \* **removeCurrentSelectionButton** = nullptr
- QToolButton \* **removeFilterModeButton** = nullptr
- QToolButton \* **setFilterModeButton** = nullptr
- QToolButton \* **setPanModeButton** = nullptr
- QToolButton \* **setRegionSelectionFromIconModeButton** = nullptr
- QToolButton \* **setSelectionModeButton** = nullptr
- QToolButton \* **setSelectThumbnailMode** = nullptr
- QToolButton \* **setZoomModeButton** = nullptr
- QMenu \* **sortMenu** = nullptr
- QStackedLayout \* **stackedLayout** = nullptr
- bool **thumbnailsHaveBeenLoaded** = false
- QTimer \* **thumbnailTimer** = nullptr
- int **thumbnailTimerCount** = 0
- GeoExtraActions **visibleExtraActions**

## 6.1057 Digikam::MapWidgetView Class Reference

Class containing digiKam's central map view.

Inheritance diagram for Digikam::MapView:



### Public Types

- enum **Application** { **ApplicationDigikam** = 1 , **ApplicationImportUI** = 2 }

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Member Functions

- [CamItemInfo](#) `currentCamItemInfo ()` const  
Returns the *CamItemInfo* for the current image.
- [ItemInfo](#) `currentItemInfo ()` const  
Returns the *ItemInfo* for the current image.
- bool `getActiveState ()` const
- [MapWidgetView](#) ([QItemSelectionModel](#) \*const selectionModel, [DCategorizedSortFilterProxyModel](#) \*const imageFilterModel, [QWidget](#) \*const parent, const [Application](#) application)  
Constructor.
- void `setActive (const bool state)`  
Set the map active/inactive.
- `~MapWidgetView ()` override  
Destructor.

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) `getStateSavingDepth ()` const
- void `loadState ()`
- void `saveState ()`
- virtual void `setConfigGroup (const KConfigGroup &group)`
- virtual void `setEntryPrefix (const QString &prefix)`
- void `setStateSavingDepth (const StateSavingDepth depth)`
- [StateSavingObject](#) ([QObject](#) \*const host)
- virtual `~StateSavingObject ()`

## Protected Member Functions

- void `doLoadState ()` override
- void `doSaveState ()` override

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- [QString](#) `entryName (const QString &base)` const
- [KConfigGroup](#) `getConfigGroup ()` const

## 6.1057.1 Constructor & Destructor Documentation

### 6.1057.1.1 MapWidgetView()

```
Digikam::MapWidgetView::MapWidgetView (
    QItemSelectionModel *const selectionModel,
    DCategorizedSortFilterProxyModel *const imageFilterModel,
    QWidget *const parent,
    const Application application ) [explicit]
```

#### Parameters

<i>selectionModel</i>	digikam's selection model
<i>imageFilterModel</i>	digikam's filter model
<i>parent</i>	the parent object
<i>application</i>	the type of application host

## 6.1057.2 Member Function Documentation

### 6.1057.2.1 currentCamItemInfo()

`CamItemInfo` Digikam::MapWidgetView::currentCamItemInfo ( ) const

### 6.1057.2.2 currentItemInfo()

`ItemInfo` Digikam::MapWidgetView::currentItemInfo ( ) const

### 6.1057.2.3 doLoadState()

`void` Digikam::MapWidgetView::doLoadState ( ) [override], [protected], [virtual]

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1057.2.4 doSaveState()

`void` Digikam::MapWidgetView::doSaveState ( ) [override], [protected], [virtual]

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1057.2.5 getActiveState()

`bool` Digikam::MapWidgetView::getActiveState ( ) const

#### Returns

The map's active state

### 6.1057.2.6 setActive()

`void` Digikam::MapWidgetView::setActive (   
 `const bool state` )

#### Parameters

<code>state</code>	If true, the map is active.
--------------------	-----------------------------

## 6.1058 Digikam::Mat Struct Reference

### Public Attributes

- int [cols](#)
- double \* [data](#)
- int [rows](#)

### 6.1058.1 Detailed Description

[Mat](#):

Normal matrix type. Indices range from [0, rows - 1 ] and [0, cols - 1].

### 6.1058.2 Member Data Documentation

#### 6.1058.2.1 cols

```
int Digikam::Mat::cols
```

Number of columns in the matrix

#### 6.1058.2.2 data

```
double* Digikam::Mat::data
```

Content of the matrix

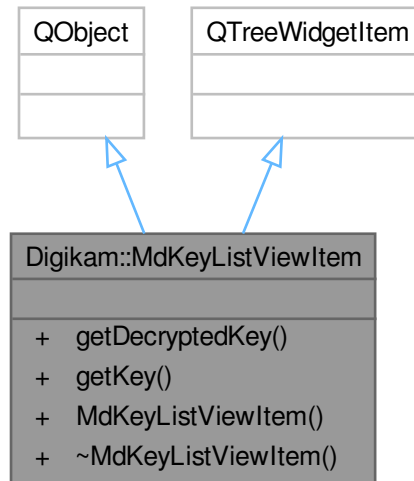
#### 6.1058.2.3 rows

```
int Digikam::Mat::rows
```

Number of rows in the matrix.

## 6.1059 Digikam::MdKeyListViewItem Class Reference

Inheritance diagram for Digikam::MdKeyListViewItem:



### Public Member Functions

- `QString` **getDecryptedKey** () const
- `QString` **getKey** () const
- **MdKeyListViewItem** (`QTreeWidgetItem *const parent`, `const QString &key`)

## 6.1060 Digikam::MediaPlayerView Class Reference

Inheritance diagram for Digikam::MediaPlayerView:



### Public Slots

- void **slotEscapePressed** ()
- void **slotRotateVideo** ()

### Signals

- void **signalEscapePreview** ()
- void **signalNextItem** ()
- void **signalPrevItem** ()

### Public Member Functions

- void **escapePreview** ()
- **MediaPlayerView** (QWidget \*const parent)
- void **reload** ()
- void **setCurrentItem** (const QUrl &url=QUrl(), bool hasPrevious=false, bool hasNext=false)
- void **setInfoInterface** ([DInfoInterface](#) \*const iface)



## Protected Member Functions

- bool `eventFilter` (QObject \*watched, QEvent \*event) override

## 6.1061 Digikam::MetadataHub Class Reference

### Public Types

- enum `Status` { `MetadataInvalid` , `MetadataAvailable` }
- enum `WriteComponents` { `WRITE_DATETIME` = 1 , `WRITE_TITLE` = 2 , `WRITE_COMMENTS` = 4 , `WRITE_PICKLABEL` = 8 , `WRITE_COLORLABEL` = 16 , `WRITE_RATING` = 32 , `WRITE_TEMPLATE` = 64 , `WRITE_TAGS` = 128 , `WRITE_POSITION` = 256 , `WRITE_ALL` = 511 }
- enum `WriteMode` { `FullWrite` , `FullWriteIfChanged` , `PartialWrite` }

### Public Member Functions

- QStringList `cleanupTags` (const QStringList &toClean)  
*cleanupTags remove duplicates and obsolete tags before setting metadata*
- void `load` (const `ItemInfo` &info)
- `MetadataHub` ()
- void `reset` ()
- bool `willWriteMetadata` (Digikam::MetadataHub::WriteComponent writeMode=WRITE\_ALL, const `MetaEngineSettingsContainer` &settings=`MetaEngineSettings::instance()` ->settings()) const  
*With the currently applied changes, the given writeMode and settings.*
- bool `write` (const `DImg` &image, WriteComponent writeMode=WRITE\_ALL, bool ignoreLazySync=false, const `MetaEngineSettingsContainer` &settings=`MetaEngineSettings::instance()` ->settings())  
*Constructs a meta engine object from the metadata stored in the given DImg object, calls the above method, and changes the stored metadata in the DImg object.*
- bool `write` (const QString &filePath, WriteComponent writeMode=WRITE\_ALL, bool ignoreLazySync=false, const `MetaEngineSettingsContainer` &settings=`MetaEngineSettings::instance()` ->settings())  
*Constructs a meta engine object for given filePath, calls the above method, writes the changes out to the file, and notifies the `ItemAttributesWatch`.*
- bool `writeTags` (const `DMetadata` &metadata, bool saveTags)  
*Used to deduplicate code from writeTags and usual write, all write to tags operations must be done here.*
- bool `writeTags` (const QString &filePath, WriteComponent writeMode=WRITE\_ALL, const `MetaEngineSettingsContainer` &settings=`MetaEngineSettings::instance()` ->settings())  
*Will write only Tags to image. Used by `TagsManager` to write tags to image Other metadata are not updated.*
- void `writeToBaloo` (const QString &filePath, const `MetaEngineSettingsContainer` &settings=`MetaEngineSettings::instance()` ->settings())  
*write tags, comments and rating to KDE Nepomuk replacement: Baloo*
- bool `writeToMetadata` (const `ItemInfo` &info, WriteComponent writeMode=WRITE\_ALL, bool ignoreLazySync=false, const `MetaEngineSettingsContainer` &settings=`MetaEngineSettings::instance()` ->settings())  
*writeToMetadata write to metadata using image info to retrieve tags and filepath use this method when multiple image infos are loaded in hub*

## Protected Member Functions

- void **applyChangeNotifications** ()
- void **load** (const QDateTime &dateTime, const [CaptionsMap](#) &titles, const [CaptionsMap](#) &comment, int colorLabel, int pickLabel, int rating, const [Template](#) &t)
- void **loadFaceTags** (const [ItemInfo](#) &info)
- void **loadTags** (const QList< int > &loadedTagIds)
- void **notifyTagDeleted** (int id)
- bool **write** ([DMetadata](#) &metadata, WriteComponent writeMode=WRITE\_ALL, const [MetaEngineSettingsContainer](#) &settings=[MetaEngineSettings::instance\(\)](#) ->settings())

*Applies the set of metadata contained in this [MetadataHub](#) to the given meta engine object.*

## 6.1061.1 Member Enumeration Documentation

### 6.1061.1.1 Status

```
enum Digikam::MetadataHub::Status
```

The status enum describes the result of joining several metadata sets. If only one set has been added, the status is always MetadataAvailable. If no set has been added, the status is always MetadataInvalid

#### Enumerator

MetadataInvalid	not yet filled with any value
MetadataAvailable	only one data set has been added, or a common value is available

### 6.1061.1.2 WriteMode

```
enum Digikam::MetadataHub::WriteMode
```

#### Enumerator

FullWrite	Write all available information
FullWriteIfChanged	Do a full write if and only if <ul style="list-style-type: none"> <li>• metadata fields changed</li> <li>• the changed fields shall be written according to write settings "Changed" in this context means changed by one of the set... methods, the <a href="#">load()</a> methods are ignored for this attribute. This mode allows to avoid write operations when e.g. the user does not want keywords to be written and only changes keywords.</li> </ul>
PartialWrite	Write only the changed parts. Metadata fields which cannot be changed from <a href="#">MetadataHub</a> (photographer ID etc.) will never be written

## 6.1061.2 Constructor & Destructor Documentation

### 6.1061.2.1 MetadataHub()

```
Digikam::MetadataHub::MetadataHub ( )
```

Constructs a [MetadataHub](#).

## 6.1061.3 Member Function Documentation

### 6.1061.3.1 cleanupTags()

```
QStringList Digikam::MetadataHub::cleanupTags (
    const QStringList & toClean )
```

#### Parameters

<i>toClean</i>	tag list to be cleared and de-duplicated
----------------	--

#### Returns

clean tag list

### 6.1061.3.2 load() [1/2]

```
void Digikam::MetadataHub::load (
    const ItemInfo & info )
```

Add metadata information contained in the [ItemInfo](#) object. This method (or in combination with the other load methods) can be called multiple times on the same [MetadataHub](#) object. In this case, the metadata will be combined.

### 6.1061.3.3 load() [2/2]

```
void Digikam::MetadataHub::load (
    const QDateTime & dateTime,
    const CaptionsMap & titles,
    const CaptionsMap & comments,
    int colorLabel,
    int pickLabel,
    int rating,
    const Template & t ) [protected]
```

private common code to load dateTime, comment, color label, pick label, rating

### 6.1061.3.4 loadTags()

```
void Digikam::MetadataHub::loadTags (
    const QList< int > & loadedTags ) [protected]
```

private common code to merge tags

### 6.1061.3.5 willWriteMetadata()

```
bool Digikam::MetadataHub::willWriteMetadata (
    Digikam::MetadataHub::WriteComponent writeMode = WRITE_ALL,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
) const
```

## Parameters

<i>writeMode</i>	The mode to write metadata
<i>settings</i>	The metadata settings to be set

## Returns

if write(DMetadata), write(QString) or write(DImg) will actually apply any changes.

**6.1061.3.6 write()** [1/3]

```
bool Digikam::MetadataHub::write (
    const DImg & image,
    WriteComponent writeMode = WRITE_ALL,
    bool ignoreLazySync = false,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
)
```

## Parameters

<i>image</i>	The <a href="#">DImg</a> container to retrieve current tags
<i>writeMode</i>	The mode to write metadata
<i>ignoreLazySync</i>	The flag to ignore the lazy sync metadata stage
<i>settings</i>	The metadata settings to be set

## Returns

Returns if the [DImg](#) object has been touched

**6.1061.3.7 write()** [2/3]

```
bool Digikam::MetadataHub::write (
    const QString & filePath,
    WriteComponent writeMode = WRITE_ALL,
    bool ignoreLazySync = false,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
)
```

## Parameters

<i>filePath</i>	The file path to retrieve current tags
<i>writeMode</i>	The mode to write metadata
<i>ignoreLazySync</i>	The flag to ignore the lazy sync metadata stage
<i>settings</i>	The metadata settings to be set

**Warning**

Do not use this method when multiple image infos are loaded It will result in disjoint tags not being written Use `writeToMetadata(Image info ...)` instead

**Returns**

Returns if the file has been touched

**6.1061.3.8 write() [3/3]**

```
bool Digikam::MetadataHub::write (
    DMetadata & metadata,
    WriteComponent writeMode = WRITE_ALL,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
) [protected]
```

**Parameters**

<i>metadata</i>	The metadata backend instance.
<i>writeMode</i>	The mode to write metadata.
<i>settings</i>	The <a href="#">MetaEngineSettingsContainer</a> determine whether data is actually set or not. The following metadata fields may be set (depending on settings): <ul style="list-style-type: none"> <li>• Comment</li> <li>• Date</li> <li>• Rating</li> <li>• Tags</li> <li>• Photographer ID (data from settings)</li> <li>• Credits (data from settings)</li> </ul>

**Note**

The data fields taken from this [MetadataHub](#) object are only set if their status is `MetadataAvailable`. If the status is `MetadataInvalid` or `MetadataDisjoint`, the respective metadata field is not touched.

**Returns**

Returns true if the metadata object has been touched.

**6.1061.3.9 writeTags() [1/2]**

```
bool Digikam::MetadataHub::writeTags (
    const DMetadata & metadata,
    bool saveTags )
```

**Parameters**

<i>metadata</i>	meta engine object that apply changes
<i>saveTags</i>	save switch

**Returns**

if tags were successfully set

**6.1061.3.10 writeTags() [2/2]**

```
bool Digikam::MetadataHub::writeTags (
    const QString & filePath,
    WriteComponent writeMode = WRITE_ALL,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
)
```

**Parameters**

<i>filePath</i>	The file path to update current tags
<i>writeMode</i>	The mode to write metadata
<i>settings</i>	The metadata settings to be set

**Returns**

if tags were successfully written.

**6.1061.3.11 writeToBaloo()**

```
void Digikam::MetadataHub::writeToBaloo (
    const QString & filePath,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
)
```

**Parameters**

<i>filePath</i>	path to file to add comments, tags and rating
<i>settings</i>	metadata settings to be set

**6.1061.3.12 writeToMetadata()**

```
bool Digikam::MetadataHub::writeToMetadata (
    const ItemInfo & info,
    WriteComponent writeMode = WRITE_ALL,
    bool ignoreLazySync = false,
    const MetaEngineSettingsContainer & settings = MetaEngineSettings::instance()->settings()
)
```

## Parameters

<i>info</i>	The image info to retrieve current tags
<i>writeMode</i>	The mode to write metadata
<i>ignoreLazySync</i>	The flag to ignore the lazy sync metadata stage
<i>settings</i>	The metadata settings to be set

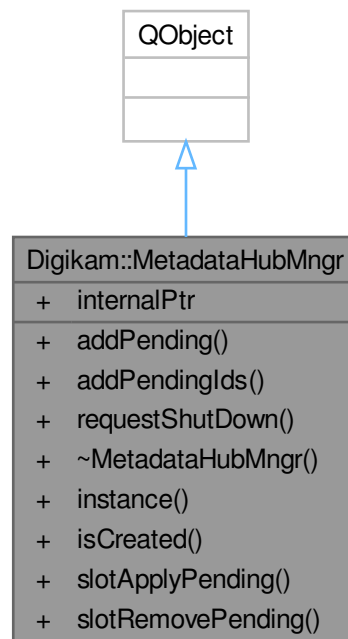
## Returns

true if everything is successful

safe method

## 6.1062 Digikam::MetadataHubMngr Class Reference

Inheritance diagram for Digikam::MetadataHubMngr:



## Public Slots

- void **slotApplyPending** ()
- void **slotRemovePending** (const [ItemInfo](#) &info)

## Signals

- void **signalPendingMetadata** (int numbers)

## Public Member Functions

- void **addPending** (const [ItemInfo](#) &info)
- void **addPendingIds** (const QList< qlonglong > &imageIds)
- void **requestShutDown** ()

## Static Public Member Functions

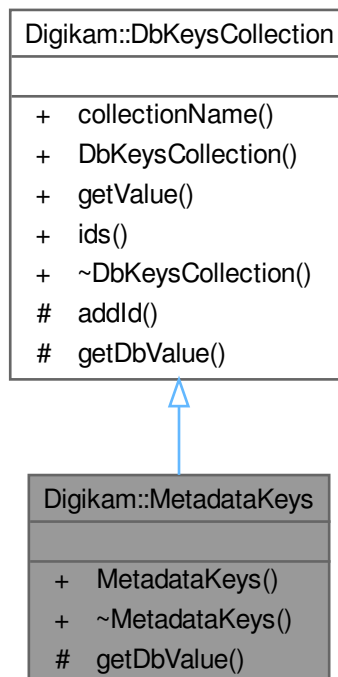
- static [MetadataHubMngr](#) \* **instance** ()
- static bool **isCreated** ()

## Static Public Attributes

- static QPointer< [MetadataHubMngr](#) > **internalPtr** = QPointer<[MetadataHubMngr](#)>()

## 6.1063 Digikam::MetadataKeys Class Reference

Inheritance diagram for Digikam::MetadataKeys:





### Protected Member Functions

- QString [getDbValue](#) (const QString &key, [ParseSettings](#) &settings) override

### Protected Member Functions inherited from [Digikam::DbKeysCollection](#)

- void [addId](#) (const QString &id, const QString &description)

### Additional Inherited Members

### Public Member Functions inherited from [Digikam::DbKeysCollection](#)

- QString [collectionName](#) () const
- [DbKeysCollection](#) (const QString &n)
- QString [getValue](#) (const QString &key, [ParseSettings](#) &settings)
- DbKeyIdsMap [ids](#) () const

## 6.1063.1 Member Function Documentation

### 6.1063.1.1 [getDbValue\(\)](#)

```
QString Digikam::MetadataKeys::getDbValue (
    const QString & key,
    ParseSettings & settings ) [override], [protected], [virtual]
```

Abstract method for retrieving the value from the database for the given key.

This method has to be implemented by all child classes. It is called by the [getValue\(\)](#) method.

#### Parameters

<i>key</i>	the key representing the value in the database
<i>settings</i>	the <a href="#">ParseSettings</a> object holding all relevant information about the image.

#### Returns

the value of the given database key

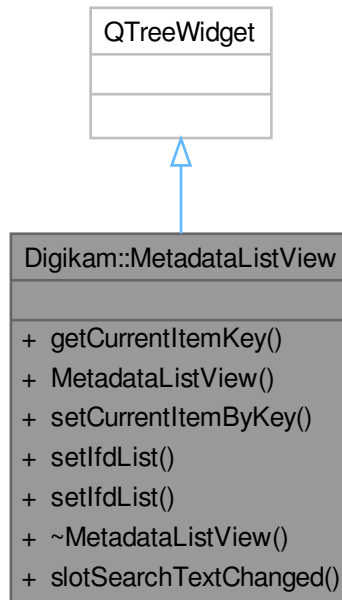
#### See also

[DbKeysCollection::getValue\(\)](#)

Implements [Digikam::DbKeysCollection](#).

## 6.1064 Digikam::MetadataListView Class Reference

Inheritance diagram for Digikam::MetadataListView:



### Public Slots

- void **slotSearchTextChanged** (const [SearchTextSettings](#) &)

### Signals

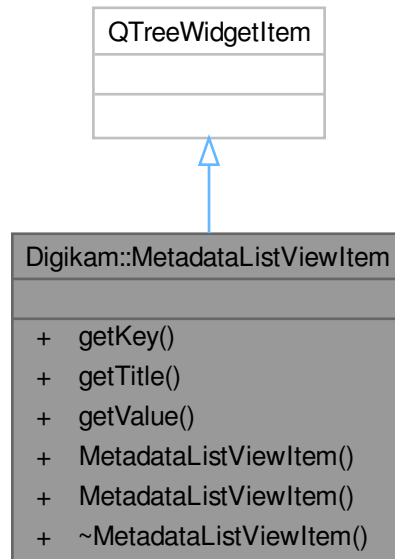
- void **signalTextFilterMatch** (bool)

### Public Member Functions

- QString **getCurrentItemKey** () const
- **MetadataListView** (QWidget \*const parent)
- void **setCurrentItemByKey** (const QString &itemKey)
- void **setIfdList** (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &keysFilter, const QStringList &tagsFilter)
- void **setIfdList** (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &tagsFilter)

## 6.1065 Digikam::MetadataListViewItem Class Reference

Inheritance diagram for Digikam::MetadataListViewItem:



### Public Member Functions

- `QString` **getKey** () const
- `QString` **getTitle** () const
- `QString` **getValue** () const
- **MetadataListViewItem** (`QTreeWidgetItem *const parent`, `const QString &key`, `const QString &title`)
- **MetadataListViewItem** (`QTreeWidgetItem *const parent`, `const QString &key`, `const QString &title`, `const QString &value`)

## 6.1066 Digikam::MetadataOption Class Reference

Inheritance diagram for Digikam::MetadataOption:



### Protected Member Functions

- QString `parseOperation` (`ParseSettings` &settings, const `QRegularExpressionMatch` &match) override

## Protected Member Functions inherited from Digikam::Rule

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from Digikam::Rule

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from Digikam::Rule

- void **signalTokenTriggered** (const QString &)

## Public Member Functions inherited from Digikam::Option

- **Option** (const QString &name, const QString &description)
- **Option** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from Digikam::Rule

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- **ParseResults** **parse** (**ParseSettings** &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Static Public Member Functions inherited from Digikam::Rule

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from Digikam::Rule

- virtual void **slotTokenTriggered** (const QString &)

## 6.1066.1 Member Function Documentation

### 6.1066.1.1 parseOperation()

```
QString Digikam::MetadataOption::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [virtual]
```

TODO: describe me

## Parameters

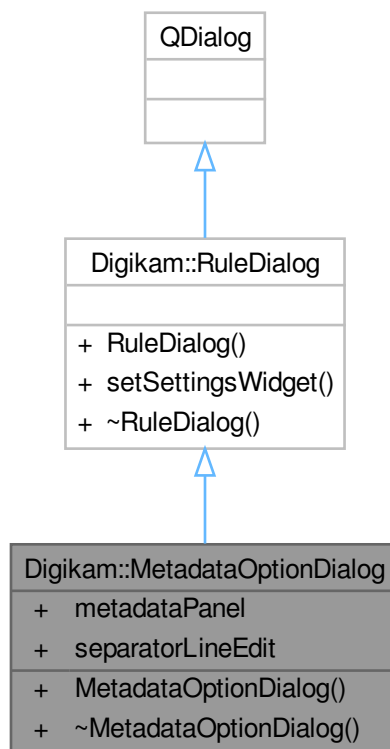
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Option](#).

## 6.1067 Digikam::MetadataOptionDialog Class Reference

Inheritance diagram for Digikam::MetadataOptionDialog:



### Public Member Functions

- `MetadataOptionDialog` (`Rule *const parent`)

## Public Member Functions inherited from Digikam::RuleDialog

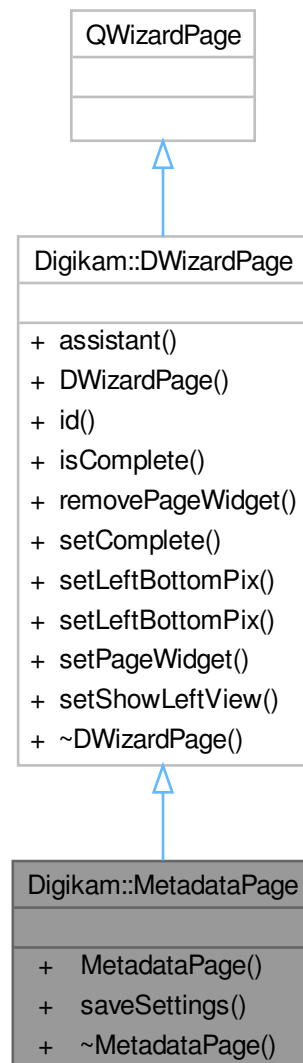
- **RuleDialog** ([Rule](#) \*const parent)
- void **setSettingsWidget** (QWidget \*const settingsWidget)

## Public Attributes

- [MetadataPanel](#) \* **metadataPanel** = nullptr
- QLineEdit \* **separatorLineEdit** = nullptr

## 6.1068 Digikam::MetadataPage Class Reference

Inheritance diagram for Digikam::MetadataPage:



### Public Member Functions

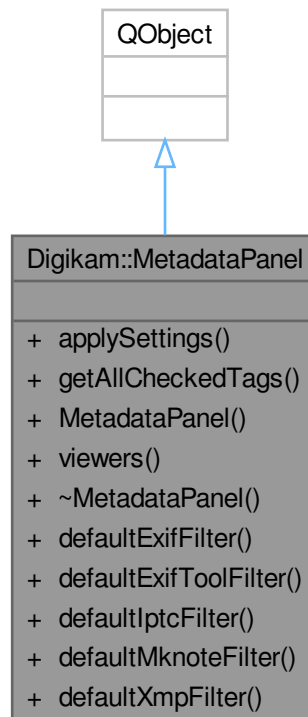
- **MetadataPage** (QWizard \*const dlg)
- void **saveSettings** ()

### Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)
- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.1069 Digikam::MetadataPanel Class Reference

Inheritance diagram for Digikam::MetadataPanel:





### Public Member Functions

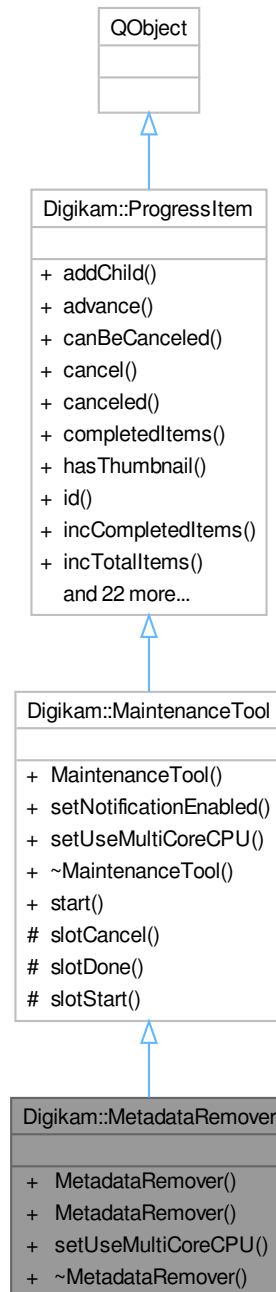
- void **applySettings** ()
- QStringList **getAllCheckedTags** () const
- **MetadataPanel** (QTabWidget \*const tab)
- QList< [MetadataSelectorView](#) \* > **viewers** () const

### Static Public Member Functions

- static QStringList **defaultExifFilter** ()
- static QStringList **defaultExifToolFilter** ()
- static QStringList **defaultIptcFilter** ()
- static QStringList **defaultMknoteFilter** ()
- static QStringList **defaultXmpFilter** ()

## 6.1070 Digikam::MetadataRemover Class Reference

Inheritance diagram for Digikam::MetadataRemover:



### Public Types

- enum **RemoveAction** { **None** = 0 , **Faces** , **Tags** }

## Public Member Functions

- [MetadataRemover](#) (const AlbumList &list=AlbumList(), RemoveAction action=None, [ProgressItem](#) \*const parent=nullptr)
- [MetadataRemover](#) (const ItemInfoList &list, RemoveAction action=None, [ProgressItem](#) \*const parent=nullptr)
- void [setUseMultiCoreCPU](#) (bool b) override

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- [MaintenanceTool](#) (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void [addChild](#) ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void [cancel](#) ()
- bool [canceled](#) () const
- unsigned int [completedItems](#) () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool [incCompletedItems](#) (unsigned int v=1)
- void [incTotalItems](#) (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool [canBeCanceled](#), bool hasThumb)
- void [removeChild](#) ([ProgressItem](#) \*const kiddo)
- void [reset](#) ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void [setComplete](#) ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool [setCompletedItems](#) (unsigned int v)
- void [setLabel](#) (const QString &v)
- void [setProgress](#) (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void [setShowAtStart](#) (bool [showAtStart](#))
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void [setStatus](#) (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void [setThumbnail](#) (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void [setTotalItems](#) (unsigned int v)
- void [setUsesBusyIndicator](#) (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*

- bool [showAtStart](#) () const
- const QString & [status](#) () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
  - Recalculate progress according to total/completed items and update.*
- bool [usesBusyIndicator](#) () const

#### Additional Inherited Members

#### Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

#### Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

#### Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)
  - Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)
  - Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)
  - Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)
  - Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)
  - Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)
  - Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)
  - Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)
  - Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

#### Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.1070.1 Constructor & Destructor Documentation

### 6.1070.1.1 MetadataRemover() [1/2]

```
Digikam::MetadataRemover::MetadataRemover (
    const AlbumList & list = AlbumList(),
    RemoveAction action = None,
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor which remove all images metadata from an Albums list. If list is empty, whole Albums collection is processed.

### 6.1070.1.2 MetadataRemover() [2/2]

```
Digikam::MetadataRemover::MetadataRemover (
    const ItemInfoList & list,
    RemoveAction action = None,
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor which remove all images metadata from an Images list

## 6.1070.2 Member Function Documentation

### 6.1070.2.1 setUseMultiCoreCPU()

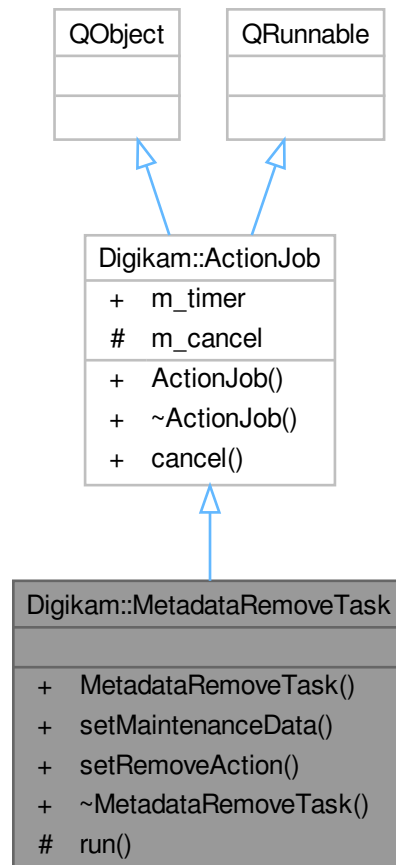
```
void Digikam::MetadataRemover::setUseMultiCoreCPU (
    bool ) [override], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented from [Digikam::MaintenanceTool](#).

## 6.1071 Digikam::MetadataRemoveTask Class Reference

Inheritance diagram for Digikam::MetadataRemoveTask:



### Signals

- void **signalFinished** (const [ItemInfo](#) &, const QImage &)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Member Functions

- void **setMaintenanceData** ([MaintenanceData](#) \*const data=nullptr)
- void **setRemoveAction** ([MetadataRemover::RemoveAction](#) action)

**Public Member Functions inherited from [Digikam::ActionJob](#)**

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

**Protected Member Functions**

- void [run](#) () override

**Additional Inherited Members****Public Slots inherited from [Digikam::ActionJob](#)**

- void [cancel](#) ()

**Public Attributes inherited from [Digikam::ActionJob](#)**

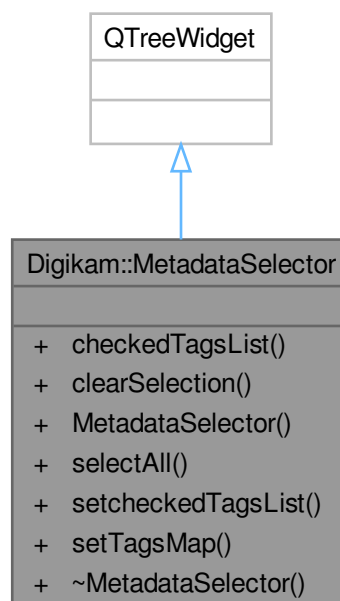
- QElapsedTimer [m\\_timer](#)

**Protected Attributes inherited from [Digikam::ActionJob](#)**

- bool [m\\_cancel](#) = false

**6.1072 Digikam::MetadataSelector Class Reference**

Inheritance diagram for Digikam::MetadataSelector:

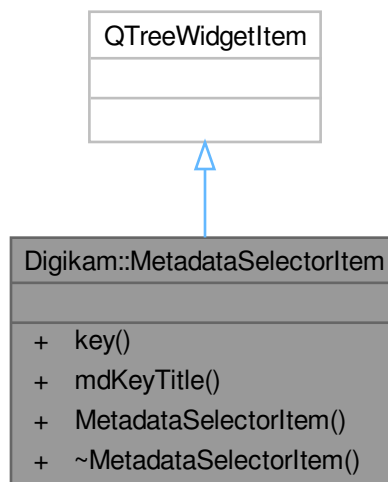


### Public Member Functions

- QStringList **checkedTagsList** ()
- void **clearSelection** ()
- **MetadataSelector** ([MetadataSelectorView](#) \*const parent)
- void **selectAll** () override
- void **setCheckedTagsList** (const QStringList &list)
- void **setTagsMap** (const [DMetadata::TagsMap](#) &map)

## 6.1073 Digikam::MetadataSelectorItem Class Reference

Inheritance diagram for Digikam::MetadataSelectorItem:



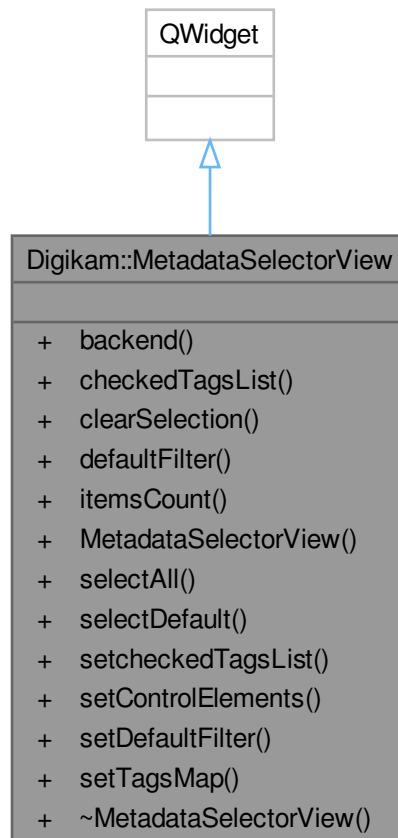
### Public Member Functions

- QString **key** () const
- QString **mdKeyTitle** () const
- **MetadataSelectorItem** ([MdKeyListViewItem](#) \*const parent, const QString &key, const QString &title, const QString &desc)



## 6.1074 Digikam::MetadataSelectorView Class Reference

Inheritance diagram for Digikam::MetadataSelectorView:



### Public Types

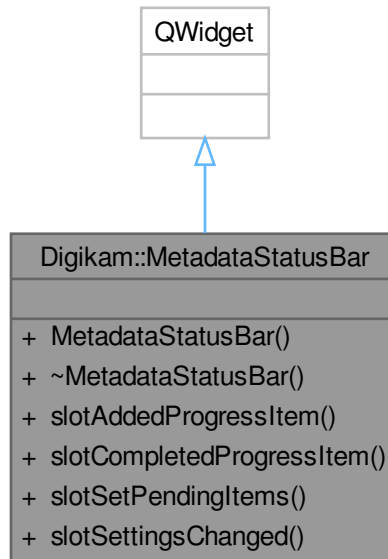
- enum **Backend** { **Exiv2Backend** = 0 , **ExifToolBackend** }
- enum **ControlElement** { **SelectAllBtn** = 0x01 , **ClearBtn** = 0x02 , **DefaultBtn** = 0x04 , **SearchBar** = 0x08 }

### Public Member Functions

- Backend **backend** () const
- QStringList **checkedTagsList** () const
- void **clearSelection** ()
- QStringList **defaultFilter** () const
- int **itemCount** () const
- **MetadataSelectorView** (QWidget \*const parent, Backend be)
- void **selectAll** ()
- void **selectDefault** ()
- void **setcheckedTagsList** (const QStringList &list)
- void **setControlElements** (ControlElements controllerMask)
- void **setDefaultFilter** (const QStringList &list)
- void **setTagsMap** (const [DMetadata::TagsMap](#) &map)

## 6.1075 Digikam::MetadataStatusBar Class Reference

Inheritance diagram for Digikam::MetadataStatusBar:



### Public Slots

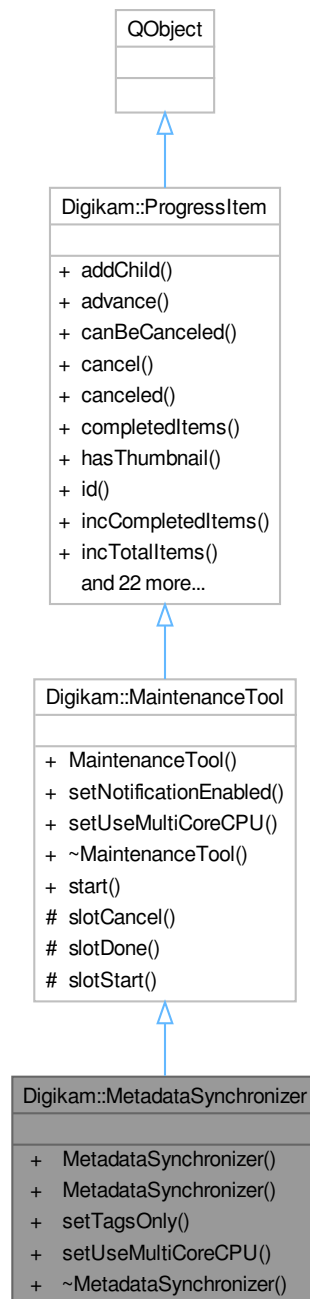
- void **slotAddedProgressItem** ([ProgressItem](#) \*item)
- void **slotCompletedProgressItem** ([ProgressItem](#) \*item)
- void **slotSetPendingItems** (int number)
- void **slotSettingsChanged** ()

### Public Member Functions

- **MetadataStatusBar** (`QWidget` \*const parent)

## 6.1076 Digikam::MetadataSynchronizer Class Reference

Inheritance diagram for Digikam::MetadataSynchronizer:



### Public Types

- enum **SyncDirection** { **WriteFromDatabaseToFile** = 0 , **ReadFromFileToDatabase** }

## Signals

- void **signalRemovePending** (const [ItemInfo](#) &info)

## Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

## Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- [MetadataSynchronizer](#) (const AlbumList &list=AlbumList(), SyncDirection direction=WriteFromDatabaseTo↔File, [ProgressItem](#) \*const parent=nullptr)
- [MetadataSynchronizer](#) (const [ItemInfoList](#) &list, SyncDirection=WriteFromDatabaseToFile, [ProgressItem](#) \*const parent=nullptr)
- void **setTagsOnly** (bool value)
- void [setUseMultiCoreCPU](#) (bool b) override

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- **MaintenanceTool** (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)

## Public Member Functions inherited from Digikam::ProgressItem

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool **advance** (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool **canBeCanceled** () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool **hasThumbnail** () const
- const QString & **id** () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & **label** () const
- [ProgressItem](#) \* **parent** () const
- unsigned int **progress** () const
- [ProgressItem](#) ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool canBeCanceled, bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void **setLabel** (const QString &v)
- void **setProgress** (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void **setShowAtStart** (bool showAtStart)
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void **setStatus** (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void **setThumbnail** (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void **setUsesBusyIndicator** (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool **showAtStart** () const
- const QString & **status** () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool **usesBusyIndicator** () const

### Additional Inherited Members

## Public Slots inherited from Digikam::MaintenanceTool

- void **start** ()

## Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

### 6.1076.1 Constructor & Destructor Documentation

#### 6.1076.1.1 [MetadataSynchronizer\(\)](#) [1/2]

```
Digikam::MetadataSynchronizer::MetadataSynchronizer (  
    const AlbumList & list = AlbumList(),  
    SyncDirection direction = WriteFromDatabaseToFile,  
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor which sync all images metadata from an Albums list. If list is empty, whole Albums collection is processed.

#### 6.1076.1.2 [MetadataSynchronizer\(\)](#) [2/2]

```
Digikam::MetadataSynchronizer::MetadataSynchronizer (  
    const ItemInfoList & list,  
    SyncDirection direction = WriteFromDatabaseToFile,  
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor which sync all images metadata from an Images list

### 6.1076.2 Member Function Documentation

#### 6.1076.2.1 [setUseMultiCoreCPU\(\)](#)

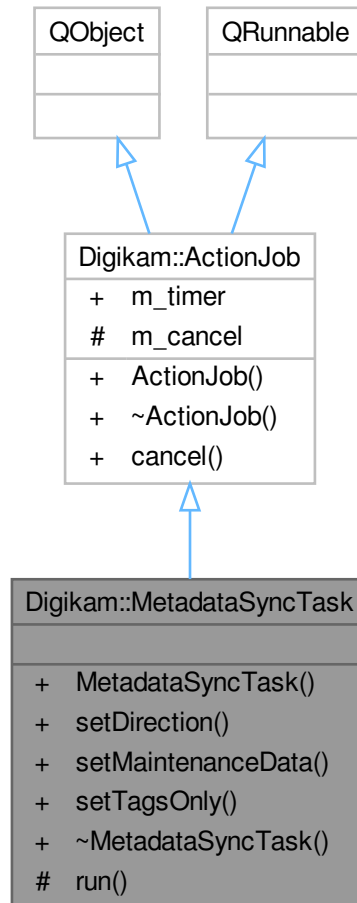
```
void Digikam::MetadataSynchronizer::setUseMultiCoreCPU (  
    bool ) [override], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented from [Digikam::MaintenanceTool](#).

## 6.1077 Digikam::MetadataSyncTask Class Reference

Inheritance diagram for Digikam::MetadataSyncTask:



### Signals

- void **signalFinished** (const [ItemInfo](#) &, const QImage &)
- void **signalRemovePending** (const [ItemInfo](#) &info)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Member Functions

- void **setDirection** (MetadataSynchronizer::SyncDirection dir)
- void **setMaintenanceData** ([MaintenanceData](#) \*const data=nullptr)
- void **setTagsOnly** (bool value)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Protected Member Functions

- void **run** () override

### Additional Inherited Members

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

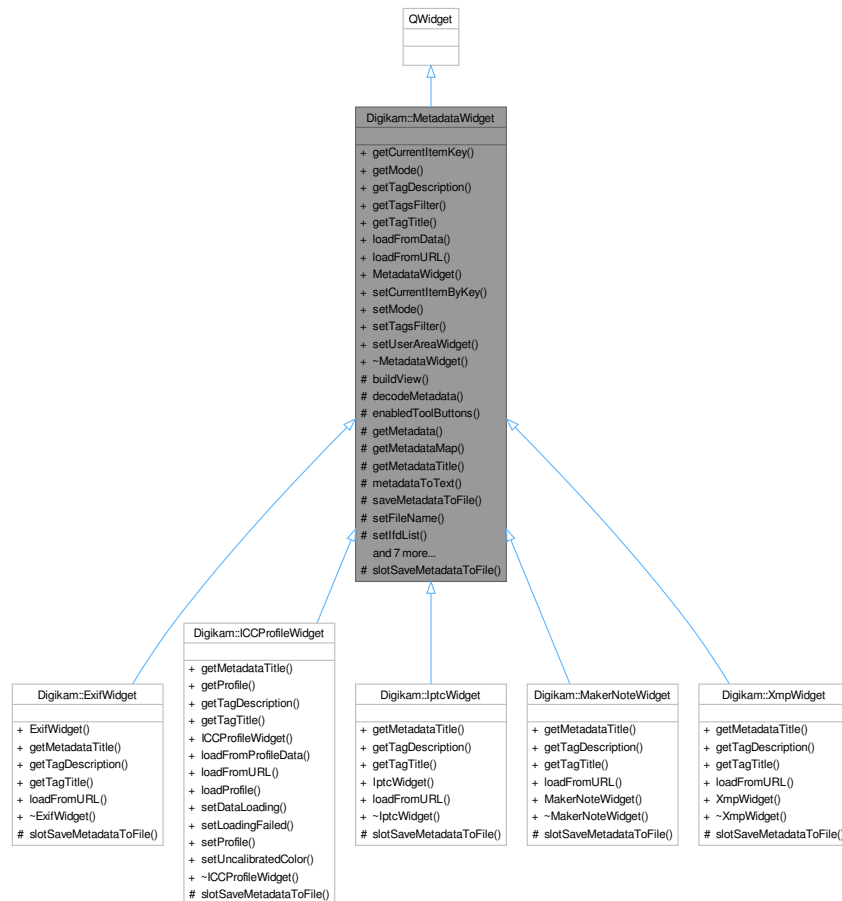
### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false



## 6.1078 Digikam::MetadataWidget Class Reference

Inheritance diagram for Digikam::MetadataWidget:



### Public Types

- enum **TagFilters** { NONE = 0 , PHOTO , CUSTOM }

### Signals

- void **signalSetupMetadataFilters** ()

### Public Member Functions

- QString **getControlItemKey** () const
- int **getMode** () const
- virtual QString **getTagDescription** (const QString &key)
- QStringList **getTagsFilter** () const
- virtual QString **getTagTitle** (const QString &key)
- virtual bool **loadFromData** (const QString &fileName, const [DMetadata](#) &data=[DMetadata](#)())

- virtual bool **loadFromURL** (const QUrl &url)=0
- **MetadataWidget** (QWidget \*const parent, const QString &name=QString())
- void **setCurrentItemByKey** (const QString &itemKey)
- void **setMode** (int mode)
- void **setTagsFilter** (const QStringList &list)
- void **setUserAreaWidget** (QWidget \*const w)

### Protected Slots

- virtual void **slotSaveMetadataToFile** ()=0

### Protected Member Functions

- virtual void **buildView** ()
- virtual bool **decodeMetadata** ()=0
- void **enabledToolButtons** (bool)
- [DMetadata](#) \* **getMetadata** () const
- const [DMetadata::MetaDatum](#) & **getMetadataMap** ()
- virtual QString **getMetadataTitle** () const =0
- QString **metadataToText** () const
- QUrl **saveMetadataToFile** (const QString &caption, const QString &fileFilter)
- void **setFileName** (const QString &fileName)
- void **setIfdList** (const [DMetadata::MetaDatum](#) &ifds, const QStringList &keysFilter, const QStringList &tagsFilter)
- void **setIfdList** (const [DMetadata::MetaDatum](#) &ifds, const QStringList &tagsFilter=QStringList())
- bool **setMetadata** (const [DMetadata](#) &data=[DMetadata](#)())
- virtual void **setMetadataEmpty** ()
- void **setMetadataMap** (const [DMetadata::MetaDatum](#) &data=[DMetadata::MetaDatum](#)())
- void **setup** ()
- bool **storeMetadataToFile** (const QUrl &url, const QByteArray &metaData)
- [MetadataListView](#) \* **view** () const

## 6.1078.1 Member Function Documentation

### 6.1078.1.1 setup()

```
void Digikam::MetadataWidget::setup ( ) [protected]
```

Call this method in children class constructors to init signal/slots connections.



## Public Types

- typedef QMap< QString, QString > [AltLangMap](#)
- enum [Backend](#) {  
[Exiv2Backend](#) = 0 , [LibRawBackend](#) , [LibHeifBackend](#) , [ImageMagickBackend](#) ,  
[FFmpegBackend](#) , [ExifToolBackend](#) , [VideoMergeBackend](#) , [NoBackend](#) }
- enum [ImageColorWorkSpace](#) { [WORKSPACE\\_UNSPECIFIED](#) = 0 , [WORKSPACE\\_SRGB](#) = 1 ,  
[WORKSPACE\\_ADOBERGB](#) = 2 , [WORKSPACE\\_UNCALIBRATED](#) = 65535 }
- enum [ImageOrientation](#) {  
[ORIENTATION\\_UNSPECIFIED](#) = 0 , [ORIENTATION\\_NORMAL](#) = 1 , [ORIENTATION\\_HFLIP](#) = 2 ,  
[ORIENTATION\\_ROT\\_180](#) = 3 ,  
[ORIENTATION\\_VFLIP](#) = 4 , [ORIENTATION\\_ROT\\_90\\_HFLIP](#) = 5 , [ORIENTATION\\_ROT\\_90](#) = 6 ,  
[ORIENTATION\\_ROT\\_90\\_VFLIP](#) = 7 ,  
[ORIENTATION\\_ROT\\_270](#) = 8 }
- typedef QMap< QString, QString > [MetaDataMap](#)
- enum [MetadataWritingMode](#) { [WRITE\\_TO\\_FILE\\_ONLY](#) = 0 , [WRITE\\_TO\\_SIDECAR\\_ONLY](#) = 1 ,  
[WRITE\\_TO\\_SIDECAR\\_AND\\_FILE](#) = 2 , [WRITE\\_TO\\_SIDECAR\\_ONLY\\_FOR\\_READ\\_ONLY\\_FILES](#) = 3  
}
- typedef QMap< QString, QStringList > [TagsMap](#)
- enum [XmpTagType](#) {  
[NormalTag](#) = 0 , [ArrayBagTag](#) = 1 , [StructureTag](#) = 2 , [ArrayLangTag](#) = 3 ,  
[ArraySeqTag](#) = 4 }

## Public Member Functions

- [MetaEngine](#) ()
- [MetaEngine](#) (const [MetaEngineData](#) &data)
- [MetaEngine](#) (const QString &filePath)
- virtual [~MetaEngine](#) ()

## General methods

- [MetaEngineData](#) **data** () const
- void **setData** (const [MetaEngineData](#) &data)
- bool **loadFromData** (const QByteArray &imgData)
- bool **loadFromDataAndMerge** (const QByteArray &imgData, const QStringList &exclude=QStringList())
- bool **isEmpty** () const
- QSize **getPixelSize** () const
- QString **getMimeType** () const
- void **setReadWithExifTool** (const bool on)
- bool **readWithExifTool** () const
- void **setWriteWithExifTool** (const bool on)
- bool **writeWithExifTool** () const
- void **setWriteRawFiles** (const bool on)
- bool **writeRawFiles** () const
- void **setWriteDngFiles** (const bool on)
- bool **writeDngFiles** () const
- void **setUseXMPSidecar4Reading** (const bool on)
- bool **useXMPSidecar4Reading** () const
- void **setUseCompatibleFileName** (const bool on)
- bool **useCompatibleFileName** () const
- void **setMetadataWritingMode** (const int mode)
- int **metadataWritingMode** () const
- void **setUpdateFileTimeStamp** (bool on)
- bool **updateFileTimeStamp** () const

## Metadata item information manipulation methods

- bool [setItemProgramId](#) (const QString &program, const QString &version) const
- QSize [getItemDimensions](#) () const
- bool [setItemDimensions](#) (const QSize &size) const
- [MetaEngine::ImageOrientation](#) [getItemOrientation](#) () const
- bool [setItemOrientation](#) (ImageOrientation orientation) const
- [MetaEngine::ImageColorWorkSpace](#) [getItemColorWorkSpace](#) () const
- bool [setItemColorWorkSpace](#) (ImageColorWorkSpace workspace) const
- QDateTime [getItemDateTime](#) () const
- bool [setImageDateTime](#) (const QDateTime &dateTime, bool setDateDigitized=false) const
- QDateTime [getDigitizationDateTime](#) (bool fallbackToCreationTime=false) const
- bool [getItemPreview](#) (QImage &preview) const
- bool [setItemPreview](#) (const QImage &preview) const
- QByteArray [getItemIccProfile](#) () const
- bool [setItemIccProfile](#) (const QByteArray &iccData) const

## Static Public Member Functions

### Static methods

- static bool [initializeExiv2](#) ()
- static bool [supportXmp](#) ()
- static bool [supportJpegXL](#) ()
- static bool [supportBmff](#) ()
- static bool [supportMetadataWriting](#) (const QString &typeMime)
- static QString [Exiv2Version](#) ()

## GPS manipulation methods

- class **MetaEnginePreviews**
- bool [initializeGPSInfo](#) ()
- bool [getGPSInfo](#) (double &altitude, double &latitude, double &longitude) const
- QString [getGPSLatitudeString](#) () const
- QString [getGPSLongitudeString](#) () const
- bool [getGPSLatitudeNumber](#) (double \*const latitude) const
- bool [getGPSLongitudeNumber](#) (double \*const longitude) const
- bool [getGPSAltitude](#) (double \*const altitude) const
- bool [setGPSInfo](#) (const double altitude, const double latitude, const double longitude)
- bool [setGPSInfo](#) (const double \*const altitude, const double latitude, const double longitude)
- bool [setGPSInfo](#) (const double altitude, const QString &latitude, const QString &longitude)
- bool [removeGPSInfo](#) ()
- static void [convertToRational](#) (const double number, long int \*const numerator, long int \*const denominator, const int rounding)
- static void [convertToRationalSmallDenominator](#) (const double number, long int \*const numerator, long int \*const denominator)
- static double [convertDegreeAngleToDouble](#) (double degrees, double minutes, double seconds)
- static QString [convertToGPSCoordinateString](#) (const long int numeratorDegrees, const long int denominatorDegrees, const long int numeratorMinutes, const long int denominatorMinutes, const long int numeratorSeconds, const long int denominatorSeconds, const char directionReference)
- static QString [convertToGPSCoordinateString](#) (const bool isLatitude, double coordinate)
- static bool [convertFromGPSCoordinateString](#) (const QString &coordinate, long int \*const numeratorDegrees, long int \*const denominatorDegrees, long int \*const numeratorMinutes, long int \*const denominatorMinutes, long int \*const numeratorSeconds, long int \*const denominatorSeconds, char \*const directionReference)
- static bool [convertFromGPSCoordinateString](#) (const QString &gpsString, double \*const coordinate)
- static bool [convertToUserPresentableNumbers](#) (const QString &coordinate, int \*const degrees, int \*const minutes, double \*const seconds, char \*const directionReference)
- static void [convertToUserPresentableNumbers](#) (const bool isLatitude, double coordinate, int \*const degrees, int \*const minutes, double \*const seconds, char \*const directionReference)
- bool [setProgramId](#) () const

**File I/O methods**

- void [setFilePath](#) (const QString &path)
- QString [getFilePath](#) () const
- bool [load](#) (const QString &filePath, [Backend](#) \*backend=nullptr)
- bool [loadFromSidecarAndMerge](#) (const QString &filePath)
- bool [save](#) (const QString &filePath, bool setVersion=false) const
- bool [applyChanges](#) (bool setVersion=false) const
- bool [exportChanges](#) (const QString &exvTmpFile) const
- static QString [sidecarFilePathForFile](#) (const QString &path)
- static QString [sidecarPath](#) (const QString &path)
- static QUrl [sidecarUrl](#) (const QUrl &url)
- static QUrl [sidecarUrl](#) (const QString &path)
- static bool [hasSidecar](#) (const QString &path)
- static QString [backendName](#) ([Backend](#) t)

**Comments manipulation methods**

- bool [hasComments](#) () const
- bool [clearComments](#) () const
- QByteArray [getComments](#) () const
- QString [getCommentsDecoded](#) () const
- bool [setComments](#) (const QByteArray &data) const
- static bool [canWriteComment](#) (const QString &filePath)
- static QString [detectLanguageAlt](#) (const QString &value, QString &lang)

**Exif manipulation methods**

- [TagsMap](#) [getStdExifTagsList](#) () const
- [TagsMap](#) [getMakernoteTagsList](#) () const
- bool [hasExif](#) () const
- bool [clearExif](#) () const
- QByteArray [getExifEncoded](#) (bool addExifHeader=false) const
- bool [setExif](#) (const QByteArray &data) const
- QImage [getExifThumbnail](#) (bool fixOrientation) const
- bool [rotateExifQImage](#) (QImage &image, [ImageOrientation](#) orientation) const
- bool [setExifThumbnail](#) (const QImage &thumb) const
- bool [removeExifThumbnail](#) () const
- bool [setTiffThumbnail](#) (const QImage &thumb) const
- QString [getExifComment](#) (bool readDescription=true) const
- QString [getExifTagComment](#) (const char \*exifTagName) const
- bool [setExifComment](#) (const QString &comment, bool writeDescription=true) const
- QString [getExifTagString](#) (const char \*exifTagName, bool escapeCR=true) const
- bool [setExifTagString](#) (const char \*exifTagName, const QString &value) const
- bool [getExifTagLong](#) (const char \*exifTagName, long &val) const
- bool [getExifTagLong](#) (const char \*exifTagName, long &val, int component) const
- bool [setExifTagLong](#) (const char \*exifTagName, long val) const
- bool [setExifTagUShort](#) (const char \*exifTagName, unsigned int val) const
- bool [getExifTagRational](#) (const char \*exifTagName, long int &num, long int &den, int component=0) const
- bool [setExifTagRational](#) (const char \*exifTagName, long int num, long int den) const
- bool [setExifTagURational](#) (const char \*exifTagName, unsigned long int num, unsigned long int den) const
- QByteArray [getExifTagData](#) (const char \*exifTagName) const
- bool [setExifTagData](#) (const char \*exifTagName, const QByteArray &data) const

- QVariant [getExifTagVariant](#) (const char \*exifTagName, bool rationalAsListOfInts=true, bool escapeCR=true, int component=0) const
- bool [setExifTagVariant](#) (const char \*exifTagName, const QVariant &data, bool rationalWantSmall←Denominator=true) const
- bool [removeExifTag](#) (const char \*exifTagName) const
- QString [getExifTagTitle](#) (const char \*exifTagName)
- QString [getExifTagDescription](#) (const char \*exifTagName)
- QString [createExifUserStringFromValue](#) (const char \*exifTagName, const QVariant &val, bool escape←CR=true)
- [MetaEngine::MetaDataMap](#) [getExifTagsDataList](#) (const QStringList &exifKeysFilter=QStringList(), bool invertSelection=false, bool extractBinary=true) const
- static bool [canWriteExif](#) (const QString &filePath)

### IPTC manipulation methods

- [MetaEngine::TagsMap](#) [getIptcTagsList](#) () const
- bool [hasIptc](#) () const
- bool [clearIptc](#) () const
- QByteArray [getIptc](#) (bool addIrbHeader=false) const
- bool [setIptc](#) (const QByteArray &data) const
- QString [getIptcTagString](#) (const char \*iptcTagName, bool escapeCR=true) const
- bool [setIptcTagString](#) (const char \*iptcTagName, const QString &value) const
- QStringList [getIptcTagsStringList](#) (const char \*iptcTagName, bool escapeCR=true) const
- bool [setIptcTagsStringList](#) (const char \*iptcTagName, int maxSize, const QStringList &oldValues, const QStringList &newValues) const
- QByteArray [getIptcTagData](#) (const char \*iptcTagName) const
- bool [setIptcTagData](#) (const char \*iptcTagName, const QByteArray &data) const
- bool [removeIptcTag](#) (const char \*iptcTagName) const
- QString [getIptcTagTitle](#) (const char \*iptcTagName)
- QString [getIptcTagDescription](#) (const char \*iptcTagName)
- [MetaEngine::MetaDataMap](#) [getIptcTagsDataList](#) (const QStringList &iptcKeysFilter=QStringList(), bool invertSelection=false) const
- QStringList [getIptcKeywords](#) () const
- bool [setIptcKeywords](#) (const QStringList &oldKeywords, const QStringList &newKeywords) const
- QStringList [getIptcSubjects](#) () const
- bool [setIptcSubjects](#) (const QStringList &oldSubjects, const QStringList &newSubjects) const
- QStringList [getIptcSubCategories](#) () const
- bool [setIptcSubCategories](#) (const QStringList &oldSubCategories, const QStringList &newSubCategories) const
- static bool [canWriteIptc](#) (const QString &filePath)

### XMP manipulation methods

- [MetaEngine::TagsMap](#) [getXmpTagsList](#) () const
- bool [hasXmp](#) () const
- bool [clearXmp](#) () const
- QByteArray [getXmp](#) () const
- bool [setXmp](#) (const QByteArray &data) const
- QString [getXmpTagString](#) (const char \*xmpTagName, bool escapeCR=true) const
- bool [setXmpTagString](#) (const char \*xmpTagName, const QString &value) const
- bool [setXmpTagString](#) (const char \*xmpTagName, const QString &value, [XmpTagType](#) type) const
- QString [getXmpTagTitle](#) (const char \*xmpTagName)
- QString [getXmpTagDescription](#) (const char \*xmpTagName)

- [MetaEngine::MetaDataMap getXmpTagsDataList](#) (const QStringList &xmpKeysFilter=QStringList(), bool invertSelection=false) const
- [MetaEngine::AltLangMap getXmpTagStringListLangAlt](#) (const char \*xmpTagName, bool escapeCR=true) const
- bool [setXmpTagStringListLangAlt](#) (const char \*xmpTagName, const [MetaEngine::AltLangMap](#) &values) const
- QString [getXmpTagStringLangAlt](#) (const char \*xmpTagName, const QString &langAlt, bool escapeCR) const
- bool [setXmpTagStringLangAlt](#) (const char \*xmpTagName, const QString &value, const QString &langAlt) const
- QStringList [getXmpTagStringSeq](#) (const char \*xmpTagName, bool escapeCR=true) const
- bool [setXmpTagStringSeq](#) (const char \*xmpTagName, const QStringList &seq) const
- QStringList [getXmpTagStringBag](#) (const char \*xmpTagName, bool escapeCR) const
- bool [setXmpTagStringBag](#) (const char \*xmpTagName, const QStringList &bag) const
- bool [addToXmpTagStringBag](#) (const char \*xmpTagName, const QStringList &entriesToAdd) const
- bool [removeFromXmpTagStringBag](#) (const char \*xmpTagName, const QStringList &entriesToRemove) const
- QVariant [getXmpTagVariant](#) (const char \*xmpTagName, bool rationalAsListOfInts=true, bool stringEscape↵CR=true) const
- QStringList [getXmpKeywords](#) () const
- bool [setXmpKeywords](#) (const QStringList &newKeywords) const
- bool [removeXmpKeywords](#) (const QStringList &keywordsToRemove)
- QStringList [getXmpSubjects](#) () const
- bool [setXmpSubjects](#) (const QStringList &newSubjects) const
- bool [removeXmpSubjects](#) (const QStringList &subjectsToRemove)
- QStringList [getXmpSubCategories](#) () const
- bool [setXmpSubCategories](#) (const QStringList &newSubCategories) const
- bool [removeXmpSubCategories](#) (const QStringList &categoriesToRemove)
- bool [removeXmpTag](#) (const char \*xmpTagName, bool family=false) const
- static bool [canWriteXmp](#) (const QString &filePath)
- static bool [registerXmpNameSpace](#) (const QString &uri, const QString &prefix)
- static bool [unregisterXmpNameSpace](#) (const QString &uri)

## 6.1079.1 Member Typedef Documentation

### 6.1079.1.1 AltLangMap

```
typedef QMap<QString, QString> Digikam::MetaEngine::AltLangMap
```

A map used to store a list of Alternative Language values. The map key is the language code following RFC3066 notation (like "fr-FR" for French), and the map value the text.

### 6.1079.1.2 MetaDataMap

```
typedef QMap<QString, QString> Digikam::MetaEngine::MetaDataMap
```

A map used to store Tags Key and Tags Value.

### 6.1079.1.3 TagsMap

```
typedef QMap<QString, QStringList> Digikam::MetaEngine::TagsMap
```

A map used to store Tags Key and a list of Tags properties :

- name,
- title,
- description.



## 6.1079.2 Member Enumeration Documentation

### 6.1079.2.1 Backend

enum `Digikam::MetaEngine::Backend`

Metadata Backend used to populate information.

Enumerator

Exiv2Backend	Default backend used by <a href="#">MetaEngine</a> .
LibRawBackend	<a href="#">DMetadadata</a> only.
LibHeifBackend	<a href="#">DMetadadata</a> only.
ImageMagickBackend	<a href="#">DMetadadata</a> only.
FFMpegBackend	<a href="#">DMetadadata</a> only.
ExifToolBackend	<a href="#">DMetadadata</a> only.
VideoMergeBackend	<a href="#">DMetadadata</a> only.
NoBackend	No backend used (aka file cannot be read).

### 6.1079.2.2 ImageColorWorkSpace

enum `Digikam::MetaEngine::ImageColorWorkSpace`

The item color workspace values given by Exif metadata.

### 6.1079.2.3 ImageOrientation

enum `Digikam::MetaEngine::ImageOrientation`

The item orientation values given by Exif metadata.

### 6.1079.2.4 MetadataWritingMode

enum `Digikam::MetaEngine::MetadataWritingMode`

The item metadata writing mode, between item file metadata and XMP sidecar file, depending on the context.

See also

[MetadataWritingMode\(\)](#), [metadataWritingMode\(\)](#)

Enumerator

WRITE_TO_FILE_ONLY	Write metadata to item file only.
WRITE_TO_SIDECAR_ONLY	Write metadata to sidecar file only.
WRITE_TO_SIDECAR_AND_FILE	Write metadata to item and sidecar files.
WRITE_TO_SIDECAR_ONLY_FOR_READ_ONLY_FILES	Write metadata to sidecar file only for read only items such as RAW files for example.

### 6.1079.2.5 XmpTagType

```
enum Digikam::MetaEngine::XmpTagType
```

Xmp tag types, used by setXmpTag, only first three types are used

## 6.1079.3 Constructor & Destructor Documentation

### 6.1079.3.1 MetaEngine() [1/3]

```
Digikam::MetaEngine::MetaEngine ( )
```

Standard constructor.

### 6.1079.3.2 MetaEngine() [2/3]

```
Digikam::MetaEngine::MetaEngine (
    const MetaEngineData & data ) [explicit]
```

Constructor to load from parsed data.

### 6.1079.3.3 MetaEngine() [3/3]

```
Digikam::MetaEngine::MetaEngine (
    const QString & filePath ) [explicit]
```

Constructor to Load Metadata from item file.

### 6.1079.3.4 ~MetaEngine()

```
Digikam::MetaEngine::~MetaEngine ( ) [virtual]
```

Standard destructor

## 6.1079.4 Member Function Documentation

### 6.1079.4.1 addToXmpTagStringBag()

```
bool Digikam::MetaEngine::addToXmpTagStringBag (
    const char * xmpTagName,
    const QStringList & entriesToAdd ) const
```

Set an Xmp tag content using a list of strings defined by the 'entriesToAdd' parameter. The existing entries are preserved. The method will compare all new with all already existing entries to prevent duplicates in the item. Return true if the entries have been added to metadata.

#### 6.1079.4.2 applyChanges()

```
bool Digikam::MetaEngine::applyChanges (
    bool setVersion = false ) const
```

The same than [save\(\)](#) method, but it apply on current item. Return true if metadata have been saved into file.

#### 6.1079.4.3 backendName()

```
QString Digikam::MetaEngine::backendName (
    Backend t ) [static]
```

Return a string of backend name used to parse metadata from file. See Backend enum for details.

#### 6.1079.4.4 canWriteComment()

```
bool Digikam::MetaEngine::canWriteComment (
    const QString & filePath ) [static]
```

Return 'true' if Comments can be written in file.

#### 6.1079.4.5 canWriteExif()

```
bool Digikam::MetaEngine::canWriteExif (
    const QString & filePath ) [static]
```

Return 'true' if Exif can be written in file.

#### 6.1079.4.6 canWriteIptc()

```
bool Digikam::MetaEngine::canWriteIptc (
    const QString & filePath ) [static]
```

Return 'true' if Iptc can be written in file.

#### 6.1079.4.7 canWriteXmp()

```
bool Digikam::MetaEngine::canWriteXmp (
    const QString & filePath ) [static]
```

Return 'true' if Xmp can be written in file.

#### 6.1079.4.8 clearComments()

```
bool Digikam::MetaEngine::clearComments ( ) const
```

Clear the Comments metadata container in memory.

#### 6.1079.4.9 clearExif()

```
bool Digikam::MetaEngine::clearExif ( ) const
```

Clear the Exif metadata container in memory.

#### 6.1079.4.10 clearIptc()

```
bool Digikam::MetaEngine::clearIptc ( ) const
```

Clear the Iptc metadata container in memory.

#### 6.1079.4.11 clearXmp()

```
bool Digikam::MetaEngine::clearXmp ( ) const
```

Clear the Xmp metadata container in memory.

#### 6.1079.4.12 convertDegreeAngleToDouble()

```
double Digikam::MetaEngine::convertDegreeAngleToDouble (
    double degrees,
    double minutes,
    double seconds ) [static]
```

Converts degrees values as a double representation. This code take a care about hemisphere position.

#### 6.1079.4.13 convertFromGPSCoordinateString() [1/2]

```
bool Digikam::MetaEngine::convertFromGPSCoordinateString (
    const QString & coordinate,
    long int *const numeratorDegrees,
    long int *const denominatorDegrees,
    long int *const numeratorMinutes,
    long int *const denominatorMinutes,
    long int *const numeratorSeconds,
    long int *const denominatorSeconds,
    char *const directionReference ) [static]
```

Converts a GPSCoordinate string as defined by XMP to three rationals and the direction reference. Returns true if the conversion was successful. If minutes is given in the fractional form, a denominator of 1000000 for the minutes will be used.

#### 6.1079.4.14 convertFromGPSCoordinateString() [2/2]

```
bool Digikam::MetaEngine::convertFromGPSCoordinateString (
    const QString & gpsString,
    double *const coordinate ) [static]
```

Convert a GPSCoordinate string as defined by XMP to a double floating point number in degrees where the sign determines the direction ref (North + / South - ; East + / West -). Returns true if the conversion was successful.

**6.1079.4.15 convertToGPSCoordinateString() [1/2]**

```
QString Digikam::MetaEngine::convertToGPSCoordinateString (
    const bool isLatitude,
    double coordinate ) [static]
```

Converts a GPS position stored as double floating point number in degrees to the form described as GPSCoordinate in the XMP specification.

**6.1079.4.16 convertToGPSCoordinateString() [2/2]**

```
QString Digikam::MetaEngine::convertToGPSCoordinateString (
    const long int numeratorDegrees,
    const long int denominatorDegrees,
    const long int numeratorMinutes,
    const long int denominatorMinutes,
    const long int numeratorSeconds,
    const long int denominatorSeconds,
    const char directionReference ) [static]
```

Converts a GPS position stored as rationals in Exif to the form described as GPSCoordinate in the XMP specification, either in the form "256,45,34N" or "256,45.566667N" Precision: A second at sea level measures 30m for our purposes, a minute 1800m. (for more details, see [https://en.wikipedia.org/wiki/Geographic\\_coordinate\\_system](https://en.wikipedia.org/wiki/Geographic_coordinate_system)) This means with a decimal precision of 8 for minutes we get +/-0,018mm. (if I calculated correctly)

**6.1079.4.17 convertToRational()**

```
void Digikam::MetaEngine::convertToRational (
    const double number,
    long int *const numerator,
    long int *const denominator,
    const int rounding ) [static]
```

This method converts 'number' to a rational value, returned in the 'numerator' and 'denominator' parameters. Set the precision using 'rounding' parameter. Use this method if you want to retrieve a most exact rational for a number without further properties, without any requirements to the denominator.

**6.1079.4.18 convertToRationalSmallDenominator()**

```
void Digikam::MetaEngine::convertToRationalSmallDenominator (
    const double number,
    long int *const numerator,
    long int *const denominator ) [static]
```

This method convert a 'number' to a rational value, returned in 'numerator' and 'denominator' parameters. This method will be able to retrieve a rational number from a double - if you constructed your double with 1.0 / 4786.0, this method will retrieve 1 / 4786. If your number is not expected to be rational, use the method above which is just as exact with rounding = 4 and more exact with rounding > 4.

#### 6.1079.4.19 `convertToUserPresentableNumbers()` [1/2]

```
void Digikam::MetaEngine::convertToUserPresentableNumbers (
    const bool isLatitude,
    double coordinate,
    int *const degrees,
    int *const minutes,
    double *const seconds,
    char *const directionReference ) [static]
```

Converts a double floating point number to user presentable numbers, integer degrees and minutes and double floating point seconds, and a direction reference ('N' or 'S', 'E' or 'W'). The method needs to know for the direction reference if the latitude or the longitude is meant by the double parameter.

#### 6.1079.4.20 `convertToUserPresentableNumbers()` [2/2]

```
bool Digikam::MetaEngine::convertToUserPresentableNumbers (
    const QString & coordinate,
    int *const degrees,
    int *const minutes,
    double *const seconds,
    char *const directionReference ) [static]
```

Converts a GPSCoordinate string to user presentable numbers, integer degrees and minutes and double floating point seconds, and a direction reference ('N' or 'S', 'E' or 'W')

#### 6.1079.4.21 `createExifUserStringFromValue()`

```
QString Digikam::MetaEngine::createExifUserStringFromValue (
    const char * exifTagName,
    const QVariant & val,
    bool escapeCR = true )
```

Takes a QVariant value as it could have been retrieved by `getExifTagVariant` with the given `exifTagName`, and returns its value properly converted to a string (including translations from Exiv2). This is equivalent to calling `getExifTagString` directly. If `escapeCR` is true CR characters will be removed from the result.

#### 6.1079.4.22 `detectLanguageAlt()`

```
QString Digikam::MetaEngine::detectLanguageAlt (
    const QString & value,
    QString & lang ) [static]
```

Language Alternative autodetection. Return a QString without language alternative header. Header is saved into 'lang'. If no language alternative is found, value is returned as well and 'lang' is set to a null string.

#### 6.1079.4.23 `Exiv2Version()`

```
QString Digikam::MetaEngine::Exiv2Version ( ) [static]
```

Return a string version of Exiv2 release in format "major.minor.patch"

#### 6.1079.4.24 exportChanges()

```
bool Digikam::MetaEngine::exportChanges (
    const QString & exvTmpFile ) const
```

Export metadata to a temporary EXV file container. 'exvTmpFile' is the path to the temporary EXV container to create.

#### 6.1079.4.25 getComments()

```
QByteArray Digikam::MetaEngine::getComments ( ) const
```

Return a Qt byte array copy of Comments container get from current item. Comments are JFIF section of JPEG images. Look Exiv2 API for more information. Return a null Qt byte array if there is no Comments metadata in memory.

#### 6.1079.4.26 getCommentsDecoded()

```
QString Digikam::MetaEngine::getCommentsDecoded ( ) const
```

Return a Qt string object of Comments from current item decoded using the 'detectEncodingAndDecode()' method. Return a null string if there is no Comments metadata available.

#### 6.1079.4.27 getDigitizationDateTime()

```
QDateTime Digikam::MetaEngine::getDigitizationDateTime (
    bool fallbackToCreationTime = false ) const
```

Return the digitization time stamp of the item. First Exif information is checked, then IPTC. If no digitization time stamp is found, [getItemDateime\(\)](#) is called if `fallbackToCreationTime` is true, or a null QDateTime is returned if `fallbackToCreationTime` is false.

#### 6.1079.4.28 getExifComment()

```
QString Digikam::MetaEngine::getExifComment (
    bool readDescription = true ) const
```

Return a QString copy of Exif user comments. Return a null string if user comments cannot be found.

#### 6.1079.4.29 getExifEncoded()

```
QByteArray Digikam::MetaEngine::getExifEncoded (
    bool addExifHeader = false ) const
```

Returns the exif data encoded to a QByteArray in a form suitable for storage in a JPEG image. Note that this encoding is a lossy operation.

Set true 'addExifHeader' parameter to add an Exif header to Exif metadata. Returns a null Qt byte array if there is no Exif metadata in memory.

#### 6.1079.4.30 `getExifTagComment()`

```
QString Digikam::MetaEngine::getExifTagComment (
    const char * exifTagName ) const
```

Return a Exif tag comment like a string. Return a null string if user comments cannot be found.

#### 6.1079.4.31 `getExifTagData()`

```
QByteArray Digikam::MetaEngine::getExifTagData (
    const char * exifTagName ) const
```

Get an Exif tag content like a bytes array. Return an empty bytes array if Exif tag cannot be found.

#### 6.1079.4.32 `getExifTagDescription()`

```
QString Digikam::MetaEngine::getExifTagDescription (
    const char * exifTagName )
```

Return the Exif Tag description or a null string.

#### 6.1079.4.33 `getExifTagLong()` [1/2]

```
bool Digikam::MetaEngine::getExifTagLong (
    const char * exifTagName,
    long & val ) const
```

Get an Exif tag content like a long value. Return true if Exif tag be found.

#### 6.1079.4.34 `getExifTagLong()` [2/2]

```
bool Digikam::MetaEngine::getExifTagLong (
    const char * exifTagName,
    long & val,
    int component ) const
```

Get an Exif tag content like a long value. Return true if Exif tag be found.

#### 6.1079.4.35 `getExifTagRational()`

```
bool Digikam::MetaEngine::getExifTagRational (
    const char * exifTagName,
    long int & num,
    long int & den,
    int component = 0 ) const
```

Get the 'component' index of an Exif tags content like a rational value. 'num' and 'den' are the numerator and the denominator of the rational value. Return true if Exif tag be found.



**6.1079.4.36 getExifTagsDataList()**

```
MetaEngine::MetaDataMap Digikam::MetaEngine::getExifTagsDataList (
    const QStringList & exifKeysFilter = QStringList(),
    bool invertSelection = false,
    bool extractBinary = true ) const
```

Return a map of Exif tags name/value found in metadata sorted by Exif keys given by 'exifKeysFilter'.

'exifKeysFilter' is a QStringList of Exif keys. For example, if you use the string list given below:

```
"Iop" "Thumbnail" "Image" "Photo"
```

List can be empty to not filter output.

... this method will return a map of all Exif tags which :

- include "Iop", or "Thumbnail", or "Image", or "Photo" in the Exif tag keys if 'invertSelection' is false.
- not include "Iop", or "Thumbnail", or "Image", or "Photo" in the Exif tag keys if 'invertSelection' is true. if 'extractBinary' is true, tags with undefined types of data are extracted (default), else contents is replaced by "Binary data ... bytes". Take a care as large binary data as original RAW data from DNG container can be huge and listing Exif tags from GUI can take a while.

**6.1079.4.37 getExifTagString()**

```
QString Digikam::MetaEngine::getExifTagString (
    const char * exifTagName,
    bool escapeCR = true ) const
```

Get an Exif tags content like a string. If 'escapeCR' parameter is true, the CR characters will be removed. If Exif tag cannot be found a null string is returned.

**6.1079.4.38 getExifTagTitle()**

```
QString Digikam::MetaEngine::getExifTagTitle (
    const char * exifTagName )
```

Return the Exif Tag title or a null string.

**6.1079.4.39 getExifTagVariant()**

```
QVariant Digikam::MetaEngine::getExifTagVariant (
    const char * exifTagName,
    bool rationalAsListofInts = true,
    bool escapeCR = true,
    int component = 0 ) const
```

Get an Exif tags content as a QVariant. Returns a null QVariant if the Exif tag cannot be found. For string and integer values the matching QVariant types will be used, for date and time values QVariant::DateTime. Rationals will be returned as QVariant::List with two integer QVariants (numerator, denominator) if rationalAsListofInts is true, as double if rationalAsListofInts is false. An exif tag of numerical type may contain more than one value; set component to the desired index.

#### 6.1079.4.40 getExifThumbnail()

```
QImage Digikam::MetaEngine::getExifThumbnail (
    bool fixOrientation ) const
```

Return a QImage copy of Exif thumbnail image. Return a null image if thumbnail cannot be found. The 'fix↔Orientation' parameter will rotate automatically the thumbnail if Exif orientation tags information are attached with thumbnail.

#### 6.1079.4.41 getFilePath()

```
QString Digikam::MetaEngine::getFilePath ( ) const
```

Return the file path of current item.

#### 6.1079.4.42 getGPSAltitude()

```
bool Digikam::MetaEngine::getGPSAltitude (
    double *const altitude ) const
```

Get GPS altitude information, in meters, relative to sea level (positive sign above sea level)

#### 6.1079.4.43 getGPSInfo()

```
bool Digikam::MetaEngine::getGPSInfo (
    double & altitude,
    double & latitude,
    double & longitude ) const
```

Get all GPS location information set in item. Return true if all information can be found.

#### 6.1079.4.44 getGPSLatitudeNumber()

```
bool Digikam::MetaEngine::getGPSLatitudeNumber (
    double *const latitude ) const
```

Get GPS location information set in the item, as a double floating point number as in degrees where the sign determines the direction ref (North + / South - ; East + / West -). Returns true if the information is available.

#### 6.1079.4.45 getGPSLatitudeString()

```
QString Digikam::MetaEngine::getGPSLatitudeString ( ) const
```

Get GPS location information set in the item, in the GPSCoordinate format as described in the XMP specification. Returns a null string in the information cannot be found.

**6.1079.4.46 getIptc()**

```
QByteArray Digikam::MetaEngine::getIptc (
    bool addIrbHeader = false ) const
```

Return a Qt byte array copy of Iptc container get from current item. Set true 'addIrbHeader' parameter to add an Irb header to Iptc metadata. Return a null Qt byte array if there is no Iptc metadata in memory.

**6.1079.4.47 getIptcKeywords()**

```
QStringList Digikam::MetaEngine::getIptcKeywords ( ) const
```

Return a strings list of Iptc keywords from item. Return an empty list if no keyword are set.

**6.1079.4.48 getIptcSubCategories()**

```
QStringList Digikam::MetaEngine::getIptcSubCategories ( ) const
```

Return a strings list of Iptc sub-categories from item. Return an empty list if no sub-category are set.

**6.1079.4.49 getIptcSubjects()**

```
QStringList Digikam::MetaEngine::getIptcSubjects ( ) const
```

Return a strings list of Iptc subjects from item. Return an empty list if no subject are set.

**6.1079.4.50 getIptcTagData()**

```
QByteArray Digikam::MetaEngine::getIptcTagData (
    const char * iptcTagName ) const
```

Get an Iptc tag content as a bytes array. Return an empty bytes array if Iptc tag cannot be found.

**6.1079.4.51 getIptcTagDescription()**

```
QString Digikam::MetaEngine::getIptcTagDescription (
    const char * iptcTagName )
```

Return the Iptc Tag description or a null string.

**6.1079.4.52 getIptcTagsDataList()**

```
MetaEngine::MetaDataMap Digikam::MetaEngine::getIptcTagsDataList (
    const QStringList & iptcKeysFilter = QStringList(),
    bool invertSelection = false ) const
```

Return a map of Iptc tags name/value found in metadata sorted by Iptc keys given by 'iptcKeysFilter'.

'iptcKeysFilter' is a QStringList of Iptc keys. For example, if you use the string list given below:

"Envelope" "Application2"

List can be empty to not filter output.

... this method will return a map of all Iptc tags which :

- include "Envelope", or "Application2" in the Iptc tag keys if 'invertSelection' is false.
- not include "Envelope", or "Application2" in the Iptc tag keys if 'invertSelection' is true.

**6.1079.4.53 getIptcTagsList()**

```
MetaEngine::TagsMap Digikam::MetaEngine::getIptcTagsList ( ) const
```

Return a map of all standard Iptc tags supported by Exiv2.

**6.1079.4.54 getIptcTagsStringList()**

```
QStringList Digikam::MetaEngine::getIptcTagsStringList (
    const char * iptcTagName,
    bool escapeCR = true ) const
```

Returns a strings list with of multiple Iptc tags from the item. Return an empty list if no tag is found. Get the values of all IPTC tags with the given tag name in a string list. (In Iptc, there can be multiple tags with the same name) If the 'escapeCR' parameter is true, the CR characters will be removed. If no tag can be found an empty list is returned.

**6.1079.4.55 getIptcTagString()**

```
QString Digikam::MetaEngine::getIptcTagString (
    const char * iptcTagName,
    bool escapeCR = true ) const
```

Get an Iptc tag content like a string. If 'escapeCR' parameter is true, the CR characters will be removed. If Iptc tag cannot be found a null string is returned.

**6.1079.4.56 getIptcTagTitle()**

```
QString Digikam::MetaEngine::getIptcTagTitle (
    const char * iptcTagName )
```

Return the Iptc Tag title or a null string.

#### 6.1079.4.57 getItemColorWorkSpace()

```
MetaEngine::ImageColorWorkSpace Digikam::MetaEngine::getItemColorWorkSpace ( ) const
```

Return the item color-space set in Exif metadata. The makernotes of item are also parsed to get this information. See ImageColorWorkSpace values for details.

#### 6.1079.4.58 getItemDateTime()

```
QDateTime Digikam::MetaEngine::getItemDateTime ( ) const
```

Return the time stamp of item. Exif information are check in first, IPTC in second if item don't have Exif information. If no time stamp is found, a null date is returned.

#### 6.1079.4.59 getItemDimensions()

```
QSize Digikam::MetaEngine::getItemDimensions ( ) const
```

Return the size of item in pixels using Exif tags. Return a null dimension if size cannot be found.

#### 6.1079.4.60 getItemIccProfile()

```
QByteArray Digikam::MetaEngine::getItemIccProfile ( ) const
```

Get image ICC profile.

#### 6.1079.4.61 getItemOrientation()

```
MetaEngine::ImageOrientation Digikam::MetaEngine::getItemOrientation ( ) const
```

Return the item orientation set in Exif metadata. The makernotes of item are also parsed to get this information. See ImageOrientation values for details.

#### 6.1079.4.62 getItemPreview()

```
bool Digikam::MetaEngine::getItemPreview (
    QImage & preview ) const
```

Return a QImage copy of Iptc preview image. Return a null item if preview cannot be found.

#### 6.1079.4.63 getMakernoteTagsList()

```
MetaEngine::TagsMap Digikam::MetaEngine::getMakernoteTagsList ( ) const
```

Return a map of all non-standard Exif tags (makernotes) supported by Exiv2.

#### 6.1079.4.64 `getMimeType()`

```
QString Digikam::MetaEngine::getMimeType ( ) const
```

Returns the mime type of this item. The information is read from the file; see the docs for [getPixelSize\(\)](#) to know when it is available.

#### 6.1079.4.65 `getPixelSize()`

```
QSize Digikam::MetaEngine::getPixelSize ( ) const
```

Returns the pixel size of the current item. This information is read from the file, not from the metadata. The returned `QSize` is valid if the [MetaEngine](#) object was *constructed* by reading a file or item data; the information is not available when the object was created from [MetaEngineData](#). Note that in the Exif or XMP metadata, there may be fields describing the item size. These fields are not accessed by this method. When replacing the metadata with `setData()`, the metadata may change; this information always keeps referring to the file it was initially read from.

#### 6.1079.4.66 `getStdExifTagsList()`

```
MetaEngine::TagsMap Digikam::MetaEngine::getStdExifTagsList ( ) const
```

Return a map of all standard Exif tags supported by Exiv2.

#### 6.1079.4.67 `getXmp()`

```
QByteArray Digikam::MetaEngine::getXmp ( ) const
```

Return a Qt byte array copy of XMP container get from current item. Return a null Qt byte array if there is no Xmp metadata in memory.

#### 6.1079.4.68 `getXmpKeywords()`

```
QStringList Digikam::MetaEngine::getXmpKeywords ( ) const
```

Return a strings list of Xmp keywords from item. Return an empty list if no keyword are set.

#### 6.1079.4.69 `getXmpSubCategories()`

```
QStringList Digikam::MetaEngine::getXmpSubCategories ( ) const
```

Return a strings list of Xmp sub-categories from item. Return an empty list if no sub-category are set.

#### 6.1079.4.70 `getXmpSubjects()`

```
QStringList Digikam::MetaEngine::getXmpSubjects ( ) const
```

Return a strings list of Xmp subjects from item. Return an empty list if no subject are set.

#### 6.1079.4.71 getXmpTagDescription()

```
QString Digikam::MetaEngine::getXmpTagDescription (
    const char * xmpTagName )
```

Return the Xmp Tag description or a null string.

#### 6.1079.4.72 getXmpTagsDataList()

```
MetaEngine::MetaDataMap Digikam::MetaEngine::getXmpTagsDataList (
    const QStringList & xmpKeysFilter = QStringList(),
    bool invertSelection = false ) const
```

Return a map of Xmp tags name/value found in metadata sorted by Xmp keys given by 'xmpKeysFilter'.

'xmpKeysFilter' is a QStringList of Xmp keys. For example, if you use the string list given below:

```
"dc" // Dublin Core schema. "xmp" // Standard Xmp schema.
```

List can be empty to not filter output.

... this method will return a map of all Xmp tags which :

- include "dc", or "xmp" in the Xmp tag keys if 'invertSelection' is false.
- not include "dc", or "xmp" in the Xmp tag keys if 'invertSelection' is true.

#### 6.1079.4.73 getXmpTagsList()

```
MetaEngine::TagsMap Digikam::MetaEngine::getXmpTagsList ( ) const
```

Return a map of all standard Xmp tags supported by Exiv2.

#### 6.1079.4.74 getXmpTagString()

```
QString Digikam::MetaEngine::getXmpTagString (
    const char * xmpTagName,
    bool escapeCR = true ) const
```

Get a Xmp tag content like a string. If 'escapeCR' parameter is true, the CR characters will be removed. If Xmp tag cannot be found a null string is returned.

#### 6.1079.4.75 getXmpTagStringBag()

```
QStringList Digikam::MetaEngine::getXmpTagStringBag (
    const char * xmpTagName,
    bool escapeCR ) const
```

Get a Xmp tag content like a bag of strings. If 'escapeCR' parameter is true, the CR characters will be removed from strings. If Xmp tag cannot be found a null string list is returned.

**6.1079.4.76 getXmpTagStringLangAlt()**

```
QString Digikam::MetaEngine::getXmpTagStringLangAlt (
    const char * xmpTagName,
    const QString & langAlt,
    bool escapeCR ) const
```

Get a Xmp tag content like a string set with an alternative language header 'langAlt' (like "fr-FR" for French - RFC3066 notation) If 'escapeCR' parameter is true, the CR characters will be removed. If Xmp tag cannot be found a null string is returned.

**6.1079.4.77 getXmpTagStringListLangAlt()**

```
MetaEngine::AltLangMap Digikam::MetaEngine::getXmpTagStringListLangAlt (
    const char * xmpTagName,
    bool escapeCR = true ) const
```

Get all redundant Alternative Language Xmp tags content like a map. See AltLangMap class description for details. If 'escapeCR' parameter is true, the CR characters will be removed from strings. If Xmp tag cannot be found a null string list is returned.

**6.1079.4.78 getXmpTagStringSeq()**

```
QStringList Digikam::MetaEngine::getXmpTagStringSeq (
    const char * xmpTagName,
    bool escapeCR = true ) const
```

Get a Xmp tag content like a sequence of strings. If 'escapeCR' parameter is true, the CR characters will be removed from strings. If Xmp tag cannot be found a null string list is returned.

**6.1079.4.79 getXmpTagTitle()**

```
QString Digikam::MetaEngine::getXmpTagTitle (
    const char * xmpTagName )
```

Return the Xmp Tag title or a null string.

**6.1079.4.80 getXmpTagVariant()**

```
QVariant Digikam::MetaEngine::getXmpTagVariant (
    const char * xmpTagName,
    bool rationalAsListOfInts = true,
    bool stringEscapeCR = true ) const
```

Get an Xmp tag content as a QVariant. Returns a null QVariant if the Xmp tag cannot be found. For string and integer values the matching QVariant types will be used, for date and time values QVariant::DateTime. Rationals will be returned as QVariant::List with two integer QVariants (numerator, denominator) if rationalAsListOfInts is true, as double if rationalAsListOfInts is false. Arrays (ordered, unordered, alternative) are returned as type QStringList. LangAlt values will have type Map (QMap<QString, QVariant>) with the language code as key and the contents as value, of type String.



**6.1079.4.81 hasComments()**

```
bool Digikam::MetaEngine::hasComments ( ) const
```

Return 'true' if metadata container in memory as Comments.

**6.1079.4.82 hasExif()**

```
bool Digikam::MetaEngine::hasExif ( ) const
```

Return 'true' if metadata container in memory as Exif.

**6.1079.4.83 hasIptc()**

```
bool Digikam::MetaEngine::hasIptc ( ) const
```

Return 'true' if metadata container in memory as Iptc.

**6.1079.4.84 hasSidecar()**

```
bool Digikam::MetaEngine::hasSidecar (
    const QString & path ) [static]
```

Performs a QFileInfo based check if the given local file has a sidecar.

**6.1079.4.85 hasXmp()**

```
bool Digikam::MetaEngine::hasXmp ( ) const
```

Return 'true' if metadata container in memory as Xmp.

**6.1079.4.86 initializeExiv2()**

```
bool Digikam::MetaEngine::initializeExiv2 ( ) [static]
```

Return true if Exiv2 library initialization is done properly. This method must be called before using libMetaEngine with multithreading. It initialize several non re-entrancy code from Adobe XMP SDK, and register a function to cleanup automatically all XMP SDK memory allocation. See Bug #166424 for details.

**6.1079.4.87 initializeGPSInfo()**

```
bool Digikam::MetaEngine::initializeGPSInfo ( )
```

Make sure all static required GPS EXIF and XMP tags exist

### 6.1079.4.88 isEmpty()

```
bool Digikam::MetaEngine::isEmpty ( ) const
```

Return 'true' if metadata container in memory as no Comments, Exif, Iptc, and Xmp.

### 6.1079.4.89 load()

```
bool Digikam::MetaEngine::load (
    const QString & filePath,
    Backend * backend = nullptr )
```

Load all metadata (Exif, Iptc, Xmp, and JFIF Comments) from a picture (JPEG, RAW, TIFF, PNG, DNG, etc...). Return true if metadata have been loaded successfully from file. If backend is non null, return the backend used to populate metadata (Exiv2). See Backend enum for details.

### 6.1079.4.90 loadFromData()

```
bool Digikam::MetaEngine::loadFromData (
    const QByteArray & imgData )
```

Load all metadata (Exif, Iptc, Xmp, and JFIF Comments) from a byte array. Return true if metadata have been loaded successfully from item data.

### 6.1079.4.91 loadFromDataAndMerge()

```
bool Digikam::MetaEngine::loadFromDataAndMerge (
    const QByteArray & imgData,
    const QStringList & exclude = QStringList () )
```

Load and merge metadata (Exif, Iptc and Xmp) from a byte array. Use 'exclude' to remove Exif tags from the 'imgData' that will not be merged. Return true if metadata have been loaded and merged successfully from item data.

### 6.1079.4.92 loadFromSidecarAndMerge()

```
bool Digikam::MetaEngine::loadFromSidecarAndMerge (
    const QString & filePath )
```

Load metadata from a sidecar file and merge. Return true if metadata have been loaded successfully from file.

### 6.1079.4.93 metadataWritingMode()

```
int Digikam::MetaEngine::metadataWritingMode ( ) const
```

Return the metadata writing mode.

#### Returns

Metadata writing mode as defined by the [MetadataWritingMode](#) enum.

#### See also

[MetadataWritingMode](#), [setMetadataWritingMode\(\)](#)

#### 6.1079.4.94 readWithExifTool()

```
bool Digikam::MetaEngine::readWithExifTool ( ) const
```

Return true if reading metadata operations with ExifTool is enabled.

#### 6.1079.4.95 registerXmpNameSpace()

```
bool Digikam::MetaEngine::registerXmpNameSpace (
    const QString & uri,
    const QString & prefix ) [static]
```

Register a namespace which Exiv2 doesn't know yet. This is only needed when new Xmp properties are added manually. 'uri' is the namespace url and 'prefix' the string used to construct new Xmp key (ex. "Xmp.digiKam.tag←List").

#### Note

If the Xmp metadata is read from an item, namespaces are decoded and registered by Exiv2 at the same time.

#### 6.1079.4.96 removeExifTag()

```
bool Digikam::MetaEngine::removeExifTag (
    const char * exifTagName ) const
```

Remove the Exif tag 'exifTagName' from Exif metadata. Return true if tag is removed successfully or if no tag was present.

#### 6.1079.4.97 removeExifThumbnail()

```
bool Digikam::MetaEngine::removeExifThumbnail ( ) const
```

Remove the Exif Thumbnail from the item

#### 6.1079.4.98 removeFromXmpTagStringBag()

```
bool Digikam::MetaEngine::removeFromXmpTagStringBag (
    const char * xmpTagName,
    const QStringList & entriesToRemove ) const
```

Remove those Xmp tag entries that are listed in entriesToRemove from the entries in metadata. Return true if tag entries are no longer contained in metadata. All other entries are preserved.

#### 6.1079.4.99 removeGPSInfo()

```
bool Digikam::MetaEngine::removeGPSInfo ( )
```

Remove all Exif tags relevant of GPS location information. Return true if all tags have been removed successfully in metadata.

#### 6.1079.4.100 removeIptcTag()

```
bool Digikam::MetaEngine::removeIptcTag (
    const char * iptcTagName ) const
```

Remove the all instance of Iptc tags 'iptcTagName' from Iptc metadata. Return true if all tags have been removed successfully (or none were present).

#### 6.1079.4.101 removeXmpKeywords()

```
bool Digikam::MetaEngine::removeXmpKeywords (
    const QStringList & keywordsToRemove )
```

Remove those Xmp keywords that are listed in keywordsToRemove from the keywords in metadata. Return true if keywords are no longer contained in metadata.

#### 6.1079.4.102 removeXmpSubCategories()

```
bool Digikam::MetaEngine::removeXmpSubCategories (
    const QStringList & categoriesToRemove )
```

Remove those Xmp sub-categories that are listed in categoriesToRemove from the sub-categories in metadata. Return true if subjects are no longer contained in metadata.

#### 6.1079.4.103 removeXmpSubjects()

```
bool Digikam::MetaEngine::removeXmpSubjects (
    const QStringList & subjectsToRemove )
```

Remove those Xmp subjects that are listed in subjectsToRemove from the subjects in metadata. Return true if subjects are no longer contained in metadata.

#### 6.1079.4.104 removeXmpTag()

```
bool Digikam::MetaEngine::removeXmpTag (
    const char * xmpTagName,
    bool family = false ) const
```

Remove the Xmp tag 'xmpTagName' from Xmp metadata. Return true if tag is removed successfully or if no tag was present.

#### 6.1079.4.105 rotateExifQImage()

```
bool Digikam::MetaEngine::rotateExifQImage (
    QImage & image,
    ImageOrientation orientation ) const
```

Fix orientation of a QImage image accordingly with Exif orientation tag. Return true if image is rotated, else false.

**6.1079.4.106 save()**

```
bool Digikam::MetaEngine::save (
    const QString & filePath,
    bool setVersion = false ) const
```

Save all metadata to a file. This one can be different than original picture to perform transfer operation Return true if metadata have been saved into file.

**6.1079.4.107 setComments()**

```
bool Digikam::MetaEngine::setComments (
    const QByteArray & data ) const
```

Set the Comments data using a Qt byte array. Return true if Comments metadata have been changed in memory.

**6.1079.4.108 setExif()**

```
bool Digikam::MetaEngine::setExif (
    const QByteArray & data ) const
```

Set the Exif data using a Qt byte array. Return true if Exif metadata have been changed in memory.

**6.1079.4.109 setExifComment()**

```
bool Digikam::MetaEngine::setExifComment (
    const QString & comment,
    bool writeDescription = true ) const
```

Set the Exif user comments from item. Look Exif specification for more details about this tag. Return true if Exif user comments have been changed in metadata.

**6.1079.4.110 setExifTagData()**

```
bool Digikam::MetaEngine::setExifTagData (
    const char * exifTagName,
    const QByteArray & data ) const
```

Set an Exif tag content using a bytes array. Return true if tag is set successfully.

**6.1079.4.111 setExifTagLong()**

```
bool Digikam::MetaEngine::setExifTagLong (
    const char * exifTagName,
    long val ) const
```

Set an Exif tag content using a long value. Return true if tag is set successfully.

**6.1079.4.112 setExifTagRational()**

```
bool Digikam::MetaEngine::setExifTagRational (
    const char * exifTagName,
    long int num,
    long int den ) const
```

Set an Exif tag content using a rational value. 'num' and 'den' are the numerator and the denominator of the rational value. Return true if tag is set successfully.

**6.1079.4.113 setExifTagString()**

```
bool Digikam::MetaEngine::setExifTagString (
    const char * exifTagName,
    const QString & value ) const
```

Set an Exif tag content using a string. Return true if tag is set successfully.

**6.1079.4.114 setExifTagURational()**

```
bool Digikam::MetaEngine::setExifTagURational (
    const char * exifTagName,
    unsigned long int num,
    unsigned long int den ) const
```

Set an Exif tag content using a unsigned rational value. 'num' and 'den' are the numerator and the denominator of the unsigned rational value. Return true if tag is set successfully.

**6.1079.4.115 setExifTagUShort()**

```
bool Digikam::MetaEngine::setExifTagUShort (
    const char * exifTagName,
    unsigned int val ) const
```

Set an Exif tag content using a unsigned short value. Return true if tag is set successfully.

**6.1079.4.116 setExifTagVariant()**

```
bool Digikam::MetaEngine::setExifTagVariant (
    const char * exifTagName,
    const QVariant & data,
    bool rationalWantSmallDenominator = true ) const
```

Set an Exif tag content using a QVariant. Returns true if tag is set successfully. All types described for the [getExifTagVariant\(\)](#) method are supported. Calling with a QVariant of type QByteArray is equivalent to calling `setExifTagData`. For the meaning of `rationalWantSmallDenominator`, see the documentation of the `convertToRational` methods. Setting a value with multiple components is currently not supported.

**6.1079.4.117 setExifThumbnail()**

```
bool Digikam::MetaEngine::setExifThumbnail (
    const QImage & thumb ) const
```

Set the Exif Thumbnail image. The thumbnail image must have the right dimensions before. Look Exif specification for details. Return true if thumbnail have been changed in metadata.

**6.1079.4.118 setFilePath()**

```
void Digikam::MetaEngine::setFilePath (
    const QString & path )
```

Set the file path of current item.

**6.1079.4.119 setGPSInfo() [1/3]**

```
bool Digikam::MetaEngine::setGPSInfo (
    const double *const altitude,
    const double latitude,
    const double longitude )
```

Set all GPS location information into item. Return true if all information have been changed in metadata. If you do not want altitude to be set, pass a null pointer.

**6.1079.4.120 setGPSInfo() [2/3]**

```
bool Digikam::MetaEngine::setGPSInfo (
    const double altitude,
    const double latitude,
    const double longitude )
```

Set all GPS location information into item. Return true if all information have been changed in metadata.

**6.1079.4.121 setGPSInfo() [3/3]**

```
bool Digikam::MetaEngine::setGPSInfo (
    const double altitude,
    const QString & latitude,
    const QString & longitude )
```

Set all GPS location information into item. Return true if all information have been changed in metadata.

**6.1079.4.122 setImageDateTime()**

```
bool Digikam::MetaEngine::setImageDateTime (
    const QDateTime & dateTime,
    bool setDateDigitized = false ) const
```

Set the Exif and Iptc time stamp. If 'setDateDigitized' parameter is true, the 'Digitalized' time stamp is set, else only 'Created' time stamp is set.

### 6.1079.4.123 setIptc()

```
bool Digikam::MetaEngine::setIptc (
    const QByteArray & data ) const
```

Set the Iptc data using a Qt byte array. Return true if Iptc metadata have been changed in memory.

### 6.1079.4.124 setIptcKeywords()

```
bool Digikam::MetaEngine::setIptcKeywords (
    const QStringList & oldKeywords,
    const QStringList & newKeywords ) const
```

Set Iptc keywords using a list of strings defined by 'newKeywords' parameter. Use 'getImageKeywords()' method to set 'oldKeywords' parameter with existing keywords from item. The method will compare all new keywords with all old keywords to prevent duplicate entries in item. Return true if keywords have been changed in metadata.

### 6.1079.4.125 setIptcSubCategories()

```
bool Digikam::MetaEngine::setIptcSubCategories (
    const QStringList & oldSubCategories,
    const QStringList & newSubCategories ) const
```

Set Iptc sub-categories using a list of strings defined by 'newSubCategories' parameter. Use 'getImageSubCategories()' method to set 'oldSubCategories' parameter with existing sub-categories from item. The method will compare all new sub-categories with all old sub-categories to prevent duplicate entries in item. Return true if sub-categories have been changed in metadata.

### 6.1079.4.126 setIptcSubjects()

```
bool Digikam::MetaEngine::setIptcSubjects (
    const QStringList & oldSubjects,
    const QStringList & newSubjects ) const
```

Set Iptc subjects using a list of strings defined by 'newSubjects' parameter. Use 'getImageSubjects()' method to set 'oldSubjects' parameter with existing subjects from item. The method will compare all new subjects with all old subjects to prevent duplicate entries in item. Return true if subjects have been changed in metadata.

### 6.1079.4.127 setIptcTagData()

```
bool Digikam::MetaEngine::setIptcTagData (
    const char * iptcTagName,
    const QByteArray & data ) const
```

Set an Iptc tag content using a bytes array. Return true if tag is set successfully.



**6.1079.4.128 setIptcTagsStringList()**

```
bool Digikam::MetaEngine::setIptcTagsStringList (
    const char * iptcTagName,
    int maxSize,
    const QStringList & oldValues,
    const QStringList & newValues ) const
```

Set multiple Iptc tags contents using a strings list. 'maxSize' is the max characters size of one entry. Return true if all tags have been set successfully.

**6.1079.4.129 setIptcTagString()**

```
bool Digikam::MetaEngine::setIptcTagString (
    const char * iptcTagName,
    const QString & value ) const
```

Set an Iptc tag content using a string. Return true if tag is set successfully.

**6.1079.4.130 setItemColorWorkSpace()**

```
bool Digikam::MetaEngine::setItemColorWorkSpace (
    ImageColorWorkSpace workspace ) const
```

Set the Exif color-space tag of item. See [ImageColorWorkSpace](#) values for details Return true if work-space have been changed in metadata.

**6.1079.4.131 setItemDimensions()**

```
bool Digikam::MetaEngine::setItemDimensions (
    const QSize & size ) const
```

Set the size of item in pixels in Exif tags. Return true if size have been changed in metadata.

**6.1079.4.132 setItemIccProfile()**

```
bool Digikam::MetaEngine::setItemIccProfile (
    const QByteArray & iccData ) const
```

Set image ICC profile.

**6.1079.4.133 setItemOrientation()**

```
bool Digikam::MetaEngine::setItemOrientation (
    ImageOrientation orientation ) const
```

Set the Exif orientation tag of item. See [ImageOrientation](#) values for details Return true if orientation have been changed in metadata.

**6.1079.4.134 setItemPreview()**

```
bool Digikam::MetaEngine::setItemPreview (
    const QImage & preview ) const
```

Set the lptc preview image. The thumbnail item must have the right size before (64Kb max with JPEG file, else 256Kb). Look lptc specification for details. Return true if preview have been changed in metadata. Re-implement this method if you want to use another item file format than JPEG to save preview.

**6.1079.4.135 setItemProgramId()**

```
bool Digikam::MetaEngine::setItemProgramId (
    const QString & program,
    const QString & version ) const
```

Set Program name and program version in Exif and lptc Metadata. Return true if information have been changed in metadata.

**6.1079.4.136 setMetadataWritingMode()**

```
void Digikam::MetaEngine::setMetadataWritingMode (
    const int mode )
```

Set metadata writing mode.

**Parameters**

<i>mode</i>	Metadata writing mode as defined by the <a href="#">MetadataWritingMode</a> enum.
-------------	---

**See also**

[MetadataWritingMode](#), [metadataWritingMode\(\)](#)

**6.1079.4.137 setProgramId()**

```
bool Digikam::MetaEngine::setProgramId ( ) const [protected]
```

Set the Program Name and Program Version information in Exif and lptc metadata

**6.1079.4.138 setReadWithExifTool()**

```
void Digikam::MetaEngine::setReadWithExifTool (
    const bool on )
```

Enable or disable reading metadata operations with ExifTool.

**6.1079.4.139 setTiffThumbnail()**

```
bool Digikam::MetaEngine::setTiffThumbnail (
    const QImage & thumb ) const
```

Adds a JPEG thumbnail to a TIFF images. Use this instead of setExifThumbnail for TIFF images.

**6.1079.4.140 setUpdateFileTimeStamp()**

```
void Digikam::MetaEngine::setUpdateFileTimeStamp (
    bool on )
```

Enable or disable file timestamp updating when metadata are saved. By default files timestamp are untouched.

**6.1079.4.141 setUseCompatibleFileName()**

```
void Digikam::MetaEngine::setUseCompatibleFileName (
    const bool on )
```

Enable or disable using compatible file name for sidecar files.

**6.1079.4.142 setUseXMPSidecar4Reading()**

```
void Digikam::MetaEngine::setUseXMPSidecar4Reading (
    const bool on )
```

Enable or disable using XMP sidecar for reading metadata.

**6.1079.4.143 setWriteDngFiles()**

```
void Digikam::MetaEngine::setWriteDngFiles (
    const bool on )
```

Enable or disable writing metadata operations to DNG files.

**6.1079.4.144 setWriteRawFiles()**

```
void Digikam::MetaEngine::setWriteRawFiles (
    const bool on )
```

Enable or disable writing metadata operations to RAW files. By default RAW files are untouched.

**6.1079.4.145 setWriteWithExifTool()**

```
void Digikam::MetaEngine::setWriteWithExifTool (
    const bool on )
```

Enable or disable writing metadata operations with ExifTool.

**6.1079.4.146 setXmp()**

```
bool Digikam::MetaEngine::setXmp (
    const QByteArray & data ) const
```

Set the Xmp data using a Qt byte array. Return true if Xmp metadata have been changed in memory.

**6.1079.4.147 setXmpKeywords()**

```
bool Digikam::MetaEngine::setXmpKeywords (
    const QStringList & newKeywords ) const
```

Set Xmp keywords using a list of strings defined by 'newKeywords' parameter. The existing keywords from item are preserved. The method will compare all new keywords with all already existing keywords to prevent duplicate entries in item. Return true if keywords have been changed in metadata.

**6.1079.4.148 setXmpSubCategories()**

```
bool Digikam::MetaEngine::setXmpSubCategories (
    const QStringList & newSubCategories ) const
```

Set Xmp sub-categories using a list of strings defined by 'newSubCategories' parameter. The existing sub-categories from item are preserved. The method will compare all new sub-categories with all already existing sub-categories to prevent duplicate entries in item. Return true if sub-categories have been changed in metadata.

**6.1079.4.149 setXmpSubjects()**

```
bool Digikam::MetaEngine::setXmpSubjects (
    const QStringList & newSubjects ) const
```

Set Xmp subjects using a list of strings defined by 'newSubjects' parameter. The existing subjects from item are preserved. The method will compare all new subject with all already existing subject to prevent duplicate entries in item. Return true if subjects have been changed in metadata.

**6.1079.4.150 setXmpTagString() [1/2]**

```
bool Digikam::MetaEngine::setXmpTagString (
    const char * xmpTagName,
    const QString & value ) const
```

Set a Xmp tag content using a string. Return true if tag is set successfully.

**6.1079.4.151 setXmpTagString() [2/2]**

```
bool Digikam::MetaEngine::setXmpTagString (
    const char * xmpTagName,
    const QString & value,
    MetaEngine::XmpTagType type ) const
```

Set a Xmp tag with a specific type. Return true if tag is set successfully. This method only accept NormalTag, ArrayBagTag and StructureTag. Other XmpTagTypes do nothing

**6.1079.4.152 setXmpTagStringBag()**

```
bool Digikam::MetaEngine::setXmpTagStringBag (
    const char * xmpTagName,
    const QStringList & bag ) const
```

Set a Xmp tag content using the bag of strings 'bag'. Return true if tag is set successfully.

**6.1079.4.153 setXmpTagStringLangAlt()**

```
bool Digikam::MetaEngine::setXmpTagStringLangAlt (
    const char * xmpTagName,
    const QString & value,
    const QString & langAlt ) const
```

Set a Xmp tag content using a string with an alternative language header. 'langAlt' contain the language alternative information (like "fr-FR" for French - RFC3066 notation) or is null to set alternative language to default settings ("x-default"). Return true if tag is set successfully.

**6.1079.4.154 setXmpTagStringListLangAlt()**

```
bool Digikam::MetaEngine::setXmpTagStringListLangAlt (
    const char * xmpTagName,
    const MetaEngine::AltLangMap & values ) const
```

Set an Alternative Language Xmp tag content using a map. See AltLangMap class description for details. If tag already exist, it will be removed before. Return true if tag is set successfully.

**6.1079.4.155 setXmpTagStringSeq()**

```
bool Digikam::MetaEngine::setXmpTagStringSeq (
    const char * xmpTagName,
    const QStringList & seq ) const
```

Set a Xmp tag content using the sequence of strings 'seq'. Return true if tag is set successfully.

**6.1079.4.156 sidecarFilePathForFile()**

```
QString Digikam::MetaEngine::sidecarFilePathForFile (
    const QString & path ) [static]
```

Return the XMP Sidecar file path for a item file path. If item file path do not include a file name or is empty, this function return a null string.

**6.1079.4.157 sidecarPath()**

```
QString Digikam::MetaEngine::sidecarPath (
    const QString & path ) [static]
```

Like [sidecarFilePathForFile\(\)](#), but works for local file path.

**6.1079.4.158 sidecarUrl()** [1/2]

```
QUrl Digikam::MetaEngine::sidecarUrl (
    const QString & path ) [static]
```

Gives a file url for a local path.

**6.1079.4.159 sidecarUrl()** [2/2]

```
QUrl Digikam::MetaEngine::sidecarUrl (
    const QUrl & url ) [static]
```

Like [sidecarFilePathForFile\(\)](#), but works for remote URLs.

**6.1079.4.160 supportBmff()**

```
bool Digikam::MetaEngine::supportBmff ( ) [static]
```

Return true if library support Base Media File Format (aka CR3, HEIF, HEIC, and AVIF). Note: use this function only after to call [initializeExiv2\(\)](#), else false will always returned. The function return true only if Exiv2  $\geq$  0.27.4 compiled with BMFF support.

**6.1079.4.161 supportJpegXL()**

```
bool Digikam::MetaEngine::supportJpegXL ( ) [static]
```

Return true if Exiv2 library is compiled with JpegXL metadata support.

**6.1079.4.162 supportMetadataWriting()**

```
bool Digikam::MetaEngine::supportMetadataWriting (
    const QString & typeMime ) [static]
```

Return true if library can write metadata to typeMime file format.

**6.1079.4.163 supportXmp()**

```
bool Digikam::MetaEngine::supportXmp ( ) [static]
```

Return true if Exiv2 library is compiled with Xmp metadata support.

**6.1079.4.164 unregisterXmpNameSpace()**

```
bool Digikam::MetaEngine::unregisterXmpNameSpace (
    const QString & uri ) [static]
```

Unregister a previously registered custom namespace

**6.1079.4.165 updateFileTimeStamp()**

```
bool Digikam::MetaEngine::updateFileTimeStamp ( ) const
```

Return true if file timestamp is updated when metadata are saved.

**6.1079.4.166 useCompatibleFileName()**

```
bool Digikam::MetaEngine::useCompatibleFileName ( ) const
```

Return true if using compatible file name for sidecar files.

**6.1079.4.167 useXMPSidecar4Reading()**

```
bool Digikam::MetaEngine::useXMPSidecar4Reading ( ) const
```

Return true if using XMP sidecar for reading metadata is enabled.

**6.1079.4.168 writeDngFiles()**

```
bool Digikam::MetaEngine::writeDngFiles ( ) const
```

Return true if writing metadata operations on DNG files is enabled.

**6.1079.4.169 writeRawFiles()**

```
bool Digikam::MetaEngine::writeRawFiles ( ) const
```

Return true if writing metadata operations on RAW files is enabled.

**6.1079.4.170 writeWithExifTool()**

```
bool Digikam::MetaEngine::writeWithExifTool ( ) const
```

Return true if writing metadata operations with ExifTool is enabled.

## 6.1080 Digikam::MetaEngine::Private Class Reference

### Public Member Functions

- QString [convertCommentValue](#) (const Exiv2::Exifdatum &exifDatum) const
- void **copyPrivateData** (const [Private](#) \*const other)
- bool [decodeGPSCoordinate](#) (const char \*exifTagName, double \*const coordinate) const
- QString [detectEncodingAndDecode](#) (const std::string &value) const
- Exiv2::ByteOrder & **exifByteOrder** ()
- const Exiv2::ByteOrder & **exifByteOrder** () const
- Exiv2::ExifData & **exifMetadata** ()
- const Exiv2::ExifData & **exifMetadata** () const
- bool **exportChanges** (const QString &exvTmpFile) const
- QString [extractIptcTagString](#) (const Exiv2::IptcData &iptcData, const Exiv2::Iptcdatum &iptcTag) const
- int **getXMPTagsListFromPrefix** (const QString &pf, [MetaEngine::TagsMap](#) &tagsMap) const
- Exiv2::DataBuf & **iccProfileBuf** ()
- const Exiv2::DataBuf & **iccProfileBuf** () const
- Exiv2::IptcData & **iptcMetadata** ()
- const Exiv2::IptcData & **iptcMetadata** () const
- bool **isUtf8** (const char \*const buffer) const
- std::string & **itemComments** ()
- const std::string & **itemComments** () const
- **Private** ([MetaEngine](#) \*const q)
- bool **saveToFile** (const QFileInfo &finfo) const
- bool **saveToXMPSidecar** (const QFileInfo &finfo) const
- bool **saveUsingExifTool** (const QFileInfo &finfo, const QDateTime &modTime) const
- bool **saveUsingExiv2** (const QFileInfo &finfo, const QDateTime &modTime, Exiv2::Image::AutoPtr image) const

### Static Public Member Functions

- static void [printExiv2ExceptionError](#) (const QString &msg, Exiv2::AnyError &e)
- static void [printExiv2MessageHandler](#) (int lvl, const char \*msg)

### Public Attributes

- QExplicitlySharedDataPointer< [MetaEngineData::Private](#) > **data**
- QString **filePath**
- bool **loadedFromSidecar** = false
  - XMP, and parts of EXIF/IPTC, were loaded from an XMP sidecar file.*
- int **metadataWritingMode** = [WRITE\\_TO\\_FILE\\_ONLY](#)
  - A mode from [MetadataWritingMode](#) enum.*
- QString  **mimeType**
- [MetaEngine](#) \* **parent** = nullptr
- QSize **pixelSize**
- bool **readWithExifTool** = false
- bool **updateFileTimeStamp** = false
- bool **useCompatibleFileName** = false
- bool **useXMPSidecar4Reading** = false
- bool **writeDngFiles** = false
- bool **writeRawFiles** = false
- bool **writeWithExifTool** = false



## 6.1080.1 Member Function Documentation

### 6.1080.1.1 convertCommentValue()

```
QString Digikam::MetaEngine::Private::convertCommentValue (
    const Exiv2::Exifdatum & exifDatum ) const
```

Wrapper method to convert a Comments content to a QString.

### 6.1080.1.2 decodeGPSCoordinate()

```
bool Digikam::MetaEngine::Private::decodeGPSCoordinate (
    const char * exifTagName,
    double *const coordinate ) const
```

Decodes Latitude or Longitude from EXIF tag name.

### 6.1080.1.3 detectEncodingAndDecode()

```
QString Digikam::MetaEngine::Private::detectEncodingAndDecode (
    const std::string & value ) const
```

Charset autodetection to convert a string to a QString.

### 6.1080.1.4 extractIptcTagString()

```
QString Digikam::MetaEngine::Private::extractIptcTagString (
    const Exiv2::IptcData & iptcData,
    const Exiv2::Iptcdatum & iptcTag ) const
```

Helper method to decode IPTC tag string contents following characters encoding preset.

### 6.1080.1.5 isUtf8()

```
bool Digikam::MetaEngine::Private::isUtf8 (
    const char *const buffer ) const
```

UTF8 autodetection from a string.

### 6.1080.1.6 printExiv2ExceptionError()

```
void Digikam::MetaEngine::Private::printExiv2ExceptionError (
    const QString & msg,
    Exiv2::AnyError & e ) [static]
```

Generic method to print the Exiv2 C++ Exception error message from 'e'. 'msg' string is printed using qDebug rules.

### 6.1080.1.7 printExiv2MessageHandler()

```
void Digikam::MetaEngine::Private::printExiv2MessageHandler (
    int lvl,
    const char * msg ) [static]
```

Generic method to print debug message from Exiv2. 'msg' string is printed using qDebug rules. 'lvl' is the debug level of Exiv2 message.

## 6.1081 Digikam::MetaEngineData Class Reference

### Classes

- class [Private](#)

### Public Member Functions

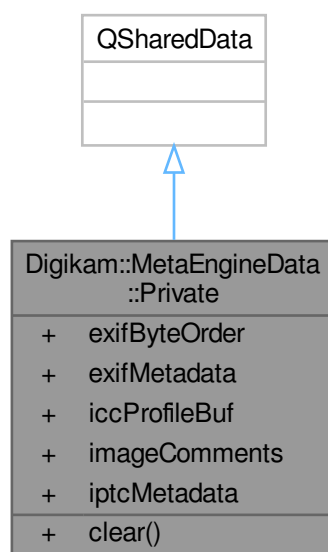
- **MetaEngineData** (const [MetaEngineData](#) &)
- [MetaEngineData](#) & **operator=** (const [MetaEngineData](#) &)

### Friends

- class **MetaEngine**

## 6.1082 Digikam::MetaEngineData::Private Class Reference

Inheritance diagram for Digikam::MetaEngineData::Private:



### Public Member Functions

- void **clear** ()

### Public Attributes

- Exiv2::ByteOrder **exifByteOrder** = Exiv2::invalidByteOrder
- Exiv2::ExifData **exifMetadata**
- Exiv2::DataBuf **iccProfileBuf**
- std::string **imageComments**
- Exiv2::IptcData **iptcMetadata**

## 6.1083 Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList > Class Template Reference

### Public Member Functions

- void [exclusiveMerge](#) (const Data &src, Data &dest)
- void [mergeAll](#) (const Data &src, Data &dest)
- void [mergeFields](#) (const Data &src, Data &dest)
- [MetaEngineMergeHelper](#) & **operator**<< (const KeyString &key)

### Public Attributes

- KeyStringList **keys**

### 6.1083.1 Member Function Documentation

#### 6.1083.1.1 exclusiveMerge()

```
template<class Data , class Key , class KeyString , class KeyStringList = QList<KeyString>>
void Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >::exclusiveMerge (
    const Data & src,
    Data & dest ) [inline]
```

Merge two (Exif,IPTC,Xmp) Data packages, the result is stored in dest. The following steps apply only to keys in "keys": The result is determined by src. Keys must exist in src to kept in dest. Fields from src take precedence over existing data from dest.

#### 6.1083.1.2 mergeAll()

```
template<class Data , class Key , class KeyString , class KeyStringList = QList<KeyString>>
void Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >::mergeAll (
    const Data & src,
    Data & dest ) [inline]
```

Merge two (Exif,IPTC,Xmp) Data packages, where the result is stored in dest and fields from src take precedence over existing data from dest.

### 6.1083.1.3 mergeFields()

```
template<class Data , class Key , class KeyString , class KeyStringList = QList<KeyString>>
void Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >::mergeFields (
    const Data & src,
    Data & dest ) [inline]
```

Merge two (Exif,IPTC,Xmp) Data packages, the result is stored in dest. Only keys in keys are considered for merging. Fields from src take precedence over existing data from dest.

## 6.1084 Digikam::MetaEnginePreviews Class Reference

### Public Member Functions

- int **count** () const  
*Returns how many embedded previews are available.*
- QByteArray **data** (int index=0)
- int **dataSize** (int index=0)
- QString **fileExtension** (int index=0)
- int **height** (int index=0)
- QImage **image** (int index=0)
- bool **isEmpty** ()  
*Returns if there are any preview images available.*
- [MetaEnginePreviews](#) (const QByteArray &imgData)
- [MetaEnginePreviews](#) (const QString &filePath)
- QString **mimeType** (int index=0)
- QString **originalMimeType** () const  
*Returns the mimeType of the original image, detected from the file's content.*
- QSize **originalSize** () const  
*Returns the pixel size of the original image, as read from the file (not the metadata).*
- int **size** () const
- int **width** (int index=0)

### 6.1084.1 Constructor & Destructor Documentation

#### 6.1084.1.1 MetaEnginePreviews() [1/2]

```
Digikam::MetaEnginePreviews::MetaEnginePreviews (
    const QString & filePath ) [explicit]
```

Open the given file and scan for embedded preview images

#### 6.1084.1.2 MetaEnginePreviews() [2/2]

```
Digikam::MetaEnginePreviews::MetaEnginePreviews (
    const QByteArray & imgData ) [explicit]
```

Open the given image data and scan the image for embedded preview images.

## 6.1084.2 Member Function Documentation

### 6.1084.2.1 data()

```
QByteArray Digikam::MetaEnginePreviews::data (
    int index = 0 )
```

Retrieve the image data for the specified embedded preview image

### 6.1084.2.2 dataSize()

```
int Digikam::MetaEnginePreviews::dataSize (
    int index = 0 )
```

For each contained preview image, return the size of the image data in bytes, width and height of the preview, the mimeType and the file extension. Ensure that `index < count()`. Previews are sorted by width\*height, largest first.

### 6.1084.2.3 image()

```
QImage Digikam::MetaEnginePreviews::image (
    int index = 0 )
```

Loads the data of the specified preview and creates a QImage from this data. Returns a null QImage if the loading failed.

## 6.1085 Digikam::MetaEngineRotation Class Reference

### Public Types

- enum [TransformationAction](#) {  
[NoTransformation](#) = 0 , [FlipHorizontal](#) = 1 , [FlipVertical](#) = 2 , [Rotate90](#) = 5 ,  
[Rotate180](#) = 6 , [Rotate270](#) = 7 }

### Public Member Functions

- [MetaEngine::ImageOrientation](#) [exifOrientation](#) () const
- bool [isNoTransform](#) () const
- [MetaEngineRotation](#) ()
- [MetaEngineRotation](#) (int m11, int m12, int m21, int m22)
- [MetaEngineRotation](#) ([MetaEngine::ImageOrientation](#) [exifOrientation](#))
- [MetaEngineRotation](#) ([TransformationAction](#) action)
- bool **operator!=** (const [MetaEngineRotation](#) &ma) const
- [MetaEngineRotation](#) & **operator\*=** (const [MetaEngineRotation](#) &ma)
- [MetaEngineRotation](#) & **operator\*=** (const QList< [TransformationAction](#) > &actions)
- [MetaEngineRotation](#) & **operator\*=** ([MetaEngine::ImageOrientation](#) [exifOrientation](#))
- [MetaEngineRotation](#) & **operator\*=** ([TransformationAction](#) action)
- bool **operator==** (const [MetaEngineRotation](#) &ma) const
- QTransform [toTransform](#) () const
- QList< [TransformationAction](#) > [transformations](#) () const

### Static Public Member Functions

- static QTransform [toTransform](#) ([MetaEngine::ImageOrientation](#) orientation)

### Protected Member Functions

- void **set** (int m11, int m12, int m21, int m22)

### Protected Attributes

- int **m** [2][2]

## 6.1085.1 Member Enumeration Documentation

### 6.1085.1.1 TransformationAction

```
enum Digikam::MetaEngineRotation::TransformationAction
```

This describes single transform primitives. Note some of the defined Exif rotation flags combine two of these actions. The enum values correspond to those defined as JXFORM\_CODE in the often used the JPEG tool transupp.h.

#### Enumerator

NoTransformation	no transformation
FlipHorizontal	horizontal flip
FlipVertical	vertical flip
Rotate90	90-degree clockwise rotation
Rotate180	180-degree rotation
Rotate270	270-degree clockwise (or 90 ccw)

## 6.1085.2 Constructor & Destructor Documentation

### 6.1085.2.1 MetaEngineRotation() [1/3]

```
Digikam::MetaEngineRotation::MetaEngineRotation ( )
```

Constructs the identity matrix (the matrix describing no transformation)

### 6.1085.2.2 MetaEngineRotation() [2/3]

```
Digikam::MetaEngineRotation::MetaEngineRotation (
    TransformationAction action ) [explicit]
```

Returns the matrix corresponding to the given TransformationAction

**6.1085.2.3 MetaEngineRotation()** [3/3]

```
Digikam::MetaEngineRotation::MetaEngineRotation (
    MetaEngine::ImageOrientation exifOrientation ) [explicit]
```

Returns the matrix corresponding to the given TransformationAction

**6.1085.3 Member Function Documentation****6.1085.3.1 exifOrientation()**

```
MetaEngine::ImageOrientation Digikam::MetaEngineRotation::exifOrientation ( ) const
```

Returns the Exif orientation flag describing this matrix. Returns ORIENTATION\_UNSPECIFIED if no flag matches this matrix.

**6.1085.3.2 isNoTransform()**

```
bool Digikam::MetaEngineRotation::isNoTransform ( ) const
```

Returns true if this matrix describes no transformation (is the identity matrix)

**6.1085.3.3 operator\*=( )** [1/3]

```
MetaEngineRotation & Digikam::MetaEngineRotation::operator*= (
    const QList< TransformationAction > & actions )
```

Applies the given transform actions to this matrix

**6.1085.3.4 operator\*=( )** [2/3]

```
MetaEngineRotation & Digikam::MetaEngineRotation::operator*= (
    MetaEngine::ImageOrientation exifOrientation )
```

Applies the given Exif orientation flag to this matrix

**6.1085.3.5 operator\*=( )** [3/3]

```
MetaEngineRotation & Digikam::MetaEngineRotation::operator*= (
    TransformationAction action )
```

Applies the given transform to this matrix

**6.1085.3.6 toTransform()** [1/2]

```
QTransform Digikam::MetaEngineRotation::toTransform ( ) const
```

Returns a QTransform representing this matrix

**6.1085.3.7 toTransform()** [2/2]

```
QTransform Digikam::MetaEngineRotation::toTransform (
    MetaEngine::ImageOrientation orientation ) [static]
```

Returns a QTransform for the given Exif orientation

**6.1085.3.8 transformations()**

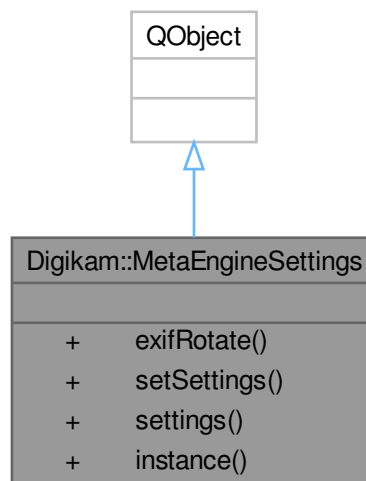
```
QList< MetaEngineRotation::TransformationAction > Digikam::MetaEngineRotation::transformations
( ) const
```

Returns the actions described by this matrix. The order matters. Not all possible matrices are supported, but all those that can be combined by Exif rotation flags and the transform actions above. If `isNoTransform()` or the matrix is not supported returns an empty list.

Converts the mathematically correct description into the primitive operations that can be carried out losslessly.

**6.1086 Digikam::MetaEngineSettings Class Reference**

Inheritance diagram for Digikam::MetaEngineSettings:

**Signals**

- void **signalMetaEngineSettingsChanged** (const [MetaEngineSettingsContainer](#) &current, const [MetaEngineSettingsContainer](#) &previous)
- void **signalSettingsChanged** ()



## Public Member Functions

- bool [exifRotate](#) () const
- void [setSettings](#) (const [MetaEngineSettingsContainer](#) &settings)
- [MetaEngineSettingsContainer](#) settings () const

## Static Public Member Functions

- static [MetaEngineSettings](#) \* [instance](#) ()

## Friends

- class [MetaEngineSettingsCreator](#)

## 6.1086.1 Member Function Documentation

### 6.1086.1.1 [exifRotate\(\)](#)

```
bool Digikam::MetaEngineSettings::exifRotate ( ) const
```

Shortcut to get exif rotation settings from container.

### 6.1086.1.2 [instance\(\)](#)

```
MetaEngineSettings * Digikam::MetaEngineSettings::instance ( ) [static]
```

Global container for Metadata settings. All accessor methods are thread-safe.

### 6.1086.1.3 [setSettings\(\)](#)

```
void Digikam::MetaEngineSettings::setSettings (
    const MetaEngineSettingsContainer & settings )
```

Sets the current Metadata settings and writes them to config.

### 6.1086.1.4 [settings\(\)](#)

```
MetaEngineSettingsContainer Digikam::MetaEngineSettings::settings ( ) const
```

Returns the current Metadata settings.

## 6.1087 Digikam::MetaEngineSettingsContainer Class Reference

The class [MetaEngineSettingsContainer](#) encapsulates all metadata related settings.

## Public Types

- enum **AlbumDateSource** {  
**NewestItemDate** = 0 , **OldestItemDate** , **AverageDate** , **FolderDate** ,  
**IgnoreDate** }
- enum **RotationBehaviorFlag** {  
**NoRotation** = 0 , **RotateByInternalFlag** = 1 << 0 , **RotateByMetadataFlag** = 1 << 1 , **RotateByLosslessRotation** = 1 << 2 ,  
**RotateByLossyRotation** = 1 << 3 , **RotatingFlags** = RotateByInternalFlag | RotateByMetadataFlag ,  
**RotatingPixels** = RotateByLosslessRotation | RotateByLossyRotation }

## Public Member Functions

- QStringList **defaultExifToolSearchPaths** () const
- void **readFromConfig** (const KConfigGroup &group)
- void **writeToConfig** (KConfigGroup &group) const

## Public Attributes

- AlbumDateSource **albumDateFrom** = OldestItemDate
- bool **exifRotate** = true
- bool **exifSetOrientation** = true
- QString **exifToolPath**
- [MetaEngine::MetadataWritingMode](#) **metadataWritingMode** = [MetaEngine::WRITE\\_TO\\_FILE\\_ONLY](#)
- bool **readWithExifTool** = false
- bool **rescanImageIfModified** = false
- RotationBehaviorFlags **rotationBehavior** = RotationBehaviorFlags(RotatingFlags | RotateByLosslessRotation)
- bool **saveColorLabel** = false
- bool **saveComments** = false
- bool **saveDateTime** = false
- bool **saveFaceTags** = false
- bool **savePickLabel** = false
- bool **savePosition** = false
- bool **saveRating** = false
- bool **saveTags** = false
- bool **saveTemplate** = false
- QStringList **sidecarExtensions**
- bool **updateFileTimeStamp** = true
- bool **useCompatibleFileName** = false
- bool **useFastScan** = false
- bool **useLazySync** = false
- bool **useXMPSidecar4Reading** = false
- bool **writeDngFiles** = false
- bool **writeRawFiles** = false
- bool **writeWithExifTool** = false

## 6.1087.1 Detailed Description

### Note

this allows supply changed arguments to [MetadataHub](#) without changing the global settings.

## 6.1087.2 Member Enumeration Documentation

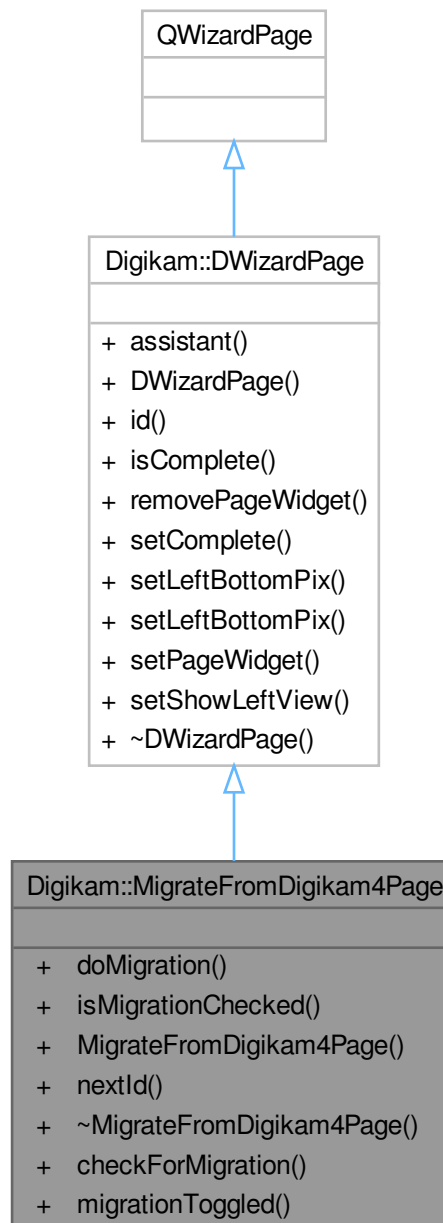
### 6.1087.2.1 RotationBehaviorFlag

enum `Digikam::MetaEngineSettingsContainer::RotationBehaviorFlag`

Describes the allowed and desired operation when rotating a picture. The modes are in escalating order and describe if an operation is allowed. What is actually done will be governed by what is possible: 1) RAW files cannot be rotated by content, setting the metadata may be problematic 2) Read-Only files cannot be edited, neither content nor metadata 3) Writable files will have lossy compression 4) Only JPEG and PGF offer lossless rotation Using a contents-based rotation always implies resetting the flag.

## 6.1088 Digikam::MigrateFromDigikam4Page Class Reference

Inheritance diagram for Digikam::MigrateFromDigikam4Page:



### Public Slots

- void **migrationToggled** (bool b)

## Public Member Functions

- void **doMigration** ()
- bool **isMigrationChecked** () const
- **MigrateFromDigikam4Page** (QWizard \*const dlg)
- int **nextId** () const override

## Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)
- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## Static Public Member Functions

- static bool **checkForMigration** ()

## 6.1088.1 Member Function Documentation

### 6.1088.1.1 **checkForMigration()**

```
bool Digikam::MigrateFromDigikam4Page::checkForMigration ( ) [static]
```

Return true if migration data are available on the system

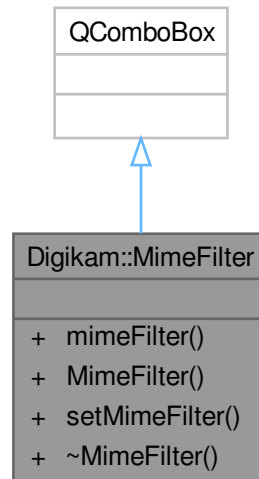
### 6.1088.1.2 **isMigrationChecked()**

```
bool Digikam::MigrateFromDigikam4Page::isMigrationChecked ( ) const
```

Returns true if the user selected to do a migration

## 6.1089 Digikam::MimeFilter Class Reference

Inheritance diagram for Digikam::MimeFilter:



### Public Types

- enum `TypeMimeFilter` {  
**AllFiles** = 0 , **ImageFiles** , **NoRAWFiles** , **JPGFiles** ,  
**JPEG2000Files** , **JPEGXLFiles** , **WEBPFiles** , **PNGFiles** ,  
**TIFFFiles** , **PGFFiles** , **HEIFFiles** , **AVIFiles** ,  
**DNGFiles** , **RAWFiles** , **MoviesFiles** , **AudioFiles** ,  
**RasterGraphics** }

### Public Member Functions

- int **mimeFilter** ()
- **MimeFilter** (QWidget \*const parent)
- void **setMimeFilter** (int filter)

## 6.1089.1 Member Enumeration Documentation

### 6.1089.1.1 TypeMimeFilter

```
enum Digikam::MimeFilter::TypeMimeFilter
```

#### Enumerator

HEIFFiles	HEVC H265 compression based containers.
RAWFiles	All Raw file formats such as nef, cr2, arw, pef, etc..
RasterGraphics	PSD, XCF, etc...

## 6.1090 Digikam::MixerContainer Class Reference

### Public Attributes

- double **blackBlueGain** = 0.0
- double **blackGreenGain** = 0.0
- double **blackRedGain** = 1.0
- double **blueBlueGain** = 1.0
- double **blueGreenGain** = 0.0
- double **blueRedGain** = 0.0
- bool **bMonochrome** = false
- bool **bPreserveLum** = true
- double **greenBlueGain** = 0.0
- double **greenGreenGain** = 1.0
- double **greenRedGain** = 0.0
- double **redBlueGain** = 0.0
- double **redGreenGain** = 0.0
- double **redRedGain** = 1.0

## 6.1091 Digikam::MixerFilter Class Reference

Inheritance diagram for Digikam::MixerFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override



- **MixerFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [MixerContainer](#) &settings=[MixerContainer](#)())
- **MixerFilter** (QObject \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.1091.1 Member Function Documentation

### 6.1091.1.1 filterAction()

`FilterAction` Digikam::MixerFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1091.1.2 filterIdentifier()

`QString` Digikam::MixerFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

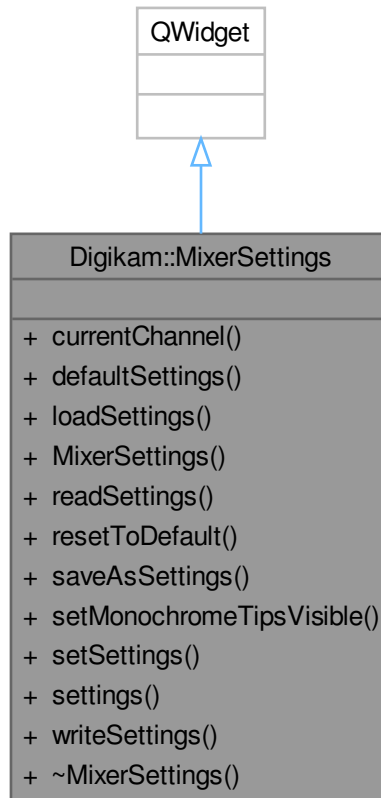
### 6.1091.1.3 readParameters()

```
void Digikam::MixerFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1092 Digikam::MixerSettings Class Reference

Inheritance diagram for Digikam::MixerSettings:



### Signals

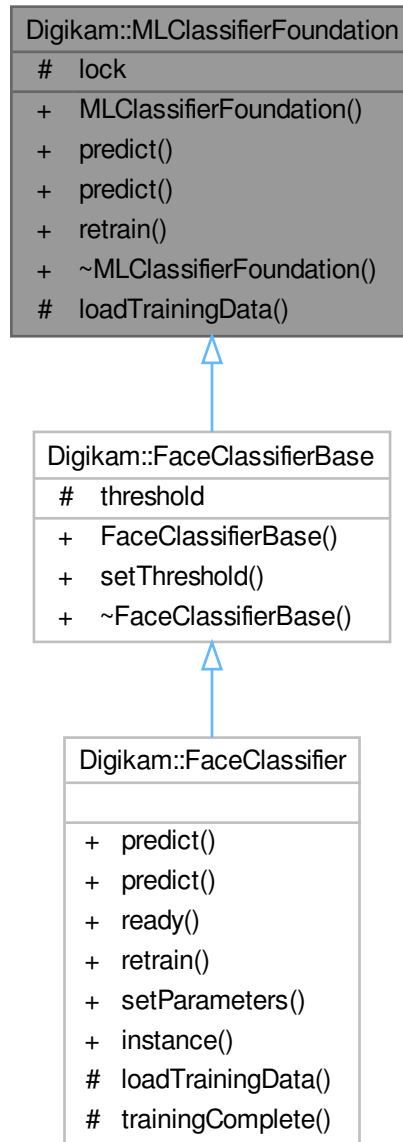
- void **signalMonochromeActivated** (bool)
- void **signalOutChannelChanged** ()
- void **signalSettingsChanged** ()

### Public Member Functions

- int **currentChannel** () const
- [MixerContainer](#) **defaultSettings** () const
- void **loadSettings** ()
- **MixerSettings** (QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **saveAsSettings** ()
- void **setMonochromeTipsVisible** (bool b)
- void **setSettings** (const [MixerContainer](#) &settings)
- [MixerContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.1093 Digikam::MLClassifierFoundation Class Reference

Inheritance diagram for Digikam::MLClassifierFoundation:



### Classes

- class [VotingGroups](#)

### Public Member Functions

- virtual int **predict** (const cv::Mat &target) const =0
- virtual int **predict** (const cv::UMat &target) const =0
- virtual bool **retrain** ()=0

### Protected Member Functions

- virtual bool **loadTrainingData** ()=0

### Protected Attributes

- QReadWriteLock **lock**

## 6.1094 Digikam::MLClassifierFoundation::VotingGroups Class Reference

### Classes

- struct [VoteTally](#)

### Public Types

- enum **WinnerType** { **VotesLowScore** , **VotesHighScore** , **LowScore** , **HighScore** }

### Public Member Functions

- void **addVote** (int label, float score)
- int **winner** (WinnerType winnerType)

## 6.1095 Digikam::MLClassifierFoundation::VotingGroups::VoteTally Struct Reference

### Public Attributes

- int **label** = 0
- float **score** = 0.0F
- int **votes** = 0

## 6.1096 Digikam::MLPipelineFoundation Class Reference

Inheritance diagram for Digikam::MLPipelineFoundation:



### Classes

- struct [\\_MLPipelinePerformanceProfile](#)

### Public Types

- enum [MLPipelineNotification](#) { [notifySkipped](#) , [notifyProcessed](#) }
- typedef struct [Digikam::MLPipelineFoundation::\\_MLPipelinePerformanceProfile](#) [MLPipelinePerformanceProfile](#)
- typedef [SharedQueue< MLPipelinePackageFoundation \\* >](#) [MLPipelineQueue](#)
- enum [MLPipelineStage](#) { [Finder](#) , [Loader](#) , [Extractor](#) , [Classifier](#) , [Trainer](#) , [Writer](#) , [None](#) }

## Signals

- void **finished** ()  
*Emitted when the last package has finished processing.*
- void **processed** (const MLPipelinePackageNotify::Ptr &package)  
*Emitted when one package has finished processing.*
- void **processing** (const MLPipelinePackageNotify::Ptr &package)  
*Emitted when one package begins processing.*
- void **progressValueChanged** (float progress)
- void **scheduled** ()  
*Emitted when processing is scheduled.*
- void **signalAddMoreWorkers** ()
- void **signalUpdateItemCount** (const qlonglong itemCount)
- void **skipped** (const MLPipelinePackageNotify::Ptr &package)  
*Emitted when one or several packages were skipped, usually because they have already been scanned.*
- void **started** (const QString &message)  
*Emitted when processing has started.*

## Public Member Functions

- virtual void **cancel** ()
- bool **hasFinished** () const
- virtual bool **start** ()

## Protected Member Functions

- virtual void **addMoreWorkers** ()=0
- bool **addWorker** (const MLPipelineStage &stage)
- bool **checkMoreWorkers** (int totalItemCount, int currentItemCount, bool useFullCpu)
- virtual bool **classifier** ()=0
- void **clearAllQueues** ()
- void **clearQueue** (MLPipelineQueue \*thisQueue)
- virtual MLPipelinePackageFoundation \* **dequeue** (MLPipelineQueue \*thisQueue)
- virtual bool **enqueue** (MLPipelineQueue \*thisQueue, MLPipelinePackageFoundation \*package)
- virtual bool **extractor** ()=0
- virtual bool **finder** ()=0
- virtual bool **loader** ()=0
- void **notify** (MLPipelineNotification notification, const QString &\_name, const QString &\_path, int \_processed, const QImage &\_thumbnail)
- void **notify** (MLPipelineNotification notification, const QString &\_name, const QString &\_path, int \_processed, const QIcon &\_thumbnail)
- void **notify** (MLPipelineNotification notification, const QString &\_name, const QString &\_path, int \_processed, const QImage &\_thumbnail)
- void **pipelinePerformanceEnd** (const MLPipelineStage &stage, int totalItemCount, QElapsedTimer &timer)
- void **pipelinePerformanceEnd** (const MLPipelineStage &stage, QElapsedTimer &timer)
- void **pipelinePerformanceStart** (const MLPipelineStage &stage, QElapsedTimer &timer)
- MLPipelinePackageFoundation \* **queueEndSignal** () const
- void **showPipelinePerformance** () const
- void **stageEnd** (MLPipelineStage thisStage, MLPipelineStage nextStage)
- void **stageStart** (QThread::Priority threadPriority, MLPipelineStage thisStage, MLPipelineStage nextStage, MLPipelineQueue \*&thisQueue, MLPipelineQueue \*&nextQueue)
- virtual bool **trainer** ()=0
- void **waitForStart** ()
- virtual bool **writer** ()=0



## Protected Attributes

- bool **cancelled** = false
- QAtomicInteger< int > **itemsProcessed** = 0
- quint64 **maxBufferSize** = 2147483648  
*2 GB default*
- QMutex **mutex**
- QMap< MLPipelineStage, MLPipelinePerformanceProfile > **performanceProfileList**
- QMap< MLPipelineStage, MLPipelineQueue \* > **queues**
- QThreadPool \* **threadPool** = nullptr
- QMutex **threadStageMutex**
- QAtomicInteger< int > **totalItemCount** = 0
- quint64 **usedBufferSize** = 0
- QList< QFutureWatcher< bool > \* > **watchList**

## 6.1096.1 Member Enumeration Documentation

### 6.1096.1.1 MLPipelineStage

enum [Digikam::MLPipelineFoundation::MLPipelineStage](#)

#### Enumerator

Finder	Finder stage finds the data for the pipeline.
Loader	Loader stage loads and prepares the data for extraction.
Extractor	Extractor stage pulls the features from the data.
Classifier	Classifier stage adds a label (face, autotag, etc) to an extracted object.
Trainer	Classifier stage adds a label (face, autotag, etc) to an extracted object.
Writer	Writer stage saves the data to the DB.
None	Empty stage.

## 6.1096.2 Member Function Documentation

### 6.1096.2.1 cancel()

```
void Digikam::MLPipelineFoundation::cancel ( ) [virtual]
```

worker threads can be in 1 of 3 states when cancel is called

1. waiting for a new package
2. processing a package
3. waiting to push a package

handle all 3 cases so the worker thread sees the cancel signal

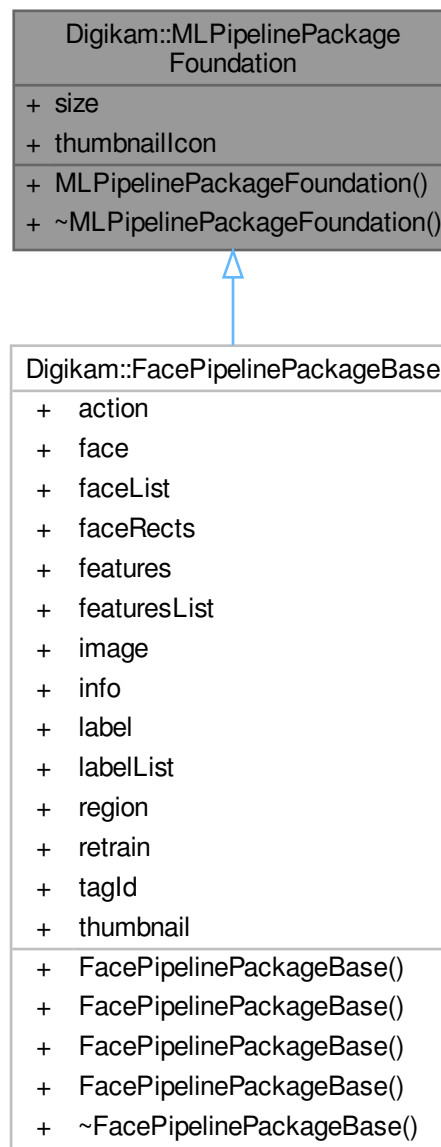
## 6.1097 Digikam::MLPipelineFoundation::\_MLPipelinePerformanceProfile Struct Reference

### Public Attributes

- QAtomicInteger< int > **currentThreadCount**
- int **elapsedTime** = 0
- int **itemCount** = 0
- int **maxElapsedTime** = 0
- int **maxQueueCount** = 0
- QAtomicInteger< int > **maxThreadCount**

## 6.1098 Digikam::MLPipelinePackageFoundation Class Reference

Inheritance diagram for Digikam::MLPipelinePackageFoundation:

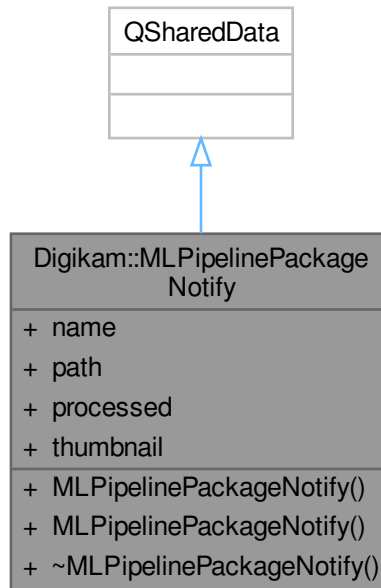


### Public Attributes

- quint64 **size** = 0
- QIcon **thumbnailIcon**

## 6.1099 Digikam::MLPipelinePackageNotify Class Reference

Inheritance diagram for Digikam::MLPipelinePackageNotify:



### Public Types

- typedef `QExplicitlySharedDataPointer< MLPipelinePackageNotify >` **Ptr**

### Public Member Functions

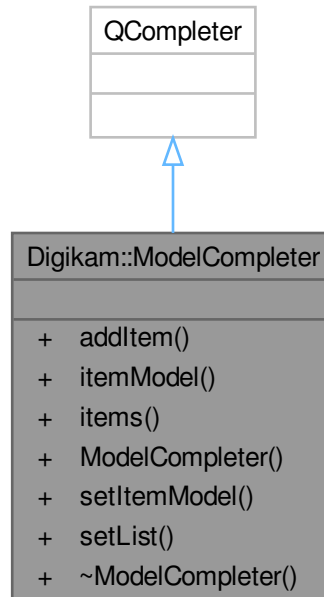
- **MLPipelinePackageNotify** (const `QString` &\_name, const `QString` &\_path, int \_processed, const `DImg` &↔\_thumbnail)
- **MLPipelinePackageNotify** (const `QString` &\_name, const `QString` &\_path, int \_processed, const `QIcon` &↔\_thumbnail)

### Public Attributes

- const `QString` **name**
- const `QString` **path**
- int **processed** = 0
- `QIcon` **thumbnail**

## 6.1100 Digikam::ModelCompleter Class Reference

Inheritance diagram for Digikam::ModelCompleter:



### Signals

- void **signalActivated** ()
- void **signalHighlighted** (int albumId)

### Public Member Functions

- void **addItem** (const QString &item)
- QAbstractItemModel \* **itemModel** () const
- QStringList **items** () const
- **ModelCompleter** (QObject \*const parent=nullptr)
- void **setItemModel** (QAbstractItemModel \*const model, int uniqueIdRole, int displayRole=Qt::DisplayRole)
- void **setList** (const QStringList &list)

### 6.1100.1 Member Function Documentation

#### 6.1100.1.1 setItemModel()

```

void Digikam::ModelCompleter::setItemModel (
    QAbstractItemModel *const model,
    int uniqueIdRole,
    int displayRole = Qt::DisplayRole )
  
```

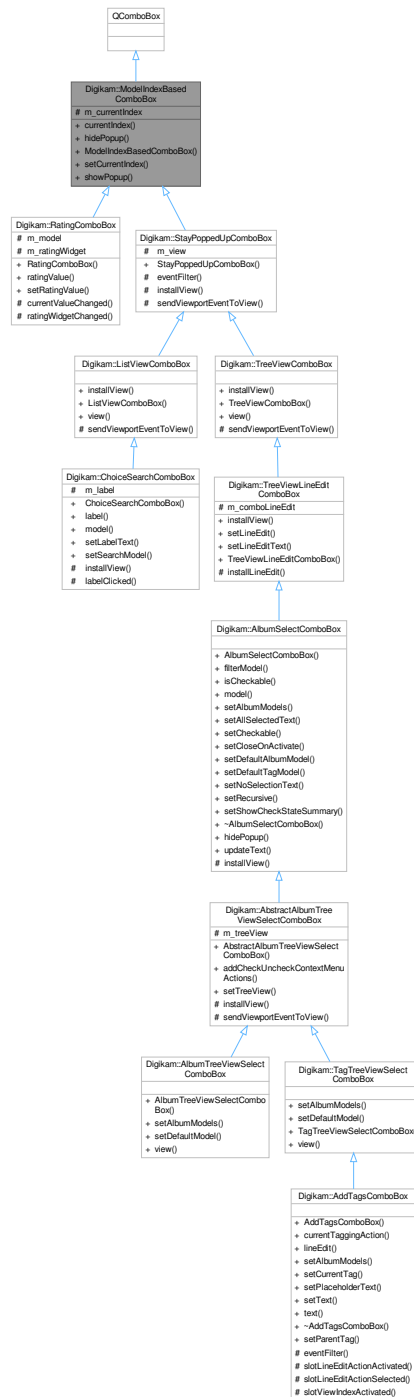
If the given model is != null, the model is used to populate the completion for this text field.

## Parameters

<i>model</i>	to fill from or null for manual mode
<i>uniqueIdRole</i>	a role for which the model will return a unique integer for each entry
<i>displayRole</i>	the role to retrieve the text for completion, default is Qt::DisplayRole.

## 6.1101 Digikam::ModelIndexBasedComboBox Class Reference

Inheritance diagram for Digikam::ModelIndexBasedComboBox:



### Public Member Functions

- QModelIndex **currentIndex** () const
- void **hidePopup** () override

- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

### Protected Attributes

- QPersistentModelIndex **m\_currentIndex**

## 6.1101.1 Constructor & Destructor Documentation

### 6.1101.1.1 QModelIndexBasedComboBox()

```
Digikam::ModelIndexBasedComboBox::ModelIndexBasedComboBox (
    QWidget *const parent = nullptr ) [explicit]
```

QComboBox has a current index based on a single integer. This is not sufficient for more complex models. This class is a combo box that stores a current index based on QModelIndex.



## 6.1102 Digikam::ModelMenu Class Reference

Inheritance diagram for Digikam::ModelMenu:



### Signals

- void **activated** (const QModelIndex &index)
- void **hovered** (const QString &text)

## Public Member Functions

- int **firstSeparator** () const
- int **hoverRole** () const
- QAction \* **makeAction** (const QIcon &icon, const QString &text, QObject \*const parent)
- int **maxRows** () const
- QAbstractItemModel \* **model** () const
- **ModelMenu** (QWidget \*const parent=nullptr)
- QModelIndex **rootIndex** () const
- int **separatorRole** () const
- void **setFirstSeparator** (int offset)
- void **setHoverRole** (int role)
- void **setMaxRows** (int max)
- void **setModel** (QAbstractItemModel \*model)
- void **setRootIndex** (const QModelIndex &index)
- void **setSeparatorRole** (int role)

## Protected Member Functions

- void **createMenu** (const QModelIndex &parent, int max, QMenu \*parentMenu=nullptr, QMenu \*menu=nullptr)  
*put all of the children of parent into menu up to max*
- virtual void **postPopulated** ()  
*add any actions after the tree*
- virtual bool **prePopulated** ()  
*add any actions before the tree, return true if any actions are added.*

### 6.1102.1 Detailed Description

A QMenu that is dynamically populated from a QAbstractItemModel

### 6.1102.2 Member Function Documentation

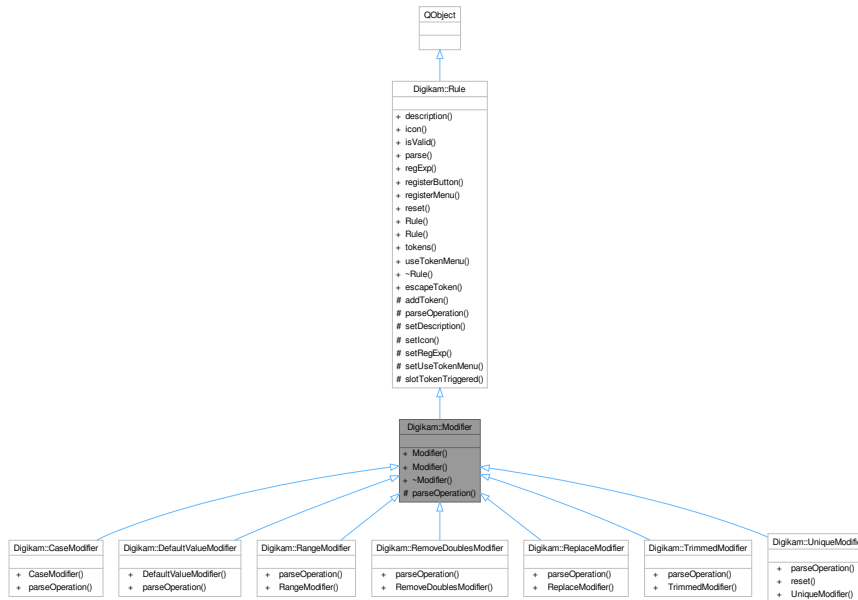
#### 6.1102.2.1 prePopulated()

```
bool Digikam::ModelMenu::prePopulated ( ) [protected], [virtual]
```

Reimplemented in [Digikam::BookmarksMenu](#).

## 6.1103 Digikam::Modifier Class Reference

Inheritance diagram for Digikam::Modifier:



### Public Member Functions

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

### Public Member Functions inherited from Digikam::Rule

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- ParseResults **parse** (ParseSettings &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

### Protected Member Functions

- QString **parseOperation** (ParseSettings &settings, const QRegularExpressionMatch &match) override=0

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void [setDescription](#) (const QString &desc)
- void [setIcon](#) (const QString &pixmap)
- void [setRegExp](#) (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum [IconType](#) { [Action](#) = 0 , [Dialog](#) }

## Signals inherited from [Digikam::Rule](#)

- void [signalTokenTriggered](#) (const QString &)

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString [escapeToken](#) (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void [slotTokenTriggered](#) (const QString &)

## 6.1103.1 Member Function Documentation

### 6.1103.1.1 [parseOperation\(\)](#)

```
QString Digikam::Modifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [pure virtual]
```

TODO: describe me

#### Parameters

<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <a href="#">Option::parse()</a>

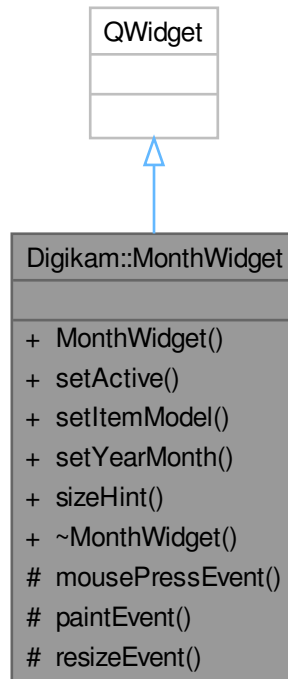
#### Returns

Implements [Digikam::Rule](#).

Implemented in [Digikam::CaseModifier](#), [Digikam::DefaultValueModifier](#), [Digikam::RangeModifier](#), [Digikam::RemoveDoublesModifier](#), [Digikam::ReplaceModifier](#), [Digikam::TrimmedModifier](#), and [Digikam::UniqueModifier](#).

## 6.1104 Digikam::MonthWidget Class Reference

Inheritance diagram for Digikam::MonthWidget:



### Public Member Functions

- **MonthWidget** (`QWidget *const parent`)
- void **setActive** (`bool val`)
- void **setItemModel** (`ItemFilterModel *const model`)
- void **setYearMonth** (`int year, int month`)
- `QSize` **sizeHint** () const override

### Protected Member Functions

- void **mousePressEvent** (`QMouseEvent *e`) override
- void **paintEvent** (`QPaintEvent *`) override
- void **resizeEvent** (`QResizeEvent *e`) override

## 6.1105 Digikam::MysqlAdminBinary Class Reference

Inheritance diagram for Digikam::MysqlAdminBinary:



### Additional Inherited Members

### Public Slots inherited from [Digikam::DBinaryIface](#)

- virtual void **slotAddPossibleSearchDirectory** (const QString &dir)

- virtual void **slotAddSearchDirectory** (const QString &dir)
- virtual void **slotNavigateAndCheck** ()

### Signals inherited from [Digikam::DBinaryIface](#)

- void **signalBinaryValid** ()
- void **signalSearchDirectoryAdded** (const QString &dir)

### Public Member Functions inherited from [Digikam::DBinaryIface](#)

- virtual QString **baseName** () const
- virtual bool **checkDir** ()
- virtual bool **checkDirForPath** (const QString &path)
- **DBinaryIface** (const QString &binaryName, const QString &minimalVersion, const QString &header, const int headerLine, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- **DBinaryIface** (const QString &binaryName, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- const QString & **description** () const
- bool **developmentVersion** () const
- virtual QString **directory** () const
- bool **hasError** () const
- bool **isFound** () const
- bool **isValid** () const
- virtual QString **minimalVersion** () const
- virtual QString **path** () const
- virtual QString **path** (const QString &dir) const
- virtual QString **projectName** () const
- virtual bool **recheckDirectories** ()
- virtual void **setup** (const QString &prev=QString())
- virtual QUrl **url** () const
- const QString & **version** () const
- bool **versionsRight** () const
- bool **versionsRight** (const float) const

### Static Public Member Functions inherited from [Digikam::DBinaryIface](#)

- static QString **goodBaseName** (const QString &b)

### Protected Member Functions inherited from [Digikam::DBinaryIface](#)

- QString **findHeader** (const QStringList &output, const QString &header) const
- virtual bool **parseHeader** (const QString &output)
- virtual QString **readConfig** ()
- void **setVersion** (QString &version)
- virtual void **writeConfig** ()

## Protected Attributes inherited from [Digikam::DBinaryIface](#)

- const QStringList **m\_binaryArguments**
- const QString **m\_binaryBaseName**
- QLabel \* **m\_binaryLabel** = nullptr
- const bool **m\_checkVersion**
- const QString **m\_configGroup**
- QString **m\_description**
- bool **m\_developmentVersion** = false
- QLabel \* **m\_downloadButton** = nullptr
- bool **m\_hasError** = false
- const int **m\_headerLine**
- const QString **m\_headerStarts**
- bool **m\_isFound** = false
- QLineEdit \* **m\_lineEdit** = nullptr
- const QString **m\_minimalVersion**
- QPushButton \* **m\_pathButton** = nullptr
- QString **m\_pathDir** = QLatin1String("")
- QFrame \* **m\_pathWidget** = nullptr
- const QString **m\_projectName**
- QSet< QString > **m\_searchPaths**
- QLabel \* **m\_statusIcon** = nullptr
- const QUrl **m\_url**
- QString **m\_version** = QLatin1String("")
- QLabel \* **m\_versionLabel** = nullptr



## 6.1106 Digikam::MysqlInitBinary Class Reference

Inheritance diagram for Digikam::MysqlInitBinary:



### Additional Inherited Members

### Public Slots inherited from [Digikam::DBinaryIface](#)

- virtual void `slotAddPossibleSearchDirectory` (const QString &dir)

- virtual void **slotAddSearchDirectory** (const QString &dir)
- virtual void **slotNavigateAndCheck** ()

### Signals inherited from [Digikam::DBinaryIface](#)

- void **signalBinaryValid** ()
- void **signalSearchDirectoryAdded** (const QString &dir)

### Public Member Functions inherited from [Digikam::DBinaryIface](#)

- virtual QString **baseName** () const
- virtual bool **checkDir** ()
- virtual bool **checkDirForPath** (const QString &path)
- **DBinaryIface** (const QString &binaryName, const QString &minimalVersion, const QString &header, const int headerLine, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- **DBinaryIface** (const QString &binaryName, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- const QString & **description** () const
- bool **developmentVersion** () const
- virtual QString **directory** () const
- bool **hasError** () const
- bool **isFound** () const
- bool **isValid** () const
- virtual QString **minimalVersion** () const
- virtual QString **path** () const
- virtual QString **path** (const QString &dir) const
- virtual QString **projectName** () const
- virtual bool **recheckDirectories** ()
- virtual void **setup** (const QString &prev=QString())
- virtual QUrl **url** () const
- const QString & **version** () const
- bool **versionsRight** () const
- bool **versionsRight** (const float) const

### Static Public Member Functions inherited from [Digikam::DBinaryIface](#)

- static QString **goodBaseName** (const QString &b)

### Protected Member Functions inherited from [Digikam::DBinaryIface](#)

- QString **findHeader** (const QStringList &output, const QString &header) const
- virtual bool **parseHeader** (const QString &output)
- virtual QString **readConfig** ()
- void **setVersion** (QString &version)
- virtual void **writeConfig** ()

**Protected Attributes inherited from [Digikam::DBinaryIface](#)**

- const QStringList **m\_binaryArguments**
- const QString **m\_binaryBaseName**
- QLabel \* **m\_binaryLabel** = nullptr
- const bool **m\_checkVersion**
- const QString **m\_configGroup**
- QString **m\_description**
- bool **m\_developmentVersion** = false
- QLabel \* **m\_downloadButton** = nullptr
- bool **m\_hasError** = false
- const int **m\_headerLine**
- const QString **m\_headerStarts**
- bool **m\_isFound** = false
- QLineEdit \* **m\_lineEdit** = nullptr
- const QString **m\_minimalVersion**
- QPushButton \* **m\_pathButton** = nullptr
- QString **m\_pathDir** = QLatin1String("")
- QFrame \* **m\_pathWidget** = nullptr
- const QString **m\_projectName**
- QSet< QString > **m\_searchPaths**
- QLabel \* **m\_statusIcon** = nullptr
- const QUrl **m\_url**
- QString **m\_version** = QLatin1String("")
- QLabel \* **m\_versionLabel** = nullptr

## 6.1107 Digikam::MysqlServerBinary Class Reference

Inheritance diagram for Digikam::MysqlServerBinary:



### Additional Inherited Members

### Public Slots inherited from [Digikam::DBinaryIface](#)

- virtual void `slotAddPossibleSearchDirectory` (const QString &dir)

- virtual void **slotAddSearchDirectory** (const QString &dir)
- virtual void **slotNavigateAndCheck** ()

### Signals inherited from [Digikam::DBinaryIface](#)

- void **signalBinaryValid** ()
- void **signalSearchDirectoryAdded** (const QString &dir)

### Public Member Functions inherited from [Digikam::DBinaryIface](#)

- virtual QString **baseName** () const
- virtual bool **checkDir** ()
- virtual bool **checkDirForPath** (const QString &path)
- **DBinaryIface** (const QString &binaryName, const QString &minimalVersion, const QString &header, const int headerLine, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- **DBinaryIface** (const QString &binaryName, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- const QString & **description** () const
- bool **developmentVersion** () const
- virtual QString **directory** () const
- bool **hasError** () const
- bool **isFound** () const
- bool **isValid** () const
- virtual QString **minimalVersion** () const
- virtual QString **path** () const
- virtual QString **path** (const QString &dir) const
- virtual QString **projectName** () const
- virtual bool **recheckDirectories** ()
- virtual void **setup** (const QString &prev=QString())
- virtual QUrl **url** () const
- const QString & **version** () const
- bool **versionsRight** () const
- bool **versionsRight** (const float) const

### Static Public Member Functions inherited from [Digikam::DBinaryIface](#)

- static QString **goodBaseName** (const QString &b)

### Protected Member Functions inherited from [Digikam::DBinaryIface](#)

- QString **findHeader** (const QStringList &output, const QString &header) const
- virtual bool **parseHeader** (const QString &output)
- virtual QString **readConfig** ()
- void **setVersion** (QString &version)
- virtual void **writeConfig** ()

## Protected Attributes inherited from [Digikam::DBinaryIface](#)

- const QStringList **m\_binaryArguments**
- const QString **m\_binaryBaseName**
- QLabel \* **m\_binaryLabel** = nullptr
- const bool **m\_checkVersion**
- const QString **m\_configGroup**
- QString **m\_description**
- bool **m\_developmentVersion** = false
- QLabel \* **m\_downloadButton** = nullptr
- bool **m\_hasError** = false
- const int **m\_headerLine**
- const QString **m\_headerStarts**
- bool **m\_isFound** = false
- QLineEdit \* **m\_lineEdit** = nullptr
- const QString **m\_minimalVersion**
- QPushButton \* **m\_pathButton** = nullptr
- QString **m\_pathDir** = QLatin1String("")
- QFrame \* **m\_pathWidget** = nullptr
- const QString **m\_projectName**
- QSet< QString > **m\_searchPaths**
- QLabel \* **m\_statusIcon** = nullptr
- const QUrl **m\_url**
- QString **m\_version** = QLatin1String("")
- QLabel \* **m\_versionLabel** = nullptr

## 6.1108 Digikam::MysqlUpgradeBinary Class Reference

Inheritance diagram for Digikam::MysqlUpgradeBinary:



### Additional Inherited Members

### Public Slots inherited from [Digikam::DBinaryIface](#)

- virtual void `slotAddPossibleSearchDirectory` (const QString &dir)

- virtual void **slotAddSearchDirectory** (const QString &dir)
- virtual void **slotNavigateAndCheck** ()

### Signals inherited from [Digikam::DBinaryIface](#)

- void **signalBinaryValid** ()
- void **signalSearchDirectoryAdded** (const QString &dir)

### Public Member Functions inherited from [Digikam::DBinaryIface](#)

- virtual QString **baseName** () const
- virtual bool **checkDir** ()
- virtual bool **checkDirForPath** (const QString &path)
- **DBinaryIface** (const QString &binaryName, const QString &minimalVersion, const QString &header, const int headerLine, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- **DBinaryIface** (const QString &binaryName, const QString &projectName, const QString &url, const QString &pluginName, const QStringList &args=QStringList(), const QString &desc=QString())
- const QString & **description** () const
- bool **developmentVersion** () const
- virtual QString **directory** () const
- bool **hasError** () const
- bool **isFound** () const
- bool **isValid** () const
- virtual QString **minimalVersion** () const
- virtual QString **path** () const
- virtual QString **path** (const QString &dir) const
- virtual QString **projectName** () const
- virtual bool **recheckDirectories** ()
- virtual void **setup** (const QString &prev=QString())
- virtual QUrl **url** () const
- const QString & **version** () const
- bool **versionsRight** () const
- bool **versionsRight** (const float) const

### Static Public Member Functions inherited from [Digikam::DBinaryIface](#)

- static QString **goodBaseName** (const QString &b)

### Protected Member Functions inherited from [Digikam::DBinaryIface](#)

- QString **findHeader** (const QStringList &output, const QString &header) const
- virtual bool **parseHeader** (const QString &output)
- virtual QString **readConfig** ()
- void **setVersion** (QString &version)
- virtual void **writeConfig** ()

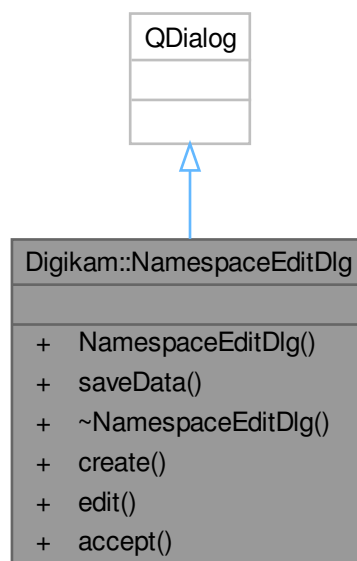


## Protected Attributes inherited from [Digikam::DBinaryIface](#)

- const QStringList **m\_binaryArguments**
- const QString **m\_binaryBaseName**
- QLabel \* **m\_binaryLabel** = nullptr
- const bool **m\_checkVersion**
- const QString **m\_configGroup**
- QString **m\_description**
- bool **m\_developmentVersion** = false
- QLabel \* **m\_downloadButton** = nullptr
- bool **m\_hasError** = false
- const int **m\_headerLine**
- const QString **m\_headerStarts**
- bool **m\_isFound** = false
- QLineEdit \* **m\_lineEdit** = nullptr
- const QString **m\_minimalVersion**
- QPushButton \* **m\_pathButton** = nullptr
- QString **m\_pathDir** = QLatin1String("")
- QFrame \* **m\_pathWidget** = nullptr
- const QString **m\_projectName**
- QSet< QString > **m\_searchPaths**
- QLabel \* **m\_statusIcon** = nullptr
- const QUrl **m\_url**
- QString **m\_version** = QLatin1String("")
- QLabel \* **m\_versionLabel** = nullptr

## 6.1109 Digikam::NamespaceEditDlg Class Reference

Inheritance diagram for Digikam::NamespaceEditDlg:



### Public Slots

- void **accept** () override

### Public Member Functions

- **NamespaceEditDlg** (bool create, [NamespaceEntry](#) &entry, QWidget \*const parent=nullptr)
- void **saveData** ([NamespaceEntry](#) &entry)

### Static Public Member Functions

- static bool **create** (QWidget \*const parent, [NamespaceEntry](#) &entry)
- static bool **edit** (QWidget \*const parent, [NamespaceEntry](#) &entry)

## 6.1110 Digikam::NamespaceEntry Class Reference

The [NamespaceEntry](#) class provide a simple container for dmetadata namespaces variables, such as names, what types of data expects and extra xml tags.

### Public Types

- enum **NamespaceType** {  
  **TAGS** = 0 , **TITLE** = 1 , **RATING** = 2 , **COMMENT** = 3 ,  
  **PICKLABEL** = 4 , **COLORLABEL** = 5 }
- enum **NsSubspace** { **EXIF** = 0 , **IPTC** = 1 , **XMP** = 2 }
- enum **SpecialOptions** {  
  **NO\_OPTS** = 0 , **COMMENT\_ALTLANG** = 1 , **COMMENT\_ATLLANGLIST** = 2 , **COMMENT\_XMP** = 3 ,  
  **COMMENT\_JPEG** = 4 , **TAG\_XMPBAG** = 5 , **TAG\_XMPSEQ** = 6 , **TAG\_ACDSEE** = 7 }
- enum **TagType** { **TAG** = 0 , **TAGPATH** = 1 }

### Public Member Functions

- **NamespaceEntry** (const [NamespaceEntry](#) &other)

### Static Public Member Functions

- static QString **DM\_COLORLABEL\_CONTAINER** ()
- static QString **DM\_COMMENT\_CONTAINER** ()
- static QString **DM\_PICKLABEL\_CONTAINER** ()
- static QString **DM\_RATING\_CONTAINER** ()
- static QString **DM\_TAG\_CONTAINER** ()
- static QString **DM\_TITLE\_CONTAINER** ()

## Public Attributes

- QString **alternativeName**
- QList< int > [convertRatio](#)
- int **index** = -1
- bool **isDefault** = true
- bool **isDisabled** = false
- QString [namespaceName](#)
- NamespaceType **nsType** = TAGS
- SpecialOptions **secondNameOpts** = NO\_OPTS
- QString **separator**
- SpecialOptions **specialOpts** = NO\_OPTS
- NsSubspace **subspace** = XMP
- TagType **tagPaths** = TAGPATH

## 6.1110.1 Member Data Documentation

### 6.1110.1.1 convertRatio

QList<int> Digikam::NamespaceEntry::convertRatio

Rating Options

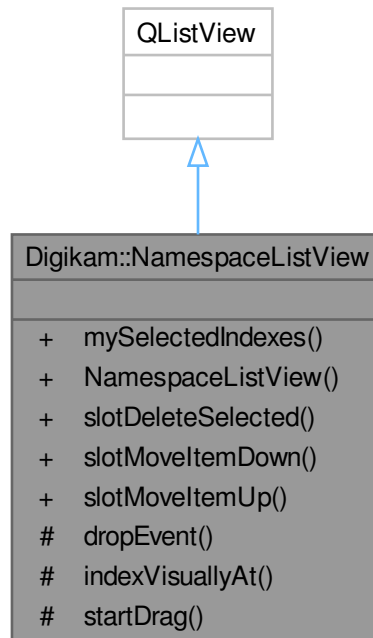
### 6.1110.1.2 namespaceName

QString Digikam::NamespaceEntry::namespaceName

Tag Options

## 6.1111 Digikam::NamespaceListView Class Reference

Inheritance diagram for Digikam::NamespaceListView:



### Public Slots

- void **slotDeleteSelected** ()  
*slotDeleteSelected - delete selected item from Quick Access List*
- void **slotMoveItemDown** ()
- void **slotMoveItemUp** ()

### Signals

- void **signalItemsChanged** ()  
*contextMenuEvent - reimplemented method from QListView to handle custom context menu*

### Public Member Functions

- QModelIndexList **mySelectedIndexes** ()
- **NamespaceListView** (QWidget \*const parent=nullptr)

### Protected Member Functions

- void **dropEvent** (QDropEvent \*e) override
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **startDrag** (Qt::DropActions supportedActions) override

## 6.1111.1 Member Function Documentation

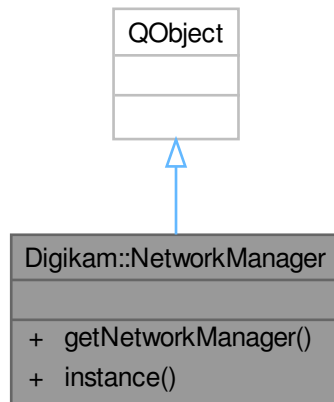
### 6.1111.1.1 startDrag()

```
void Digikam::NamespaceListView::startDrag (
    Qt::DropActions supportedActions ) [override], [protected]
```

Reimplemented methods to enable custom drag-n-drop in QListView

## 6.1112 Digikam::NetworkManager Class Reference

Inheritance diagram for Digikam::NetworkManager:



### Public Member Functions

- `QNetworkAccessManager * getNetworkManager (QObject *const object) const`

### Static Public Member Functions

- `static NetworkManager * instance ()`

### Friends

- class `NetworkManagerCreator`

## 6.1112.1 Member Function Documentation

### 6.1112.1.1 `getNetworkManager()`

```
QNetworkAccessManager * Digikam::NetworkManager::getNetworkManager (
    QObject *const object ) const
```

Get the current QNetworkAccessManager or create a new QNetworkAccessManager if the passed QObject runs on a different thread.

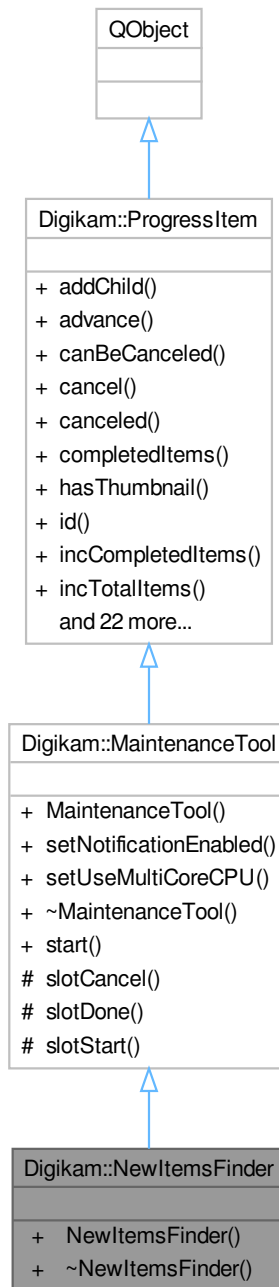
### 6.1112.1.2 `instance()`

```
NetworkManager * Digikam::NetworkManager::instance ( ) [static]
```

Global instance of internal network manager. All accessor methods are thread-safe.

## 6.1113 Digikam::NewItemsFinder Class Reference

Inheritance diagram for Digikam::NewItemsFinder:



### Public Types

- enum `FinderMode` { `CompleteCollectionScan`, `ScanDeferredFiles`, `ScheduleCollectionScan` }

## Public Member Functions

- **NewItemsFinder** (const [FinderMode](#) mode=[CompleteCollectionScan](#), const QStringList &foldersToScan=QStringList(), [ProgressItem](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- **MaintenanceTool** (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)
- virtual void [setUseMultiCoreCPU](#) (bool)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- **ProgressItem** ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool [canBeCanceled](#), bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void [setLabel](#) (const QString &v)
- void [setProgress](#) (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void [setShowAtStart](#) (bool showAtStart)
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void [setStatus](#) (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void [setThumbnail](#) (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void [setUsesBusyIndicator](#) (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool [showAtStart](#) () const
- const QString & [status](#) () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool [usesBusyIndicator](#) () const



## Additional Inherited Members

### Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

### Signals inherited from [Digikam::MaintenanceTool](#)

- void [signalCanceled](#) ()
- void [signalComplete](#) ()

### Signals inherited from [Digikam::ProgressItem](#)

- void [progressItemAdded](#) ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void [progressItemCanceled](#) ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void [progressItemCompleted](#) ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void [progressItemLabel](#) ([ProgressItem](#) \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void [progressItemProgress](#) ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void [progressItemStatus](#) ([ProgressItem](#) \*item, const QString &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

### Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.1113.1 Member Enumeration Documentation

### 6.1113.1.1 FinderMode

```
enum Digikam::NewItemFinder::FinderMode
```

## Enumerator

CompleteCollectionScan	Scan whole collection immediately.
ScanDeferredFiles	Defer whole collection scan.
ScheduleCollectionScan	Scan immediately folders list passed in constructor.

## 6.1114 Digikam::NewlyAppearedFile Class Reference

### Public Member Functions

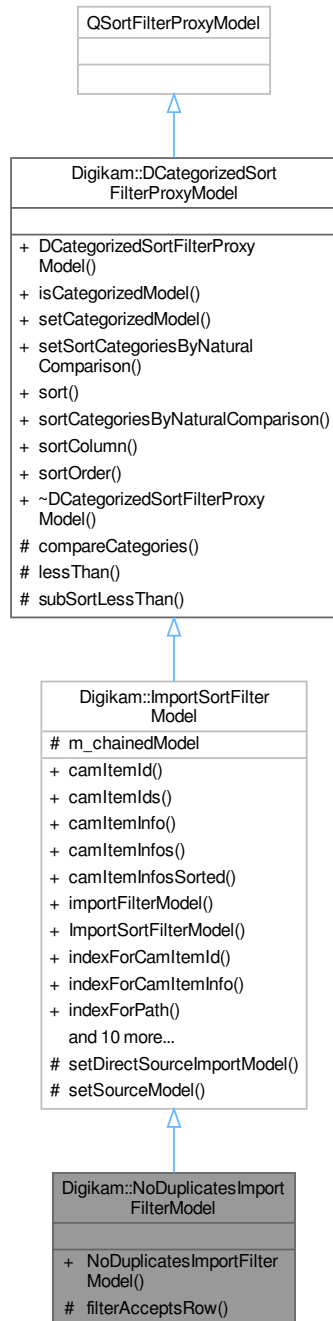
- **NewlyAppearedFile** (int albumId, const QString &fileName)
- bool **operator==** (const [NewlyAppearedFile](#) &other) const

### Public Attributes

- int **albumId** = 0
- QString **fileName**

## 6.1115 Digikam::NoDuplicatesImportFilterModel Class Reference

Inheritance diagram for Digikam::NoDuplicatesImportFilterModel:



### Public Member Functions

- **NoDuplicatesImportFilterModel** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ImportSortFilterModel](#)

- qlonglong **camlItemId** (const QModelIndex &index) const
- QList< qlonglong > **camlItemIds** (const QList< QModelIndex > &indexes) const
- [CamlItemInfo](#) **camlItemInfo** (const QModelIndex &index) const
- QList< [CamlItemInfo](#) > **camlItemInfos** (const QList< QModelIndex > &indexes) const
- QList< [CamlItemInfo](#) > **camlItemInfosSorted** () const
- virtual [ImportFilterModel](#) \* **importFilterModel** () const  
*Returns this, any chained [ImportFilterModel](#), or 0.*
- **ImportSortFilterModel** (QObject \*const parent=nullptr)
- QModelIndex **indexForCamlItemId** (qlonglong id) const
- QModelIndex **indexForCamlItemInfo** (const [CamlItemInfo](#) &info) const
- QModelIndex **indexForPath** (const QString &filePath) const
- QModelIndex **mapFromDirectSourceToSourceImportModel** (const QModelIndex &sourceModelIndex) const
- QModelIndex **mapFromSourceImportModel** (const QModelIndex &importModelIndex) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceImportModel** (const QModelIndex &proxyIndex) const
- void **setSourceFilterModel** ([ImportSortFilterModel](#) \*const sourceModel)
- void **setSourceImportModel** ([ImportItemModel](#) \*const sourceModel)
- [ImportSortFilterModel](#) \* **sourceFilterModel** () const
- [ImportItemModel](#) \* **sourceImportModel** () const

## Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- **DCategorizedSortFilterProxyModel** (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void **setCategorizedModel** (bool categorizedModel)
- void **setSortCategoriesByNaturalComparison** (bool [sortCategoriesByNaturalComparison](#))
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool **sortCategoriesByNaturalComparison** () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

## Protected Member Functions

- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override

## Protected Member Functions inherited from [Digikam::ImportSortFilterModel](#)

- virtual void **setDirectSourceImportModel** ([ImportItemModel](#) \*const sourceModel)  
*Reimplement if needed. Called only when model shall be set as (direct) sourceModel.*
- void **setSourceModel** (QAbstractItemModel \*sourceModel) override

## Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- virtual int **compareCategories** (const QModelIndex &left, const QModelIndex &right) const
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool **subSortLessThan** (const QModelIndex &left, const QModelIndex &right) const

### Additional Inherited Members

### Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

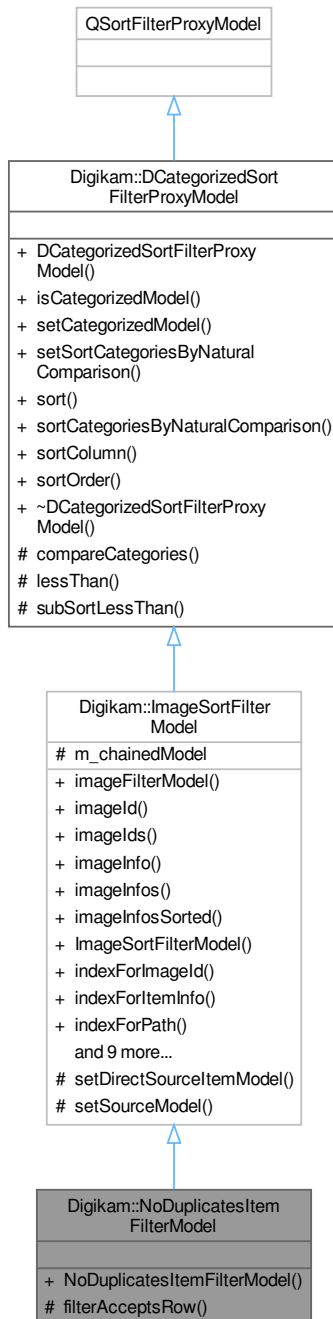
- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

### Protected Attributes inherited from [Digikam::ImportSortFilterModel](#)

- [ImportSortFilterModel](#) \* [m\\_chainedModel](#) = nullptr

## 6.1116 Digikam::NoDuplicatesItemFilterModel Class Reference

Inheritance diagram for Digikam::NoDuplicatesItemFilterModel:



### Public Member Functions

- **NoDuplicatesItemFilterModel** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ImageSortFilterModel](#)

- virtual [ItemFilterModel](#) \* [imageFilterModel](#) () const
- qlonglong [imageld](#) (const QModelIndex &index) const
- QList< qlonglong > [imageIds](#) (const QList< QModelIndex > &indexes) const
- [ItemInfo](#) [imageInfo](#) (const QModelIndex &index) const
- QList< [ItemInfo](#) > [imageInfos](#) (const QList< QModelIndex > &indexes) const
- QList< [ItemInfo](#) > [imageInfosSorted](#) () const
- [ImageSortFilterModel](#) (QObject \*const parent=nullptr)
- QModelIndex [indexForImageId](#) (qlonglong id) const
- QModelIndex [indexForItemInfo](#) (const [ItemInfo](#) &info) const
- QModelIndex [indexForPath](#) (const QString &filePath) const
- QModelIndex [mapFromDirectSourceToSourceItemModel](#) (const QModelIndex &sourceModel\_index) const
- QModelIndex [mapFromSourceItemModel](#) (const QModelIndex &imagemodel\_index) const
- QList< QModelIndex > [mapListFromSource](#) (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > [mapListToSource](#) (const QList< QModelIndex > &indexes) const
- QModelIndex [mapToSourceItemModel](#) (const QModelIndex &index) const
- void [setSourceFilterModel](#) ([ImageSortFilterModel](#) \*const model)
- void [setSourceItemModel](#) ([ItemModel](#) \*const model)
- [ImageSortFilterModel](#) \* [sourceFilterModel](#) () const
- [ItemModel](#) \* [sourceItemModel](#) () const

## Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- [DCategorizedSortFilterProxyModel](#) (QObject \*const parent=nullptr)
- bool [isCategorizedModel](#) () const
- void [setCategorizedModel](#) (bool categorizedModel)
- void [setSortCategoriesByNaturalComparison](#) (bool [sortCategoriesByNaturalComparison](#))
- void [sort](#) (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool [sortCategoriesByNaturalComparison](#) () const
- int [sortColumn](#) () const
- Qt::SortOrder [sortOrder](#) () const

## Protected Member Functions

- bool [filterAcceptsRow](#) (int source\_row, const QModelIndex &source\_parent) const override

## Protected Member Functions inherited from [Digikam::ImageSortFilterModel](#)

- virtual void [setDirectSourceItemModel](#) ([ItemModel](#) \*const model)
- void [setSourceModel](#) (QAbstractItemModel \*const model) override

*NOTE: made protected.*

## Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- virtual int [compareCategories](#) (const QModelIndex &left, const QModelIndex &right) const
- bool [lessThan](#) (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool [subSortLessThan](#) (const QModelIndex &left, const QModelIndex &right) const

### Additional Inherited Members

### Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

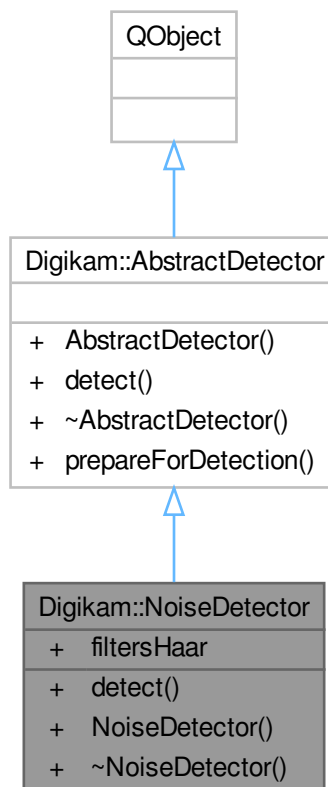
- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

### Protected Attributes inherited from [Digikam::ImageSortFilterModel](#)

- [ImageSortFilterModel](#) \* `m_chainedModel` = nullptr

## 6.1117 Digikam::NoiseDetector Class Reference

Inheritance diagram for Digikam::NoiseDetector:



### Public Types

- typedef `QList< cv::Mat >` **Mat3D**



### Public Member Functions

- float [detect](#) (const cv::Mat &image) const override

### Public Member Functions inherited from [Digikam::AbstractDetector](#)

- **AbstractDetector** (QObject \*const parent=nullptr)

### Static Public Attributes

- static const Mat3D **filtersHaar** = initFiltersHaar()

### Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::AbstractDetector](#)

- static cv::Mat [prepareForDetection](#) (const [DImg](#) &inputImage)

## 6.1117.1 Member Function Documentation

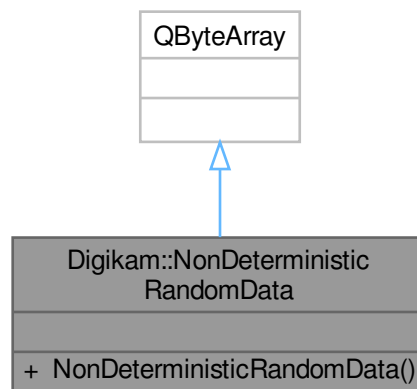
### 6.1117.1.1 detect()

```
float Digikam::NoiseDetector::detect (
    const cv::Mat & image ) const [override], [virtual]
```

Implements [Digikam::AbstractDetector](#).

## 6.1118 Digikam::NonDeterministicRandomData Class Reference

Inheritance diagram for Digikam::NonDeterministicRandomData:



## Public Member Functions

- [NonDeterministicRandomData](#) (int size)

## 6.1118.1 Constructor & Destructor Documentation

### 6.1118.1.1 NonDeterministicRandomData()

```
Digikam::NonDeterministicRandomData::NonDeterministicRandomData (  
    int size ) [explicit]
```

Constructs a QByteArray of given byte size filled with non-deterministic random data. For larger quantities of data, prefer using a [RandomNumberGenerator](#) seeded with non-deterministic data.

## 6.1119 Digikam::NormalizeFilter Class Reference

Inheritance diagram for Digikam::NormalizeFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **NormalizeFilter** ([DImg](#) \*const orgImage, const [DImg](#) \*const reflImage, [QObject](#) \*const parent=nullptr)
- **NormalizeFilter** ([QObject](#) \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static [QString](#) **DisplayableName** ()
- static [QString](#) **FilterIdentifier** ()
- static [QList](#)< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.1119.1 Member Function Documentation

### 6.1119.1.1 filterAction()

`FilterAction` Digikam::NormalizeFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1119.1.2 filterIdentifier()

`QString` Digikam::NormalizeFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

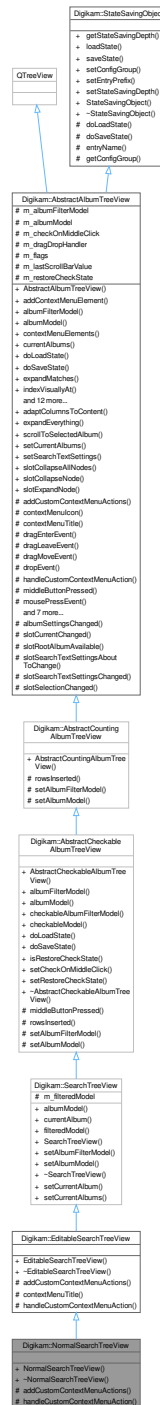
### 6.1119.1.3 readParameters()

```
void Digikam::NormalizeFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1120 Digikam::NormalSearchTreeView Class Reference

Inheritance diagram for Digikam::NormalSearchTreeView:



### Signals

- void [copySearch](#) (SAAlbum \*album)
- void [editSearch](#) (SAAlbum \*album)
- void [newSearch](#) ()

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const [QList](#)< [Album](#) \* > &selectedAlbums)

## Public Member Functions

- [NormalSearchTreeView](#) ([QWidget](#) \*const parent, [searchModel](#) \*const searchModel, [SearchModificationHelper](#) \*const searchModificationHelper)
- [~NormalSearchTreeView](#) () override

## Public Member Functions inherited from [Digikam::EditableSearchTreeView](#)

- [EditableSearchTreeView](#) ([QWidget](#) \*const parent, [searchModel](#) \*const searchModel, [SearchModificationHelper](#) \*const searchModificationHelper)
- [~EditableSearchTreeView](#) () override

## Public Member Functions inherited from [Digikam::SearchTreeView](#)

- [searchModel](#) \* [albumModel](#) () const  
*Note: not filtered by search type.*
- [SAlbum](#) \* [currentAlbum](#) () const
- [searchFilterModel](#) \* [filteredModel](#) () const  
*Contains only the searches with appropriate type - prefer to [albumModel\(\)](#)*
- [SearchTreeView](#) ([QWidget](#) \*const parent=nullptr, [Flags](#) flags=DefaultFlags)
- void [setAlbumFilterModel](#) ([searchFilterModel](#) \*const filteredModel, [CheckableAlbumFilterModel](#) \*const model)
- void [setAlbumModel](#) ([searchModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) ([QWidget](#) \*const parent, [Flags](#) flags)
- [CheckableAlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [albumModel](#) () const
- [CheckableAlbumFilterModel](#) \* [checkableAlbumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [checkableModel](#) () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) ([QWidget](#) \*const parent, [Flags](#) flags)



## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Protected Member Functions

- void **addCustomContextMenuActions** ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override
- void **handleCustomContextMenuAction** (QAction \*action, const [AlbumPointer](#)< [Album](#) > &album) override

## Protected Member Functions inherited from [Digikam::EditableSearchTreeView](#)

- QString **contextMenuTitle** () const override

## Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void **middleButtonPressed** ([Album](#) \*a) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **setAlbumFilterModel** ([CheckableAlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AbstractCheckableAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **setAlbumFilterModel** ([AlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AbstractCountingAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual QPixmap **contextMenuIcon** () const
  - void **dragEnterEvent** (QDragEnterEvent \*e) override
  - void **dragLeaveEvent** (QDragLeaveEvent \*e) override
  - void **dragMoveEvent** (QDragMoveEvent \*e) override
  - void **dropEvent** (QDropEvent \*e) override
  - void **mousePressEvent** (QMouseEvent \*e) override
- Other helper methods.*
- virtual QPixmap  **pixmapForDrag** (const QStyleOptionViewItem &option, QList< QModelIndex > indexes)
  - void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
  - void **rowsInserted** (const QModelIndex &index, int start, int end) override
  - void **setAlbumFilterModel** ([AlbumFilterModel](#) \*const filterModel)
  - void **setAlbumModel** ([AbstractSpecificAlbumModel](#) \*const model)
  - virtual bool **showContextMenuAt** (QContextMenuEvent \*event, [Album](#) \*albumForEvent)
  - void **startDrag** (Qt::DropActions supportedActions) override

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString **entryName** (const QString &base) const
- KConfigGroup **getConfigGroup** () const

### Additional Inherited Members

### Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum **Flag** {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum **StateSavingDepth** { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Public Slots inherited from [Digikam::SearchTreeView](#)

- void **setCurrentAlbum** (int searchId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()
  - slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()
  - slotExpandNode - expands recursively selected nodes*

## Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Attributes inherited from [Digikam::SearchTreeView](#)

- [SearchFilterModel](#) \* [m\\_filteredModel](#) = nullptr

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* [m\\_albumFilterModel](#) = nullptr
- [AbstractSpecificAlbumModel](#) \* [m\\_albumModel](#) = nullptr
- bool [m\\_checkOnMiddleClick](#) = false
- [AlbumModelDragDropHandler](#) \* [m\\_dragDropHandler](#) = nullptr
- Flags [m\\_flags](#) = DefaultFlags
- int [m\\_lastScrollBarValue](#) = 0
- bool [m\\_restoreCheckState](#) = false

### 6.1120.1 Detailed Description

Tree view for all saved "normal" searches. Allows editing and creating searches in the context menu.

Author

jwienke

### 6.1120.2 Constructor & Destructor Documentation

#### 6.1120.2.1 NormalSearchTreeView()

```
Digikam::NormalSearchTreeView::NormalSearchTreeView (
    QWidget *const parent,
    SearchModel *const searchModel,
    SearchModificationHelper *const searchModificationHelper )
```

Constructor.

## Parameters

<i>parent</i>	qt parent
<i>searchModel</i>	the model this view should act on
<i>searchModificationHelper</i>	the modification helper object used to perform operations on the displayed searches

**6.1120.2.2 ~NormalSearchTreeView()**

```
Digikam::NormalSearchTreeView::~NormalSearchTreeView ( ) [override]
```

Destructor.

**6.1120.3 Member Function Documentation****6.1120.3.1 addCustomContextMenuActions()**

```
void Digikam::NormalSearchTreeView::addCustomContextMenuActions (
    ContextMenuHelper & cmh,
    Album * album ) [override], [protected], [virtual]
```

Adds actions to delete or rename existing searches.

Reimplemented from [Digikam::EditableSearchTreeView](#).

**6.1120.3.2 copySearch**

```
void Digikam::NormalSearchTreeView::copySearch (
    SAlbum * album ) [signal]
```

Emitted if the given search shall be copied.

## Parameters

<i>album</i>	search to copy
--------------	----------------

**6.1120.3.3 editSearch**

```
void Digikam::NormalSearchTreeView::editSearch (
    SAlbum * album ) [signal]
```

Emitted if the given search shall be edited.

## Parameters

<i>album</i>	search to edit
--------------	----------------

### 6.1120.3.4 handleCustomContextMenuAction()

```
void Digikam::NormalSearchTreeView::handleCustomContextMenuAction (
    QAction * action,
    const AlbumPointer< Album > & album ) [override], [protected], [virtual]
```

Handles deletion and renaming actions.

Reimplemented from [Digikam::EditableSearchTreeView](#).

### 6.1120.3.5 newSearch

```
void Digikam::NormalSearchTreeView::newSearch ( ) [signal]
```

Emitted of a new search shall be created.

## 6.1121 Digikam::NRContainer Class Reference

### Public Attributes

- double **softness** [3] = { 0.9 }  
*Y, Cb, Cr softness.*
- double **thresholds** [3] = { 1.2 }  
*Y, Cb, Cr thresholds.*

### 6.1121.1 Member Data Documentation

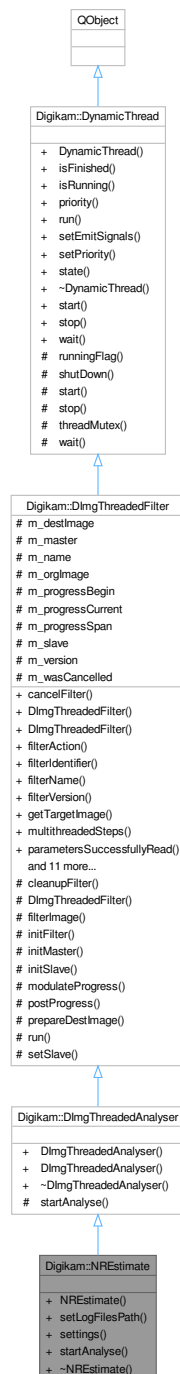
#### 6.1121.1.1 thresholds

```
double Digikam::NRContainer::thresholds[3] = { 1.2 }
```

Separated values per channel

## 6.1122 Digikam::NREstimate Class Reference

Inheritance diagram for Digikam::NREstimate:



### Public Member Functions

- [NREstimate](#) (`DImg *const img`, `QObject *const parent=nullptr`)
- void [setLogFilesPath](#) (`const QString &path`)
- [NRContainer settings](#) () const
- void [startAnalyse](#) () override

## Public Member Functions inherited from [Digikam::DImgThreadedAnalyser](#)

- **DImgThreadedAnalyser** ([DImg](#) \*const orgImage, [QObject](#) \*const parent=nullptr, const [QString](#) &name=QString())  
*Constructs an image analyser with all arguments (ready to use). The given original image will be copied. You need to call [startFilter\(\)](#) to start the threaded computation. To run analyser without to use multithreading, call [startFilterDirectly\(\)](#).*
- **DImgThreadedAnalyser** ([QObject](#) \*const parent=nullptr, const [QString](#) &name=QString())  
*Constructs a filter without argument. You need to call [setupFilter\(\)](#) and [startFilter\(\)](#) to start the threaded computation. To run filter without to use multithreading, call [startFilterDirectly\(\)](#).*

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=QString())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=QString())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false



## 6.1122.1 Constructor & Destructor Documentation

### 6.1122.1.1 NREstimate()

```
Digikam::NREstimate::NREstimate (
    DImg *const img,
    QObject *const parent = nullptr ) [explicit]
```

Standard constructor with image container to parse

## 6.1122.2 Member Function Documentation

### 6.1122.2.1 setLogFilePath()

```
void Digikam::NREstimate::setLogFilePath (
    const QString & path )
```

To set image path where log files will be created to host computation algorithm results, for hacking purpose. If path is not set, no log files will be created.

### 6.1122.2.2 settings()

```
NRContainer Digikam::NREstimate::settings ( ) const
```

Return all Wavelets noise reduction settings computed by image analys.

### 6.1122.2.3 startAnalyse()

```
void Digikam::NREstimate::startAnalyse ( ) [override], [virtual]
```

Perform estimate noise.

Implements [Digikam::DImgThreadedAnalyser](#).

## 6.1123 Digikam::NRFilter Class Reference

Inheritance diagram for Digikam::NRFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **NRFilter** ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [NRContainer](#) &settings)
- **NRFilter** ([QObject](#) \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static [QString](#) **DisplayableName** ()
- static [QString](#) **FilterIdentifier** ()
- static void **srgb2ycbcr** (float \*\*const fimg, uint size)
- static [QList](#)< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.1123.1 Member Function Documentation

### 6.1123.1.1 filterAction()

`FilterAction` Digikam::NRFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1123.1.2 filterIdentifier()

`QString` Digikam::NRFilter::filterIdentifier ( ) const [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

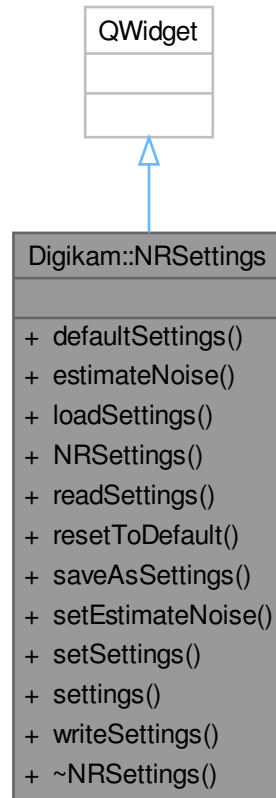
### 6.1123.1.3 readParameters()

```
void Digikam::NRFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1124 Digikam::NRSettings Class Reference

Inheritance diagram for Digikam::NRSettings:



### Signals

- void **signalEstimateNoise** ()
- void **signalSettingsChanged** ()

### Public Member Functions

- [NRContainer](#) **defaultSettings** () const
- bool **estimateNoise** () const
- void **loadSettings** ()
- **NRSettings** (QWidget \*const parent)
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **saveAsSettings** ()
- void **setEstimateNoise** (bool b)
- void **setSettings** (const [NRContainer](#) &settings)
- [NRContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.1125 Digikam::OilPaintFilter Class Reference

Inheritance diagram for Digikam::OilPaintFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **OilPaintFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, int brushSize=1, int smoothness=30)
- **OilPaintFilter** (QObject \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static QString [DisplayableName](#) ()
- static QString [FilterIdentifier](#) ()
- static QList< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }



## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.1125.1 Member Function Documentation

### 6.1125.1.1 filterAction()

`FilterAction` Digikam::OilPaintFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1125.1.2 filterIdentifier()

`QString` Digikam::OilPaintFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

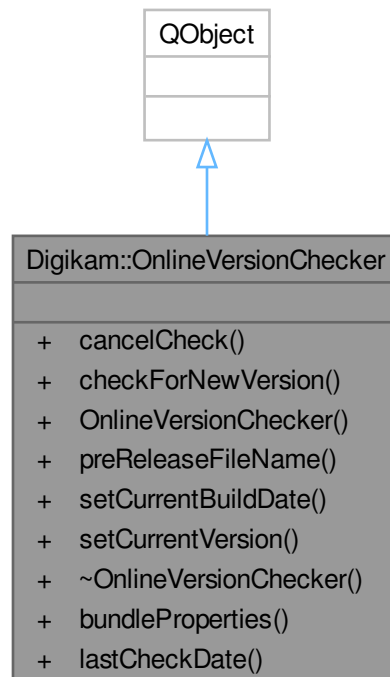
### 6.1125.1.3 readParameters()

`void` Digikam::OilPaintFilter::readParameters (   
     const `FilterAction` & *action* ) [override], [virtual]

Implements [Digikam::DImgThreadedFilter](#).

## 6.1126 Digikam::OnlineVersionChecker Class Reference

Inheritance diagram for Digikam::OnlineVersionChecker:



## Signals

- void **signalNewVersionAvailable** (const QString &version)
- void **signalNewVersionCheckError** (const QString &error)

## Public Member Functions

- void **cancelCheck** ()
- void **checkForNewVersion** ()
- **OnlineVersionChecker** (QObject \*const parent, bool checkPreRelease=false)
- QString **preReleaseFileName** () const
- void **setCurrentBuildDate** (const QDateTime &dt)
- void **setCurrentVersion** (const QString &version)

## Static Public Member Functions

- static bool **bundleProperties** (QString &arch, QString &ext, QString &qtVersion, QString &dir)
- static QString **lastCheckDate** ()

## 6.1126.1 Member Function Documentation

### 6.1126.1.1 bundleProperties()

```
bool Digikam::OnlineVersionChecker::bundleProperties (
    QString & arch,
    QString & ext,
    QString & qtVersion,
    QString & dir ) [static]
```

Return true if the system and architecture are supported by the bundle workflow. 'arch' is the relevant prefix for the bundle architecture. 'ext' is the relevant bundle file extension. 'qtVersion' is the relevant version of Qt used in the bundle file-name. 'dir' is the subdirectory if any to get the bundle file.

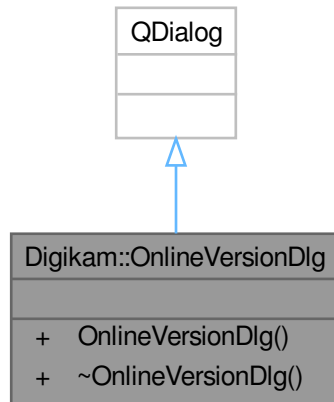
### 6.1126.1.2 lastCheckDate()

```
QString Digikam::OnlineVersionChecker::lastCheckDate ( ) [static]
```

Return the last date as string when have been performed a check for new version.

## 6.1127 Digikam::OnlineVersionDlg Class Reference

Inheritance diagram for Digikam::OnlineVersionDlg:



### Signals

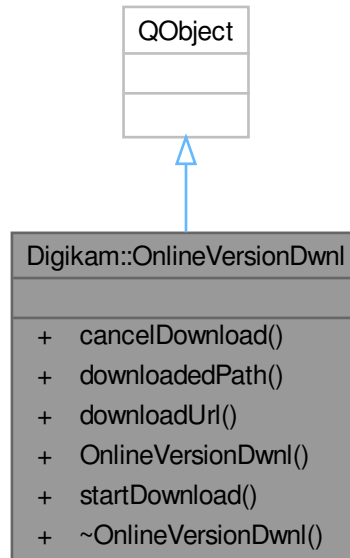
- void **signalSetupUpdate** ()

### Public Member Functions

- **OnlineVersionDlg** (QWidget \*const parent=nullptr, const QString &version=QLatin1String(digikam\_↵  
version\_short), const QDateTime &buildDt=digiKamBuildDate(), bool checkPreRelease=false, bool update↵  
WithDebug=false)

## 6.1128 Digikam::OnlineVersionDwnl Class Reference

Inheritance diagram for Digikam::OnlineVersionDwnl:



### Signals

- void **signalComputeChecksum** ()
- void **signalDownloadError** (const QString &error)
- void **signalDownloadProgress** (qint64 bytesReceived, qint64 bytesTotal)

### Public Member Functions

- void **cancelDownload** ()
- QString **downloadedPath** () const
- QString **downloadUrl** () const
- **OnlineVersionDwnl** (QObject \*const parent=nullptr, bool checkPreRelease=false, bool updateWith↔ Debug=false)
- void **startDownload** (const QString &version)

## 6.1129 Digikam::OpenCVDNNFaceDetector Class Reference

### Public Member Functions

- std::vector< cv::Rect > **cvDetectFaces** (const cv::Mat &inputImage, const cv::Size &paddedSize)
- QList< QRect > **detectFaces** (const cv::Mat &inputImage, const cv::Size &paddedSize)
- **OpenCVDNNFaceDetector** ([DetectorNNModel](#) model=[DetectorNNModel::DNNDetectorYuNet](#))

- cv::Mat **prepareForDetection** (const [DImg](#) &inputImage, cv::Size &paddedSize) const
- cv::Mat **prepareForDetection** (const QImage &inputImage, cv::Size &paddedSize) const
- cv::Mat **prepareForDetection** (const QString &inputImagePath, cv::Size &paddedSize) const
- cv::Mat **prepareForDetectionYuNet** (cv::Mat &cvImage, cv::Size &paddedSize) const
- void **setAccuracy** (const int accuracy)
- void **setFaceDetectionSize** ([FaceScanSettings::FaceDetectionSize](#) size)

### Static Public Member Functions

- static int [recommendedImageSizeForDetection](#) ()

## 6.1129.1 Member Function Documentation

### 6.1129.1.1 detectFaces()

```
QList< QRect > Digikam::OpenCVDNNFaceDetector::detectFaces (
    const cv::Mat & inputImage,
    const cv::Size & paddedSize )
```

There is no proof that doing this will help, since face can be detected at various positions (even half, masked faces can be detected), not only frontal. Effort on doing this should be questioned. TODO: Restructure and improve Face Detection module.

void OpenCVDNNFaceDetector::resizeBboxToStandardHumanFace(int& width, int& height) { Human head sizes data. [https://en.wikipedia.org/wiki/Human\\_head#Average\\_head\\_sizes](https://en.wikipedia.org/wiki/Human_head#Average_head_sizes)

```
float maxRatioFrontalFace    = 15.4 / 15.5;
float minRatioNonFrontalFace = 8.6  / 21.6;

float r = width*1.0/height, rReference;

if      ((r >= minRatioNonFrontalFace*0.9) && r <= (maxRatioFrontalFace * 1.1))
{
    rReference = r;
}
else if (r <= 0.25)
{
    rReference = r * 1.5;
}
else if (r >= 4)
{
    rReference = r / 1.5;
}
else if (r < minRatioNonFrontalFace * 0.9)
{
    rReference = minRatioNonFrontalFace;
}
else if (r > maxRatioFrontalFace * 1.1)
{
    rReference = maxRatioFrontalFace;
}

if (width > height)
{
    height = width / rReference;
}
else
{
    width = height * rReference;
}

}
```

### 6.1129.1.2 recommendedImageSizeForDetection()

```
int Digikam::OpenCVDNNFaceDetector::recommendedImageSizeForDetection ( ) [static]
```

Returns the image size (one dimension). recommended for face detection. If the image is considerably larger, it will be rescaled automatically.

## 6.1130 Digikam::OpenCVDNNFaceRecognizer Class Reference

### Classes

- class [Private](#)

### Public Types

- enum [Classifier](#) { [SVM](#) = 0 , [OpenCV\\_KNN](#) , [Tree](#) , [DB](#) }

### Public Member Functions

- void [clearTraining](#) (const QList< int > &idsToClear)
- **OpenCVDNNFaceRecognizer** ([Classifier](#) method, [FaceScanSettings::FaceRecognitionModel](#) recModel)  
*OpenCVDNNFaceRecognizer: Master class to control entire recognition using OpenFace algorithm.*
- QVector< int > [recognize](#) (const QList< QPair< QImage \*, QString > > &inputImages)
- int [recognize](#) (const QPair< QImage \*, QString > &inputImage)
- bool **registerTrainingData** (const cv::Mat &preprocessedImage, int label)  
*register training data for unit test.*
- bool [remove](#) (const QString &hash)
- void [setNbNeighbors](#) (int k)
- void [setThreshold](#) (int threshold)
- void [train](#) (const QList< QPair< QImage \*, QString > > &images, const int label)
- int **verifyTestData** (const cv::Mat &preprocessedImage)  
*predict label of test data for unit test.*

### Static Public Member Functions

- static cv::Mat [prepareForRecognition](#) (const cv::Mat &cvinputImage)
- static cv::Mat [prepareForRecognition](#) (QImage &inputImage)

## 6.1130.1 Member Enumeration Documentation

### 6.1130.1.1 Classifier

```
enum Digikam::OpenCVDNNFaceRecognizer::Classifier
```

## Enumerator

SVM	Support Vector Machines ( <a href="https://docs.opencv.org/4.x/dc/dd6/ml_intro.html#ml_intro_svm">https://docs.opencv.org/4.x/dc/dd6/ml_intro.html#ml_intro_svm</a> )
OpenCV_KNN	K-Nearest Neighbors ( <a href="https://docs.opencv.org/4.x/dc/dd6/ml_intro.html#ml_intro_knn">https://docs.opencv.org/4.x/dc/dd6/ml_intro.html#ml_intro_knn</a> )
Tree	K-Nearest Neighbors Tree ( <a href="https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm">https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm</a> )
DB	Closest Neighbors Tree from the database.

## 6.1130.2 Member Function Documentation

### 6.1130.2.1 clearTraining()

```
void Digikam::OpenCVDNNFaceRecognizer::clearTraining (
    const QList< int > & idsToClear )
```

Clear specified trained data.

### 6.1130.2.2 prepareForRecognition() [1/2]

```
cv::Mat Digikam::OpenCVDNNFaceRecognizer::prepareForRecognition (
    const cv::Mat & cvinputImage ) [static]
```

Returns a cvMat created from the cvinputImage, optimized for recognition.

### 6.1130.2.3 prepareForRecognition() [2/2]

```
cv::Mat Digikam::OpenCVDNNFaceRecognizer::prepareForRecognition (
    QImage & inputImage ) [static]
```

Returns a cvMat created from the inputImage, optimized for recognition.

### 6.1130.2.4 recognize() [1/2]

```
QVector< int > Digikam::OpenCVDNNFaceRecognizer::recognize (
    const QList< QPair< QImage *, QString > > & inputImages )
```

Try to recognize a list of given images. Returns a list of identity ids. If an identity cannot be recognized, returns -1.

### 6.1130.2.5 recognize() [2/2]

```
int Digikam::OpenCVDNNFaceRecognizer::recognize (
    const QPair< QImage *, QString > & inputImage )
```

Try to recognize the given image. Returns the identity id. If the identity cannot be recognized, returns -1. TODO: verify workflow to economize this routine.



### 6.1130.2.6 remove()

```
bool Digikam::OpenCVDNNFaceRecognizer::remove (
    const QString & hash )
```

Returns a cvMat of the extracted features from the cvinputImage, optimized for recognition.

### 6.1130.2.7 setNbNeighbors()

```
void Digikam::OpenCVDNNFaceRecognizer::setNbNeighbors (
    int k )
```

Set K parameter of K-Nearest neighbors algorithm.

### 6.1130.2.8 setThreshold()

```
void Digikam::OpenCVDNNFaceRecognizer::setThreshold (
    int threshold )
```

Set maximum square distance of 2 vectors.

### 6.1130.2.9 train()

```
void Digikam::OpenCVDNNFaceRecognizer::train (
    const QList< QPair< QImage *, QString > > & images,
    const int label )
```

Register faces corresponding to an identity.

## 6.1131 Digikam::OpenCVDNNFaceRecognizer::Private Class Reference

### Classes

- class [ParallelRecognizer](#)
- class [ParallelTrainer](#)

### Public Member Functions

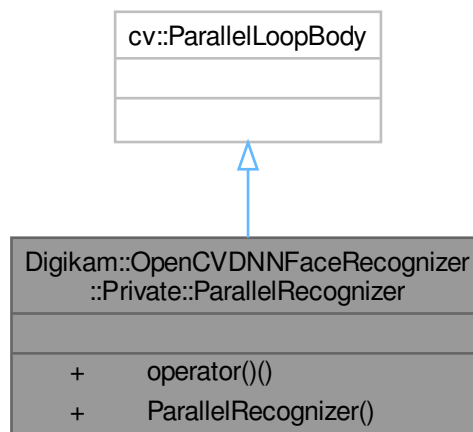
- bool **insertData** (const cv::Mat &position, const int label, const QString &context=QString())
- int **predictDb** (const cv::Mat &faceEmbedding) const
- int **predictKDTree** (const cv::Mat &faceEmbedding) const
- int **predictKNN** (const cv::Mat &faceEmbedding)
- int **predictSFace** (const cv::Mat &faceEmbedding) const
- int **predictSVM** (const cv::Mat &faceEmbedding)
- **Private** ([Classifier](#) mthd, [FaceScanSettings::FaceRecognitionModel](#) recModel)
- bool **trainKNN** ()
- bool **trainSVM** ()

## Public Attributes

- QVector< [DNNFaceExtractorBase](#) \* > **extractors**
- int **kNeighbors** = 5
- cv::Ptr< cv::ml::KNearest > **knn**
- [Classifier](#) **method**
- bool **newDataAdded** = true
- [FaceScanSettings::FaceRecognitionModel](#) **recognizeModel** = [FaceScanSettings::FaceRecognitionModel::SFace](#)
- int **ref** = 1
- cv::Ptr< cv::ml::SVM > **svm**
- [KDTreeBase](#) \* **tree** = nullptr
- int **uiThreshold** = DNN\_MODEL\_THRESHOLD\_NOT\_SET

## 6.1132 Digikam::OpenCVDNNFaceRecognizer::Private::ParallelRecognizer Class Reference

Inheritance diagram for Digikam::OpenCVDNNFaceRecognizer::Private::ParallelRecognizer:

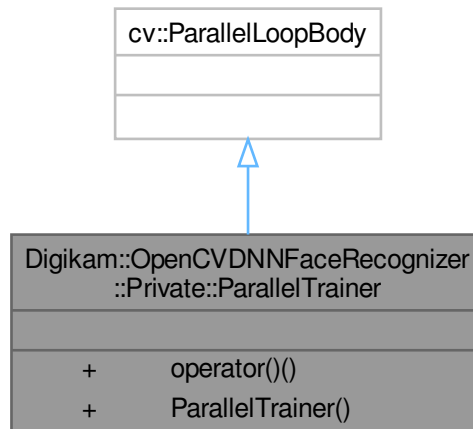


## Public Member Functions

- void **operator()()** (const cv::Range &range) const override
- **ParallelRecognizer** ([OpenCVDNNFaceRecognizer::Private](#) \*const d, const QList< QPair< QImage \*, QString > > &images, QVector< int > &ids)

## 6.1133 Digikam::OpenCVDNNFaceRecognizer::Private::ParallelTrainer Class Reference

Inheritance diagram for Digikam::OpenCVDNNFaceRecognizer::Private::ParallelTrainer:



### Public Member Functions

- void **operator()** (const cv::Range &range) const override
- **ParallelTrainer** ([OpenCVDNNFaceRecognizer::Private](#) \*const d, const QList< QPair< QImage \*, QString > > &images, const int &id)

## 6.1134 Digikam::OpenfacePreprocessor Class Reference

### Public Member Functions

- bool [loadModels](#) ()
- cv::Mat **process** (const cv::Mat &image)

### 6.1134.1 Member Function Documentation

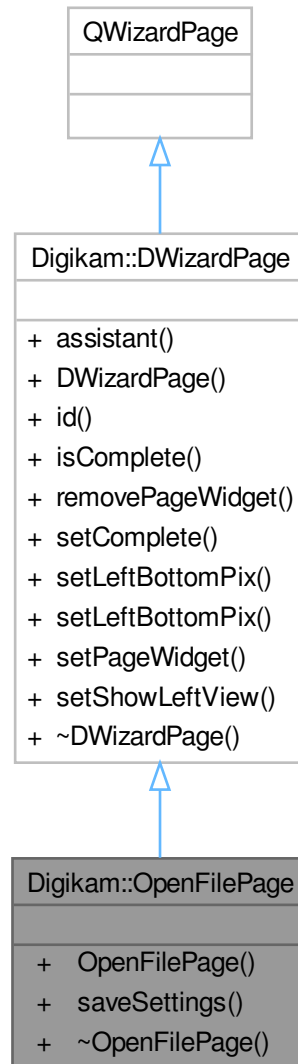
#### 6.1134.1.1 loadModels()

```
bool Digikam::OpenfacePreprocessor::loadModels ( )
```

Load shapepredictor model for face alignment with 68 points of face landmark extraction.

## 6.1135 Digikam::OpenFilePage Class Reference

Inheritance diagram for Digikam::OpenFilePage:



### Public Member Functions

- **OpenFilePage** (QWizard \*const dlg)
- void **saveSettings** ()

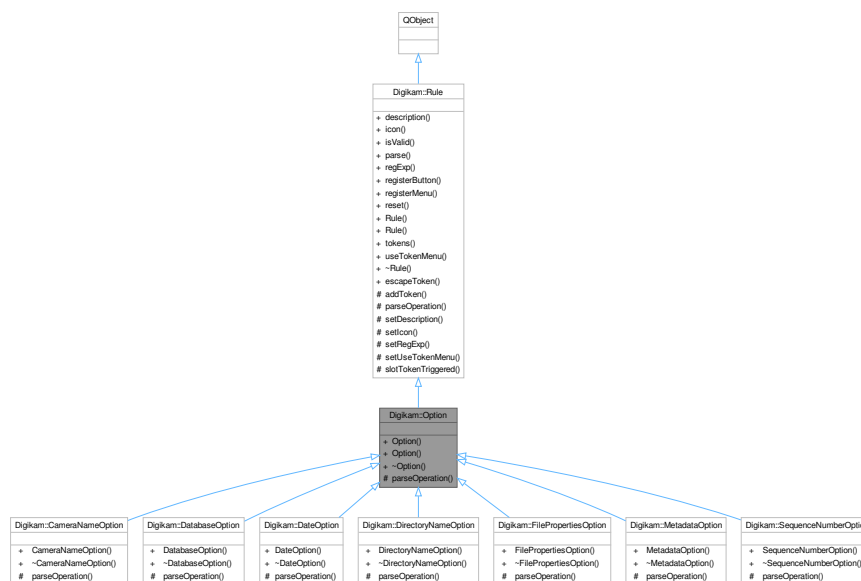
### Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)

- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.1136 Digikam::Option Class Reference

Inheritance diagram for Digikam::Option:



### Public Member Functions

- **Option** (const QString &name, const QString &description)
- **Option** (const QString &name, const QString &description, const QString &icon)

### Public Member Functions inherited from Digikam::Rule

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- ParseResults **parse** (ParseSettings &settings)
- QRegularExpression & **regexp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

### Protected Member Functions

- QString [parseOperation](#) ([ParseSettings](#) &settings, const QRegularExpressionMatch &match) override=0

### Protected Member Functions inherited from [Digikam::Rule](#)

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void [setDescription](#) (const QString &desc)
- void [setIcon](#) (const QString &pixmap)
- void [setRegExp](#) (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

### Additional Inherited Members

### Public Types inherited from [Digikam::Rule](#)

- enum [IconType](#) { [Action](#) = 0 , [Dialog](#) }

### Signals inherited from [Digikam::Rule](#)

- void [signalTokenTriggered](#) (const QString &)

### Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString [escapeToken](#) (const QString &token)

### Protected Slots inherited from [Digikam::Rule](#)

- virtual void [slotTokenTriggered](#) (const QString &)

## 6.1136.1 Member Function Documentation

### 6.1136.1.1 [parseOperation\(\)](#)

```
QString Digikam::Option::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [pure virtual]
```

TODO: describe me

#### Parameters

<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

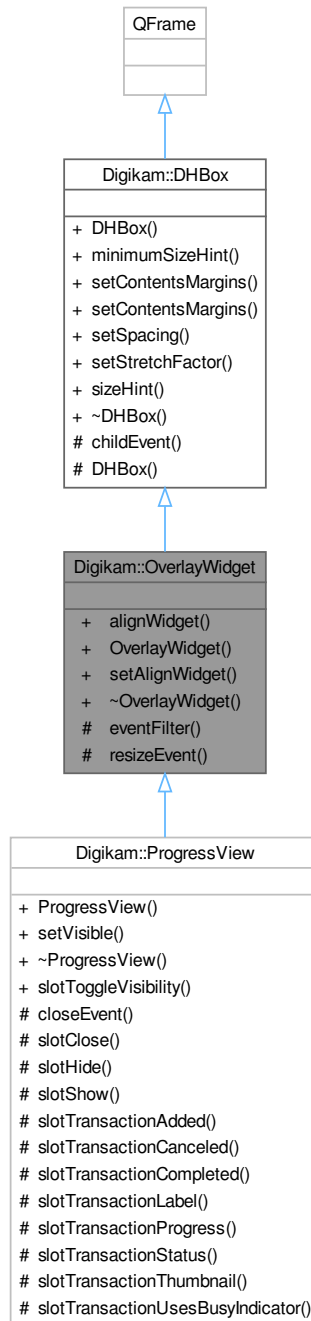
Returns

Implements [Digikam::Rule](#).

Implemented in [Digikam::CameraNameOption](#), [Digikam::DatabaseOption](#), [Digikam::DateOption](#), [Digikam::DirectoryNameOption](#), [Digikam::FilePropertiesOption](#), [Digikam::MetadataOption](#), and [Digikam::SequenceNumberOption](#).

## 6.1137 Digikam::OverlayWidget Class Reference

Inheritance diagram for Digikam::OverlayWidget:



### Public Member Functions

- `QWidget * alignWidget ()` const
- **OverlayWidget** (`QWidget *const alignWidget`, `QWidget *const parent`, `const QString &name=QString()`)
- void **setAlignWidget** (`QWidget *const alignWidget`)



**Public Member Functions inherited from [Digikam::DHBox](#)**

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

**Protected Member Functions**

- bool **eventFilter** (QObject \*o, QEvent \*e) override
- void **resizeEvent** (QResizeEvent \*ev) override

**Protected Member Functions inherited from [Digikam::DHBox](#)**

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

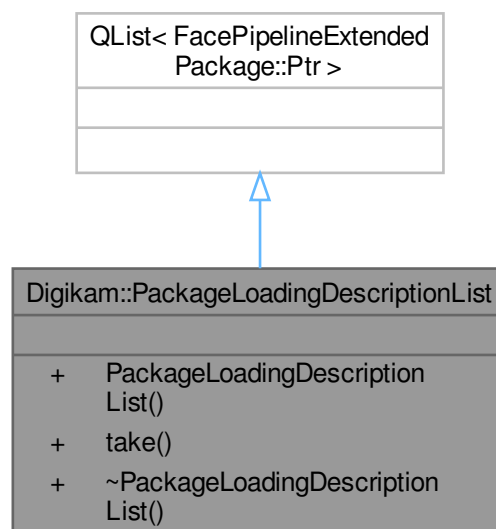
**6.1137.1 Detailed Description**

This is a widget that can align itself with another one, without using a layout, so that it can actually be on top of other widgets. Currently the only supported type of alignment is "right aligned, on top of the other widget".

[OverlayWidget](#) inherits [DHBox](#) for convenience purposes (layout, and frame)

**6.1138 Digikam::PackageLoadingDescriptionList Class Reference**

Inheritance diagram for Digikam::PackageLoadingDescriptionList:



### Public Member Functions

- FacePipelineExtendedPackage::Ptr **take** (const [LoadingDescription](#) &description)

## 6.1139 Digikam::PageItem Class Reference

### Public Member Functions

- void **appendChild** ([PageItem](#) \*const child)
- [PageItem](#) \* **child** (int row)
- int **childCount** () const
- int **columnCount** () const
- void **dump** (int indent=0)
- [PageItem](#) \* **findChild** (const [DConfigDlgWdgItem](#) \*item)
- void **insertChild** (int row, [PageItem](#) \*const child)
- [PageItem](#) ([DConfigDlgWdgItem](#) \*const pageItem, [PageItem](#) \*const parent=nullptr)
- [DConfigDlgWdgItem](#) \* **pageWidgetItem** () const
- [PageItem](#) \* **parent** () const
- void **removeChild** (int row)
- int **row** () const

## 6.1140 Digikam::PAIbum Class Reference

Inheritance diagram for Digikam::PAIbum:



### Public Member Functions

- `QString albumPath () const`
- `int albumRootId () const`

- QString **albumRootLabel** () const
- QString **albumRootPath** () const
- QString **caption** () const
- QString **category** () const
- [CoreDbUrl databaseUrl](#) () const override
- QDate **date** () const
- QUrl **fileUrl** () const
- QString **folderPath** () const
- qlonglong **iconId** () const
- bool **isAlbumRoot** () const
- **PAAlbum** (const QString &parentPath, int albumRoot)  
*Constructor for Trash album.*
- **PAAlbum** (const QString &title)  
*Constructor for root album.*
- **PAAlbum** (int albumRoot, const QString &label)  
*Constructor for album root albums.*
- **PAAlbum** (int albumRoot, const QString &parentPath, const QString &title, int id)  
*Constructor for normal albums.*
- QString **prettyUrl** () const
- void **setCaption** (const QString &caption)
- void **setCategory** (const QString &category)
- void **setDate** (const QDate &date)

## Public Member Functions inherited from [Digikam::Album](#)

- QList< int > [childAlbumIds](#) (bool recursive=false)
- AlbumList [childAlbums](#) (bool recursive=false)
- [Album](#) \* [childAtRow](#) (int row) const
- int [childCount](#) () const
- void \* [extraData](#) (const void \*const key) const
- [Album](#) \* [firstChild](#) () const
- int [globalID](#) () const
- int [id](#) () const
- bool [isAncestorOf](#) ([Album](#) \*const album) const
- bool [isRoot](#) () const
- bool [isTrashAlbum](#) () const
- bool [isUsedByLabelsTree](#) () const
- [Album](#) \* [lastChild](#) () const
- [Album](#) \* [next](#) () const
- [Album](#) \* [parent](#) () const
- void [prepareForDeletion](#) ()
- [Album](#) \* [prev](#) () const
- void [removeExtraData](#) (const void \*const key)
- int [rowFromAlbum](#) () const
- void [setExtraData](#) (const void \*const key, void \*const value)
- void [setUsedByLabelsTree](#) (bool isUsed)
- QString [title](#) () const
- [Type](#) [type](#) () const

## Friends

- class [AlbumManager](#)

## Additional Inherited Members

## Public Types inherited from [Digikam::Album](#)

- enum [Type](#) {  
    [PHYSICAL](#) = 0 , [TAG](#) , [DATE](#) , [SEARCH](#) ,  
    [FACE](#) }

## Static Public Member Functions inherited from [Digikam::Album](#)

- static int [globalID](#) ([Type](#) type, int id)  
    *Produces the global id.*

## Protected Member Functions inherited from [Digikam::Album](#)

- [Album](#) ([Album::Type](#) type, int id, bool root)
- void [clear](#) ()
- void [insertChild](#) ([Album](#) \*const child)
- void [removeChild](#) ([Album](#) \*const child)
- void [setParent](#) ([Album](#) \*const [parent](#))
- void [setTitle](#) (const QString &[title](#))
- virtual [~Album](#) ()

### 6.1140.1 Detailed Description

A Physical [Album](#) representation

### 6.1140.2 Member Function Documentation

#### 6.1140.2.1 [databaseUrl\(\)](#)

`CoreDbUrl` [Digikam::PAlbum::databaseUrl](#) ( ) const [override], [virtual]

#### Returns

the kde url of the album

Implements [Digikam::Album](#).

## 6.1141 [Digikam::PAlbumPath](#) Class Reference

### Public Member Functions

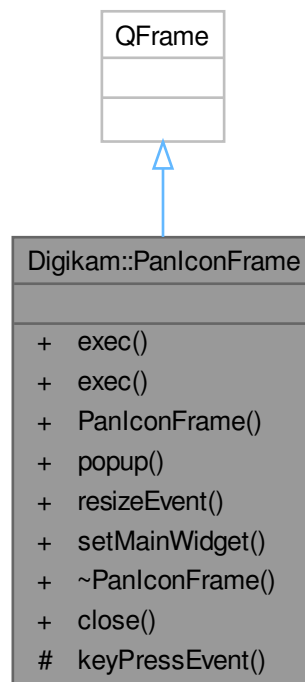
- bool [operator==](#) (const [PAlbumPath](#) &other) const
- [PAlbumPath](#) (const [PAlbum](#) \*const album)
- [PAlbumPath](#) (int albumRootId, const QString &albumPath)

**Public Attributes**

- QString **albumPath**
- int **albumRootId** = -1

**6.1142 Digikam::PanIconFrame Class Reference**

Inheritance diagram for Digikam::PanIconFrame:

**Public Slots**

- void `close` (int r)

**Signals**

- void `leaveModality` ()

**Public Member Functions**

- int `exec` (const QPoint &pos)
- int `exec` (int x, int y)
- **PanIconFrame** (QWidget \*const parent=nullptr)
- void `popup` (const QPoint &pos)
- void `resizeEvent` (QResizeEvent \*resize) override
- void `setMainWidget` (QWidget \*const main)

## Protected Member Functions

- void [keyPressEvent](#) (QKeyEvent \*e) override

## Friends

- class **Private**

## 6.1142.1 Detailed Description

Frame with popup menu behavior to host [PanIconWidget](#).

## 6.1142.2 Member Function Documentation

### 6.1142.2.1 close

```
void Digikam::PanIconFrame::close (  
    int r ) [slot]
```

Close the popup window. This is called from the main widget, usually. `r` is the result returned from [exec\(\)](#).

### 6.1142.2.2 exec() [1/2]

```
int Digikam::PanIconFrame::exec (  
    const QPoint & pos )
```

Execute the popup window.

### 6.1142.2.3 exec() [2/2]

```
int Digikam::PanIconFrame::exec (  
    int x,  
    int y )
```

Execute the popup window.

### 6.1142.2.4 keyPressEvent()

```
void Digikam::PanIconFrame::keyPressEvent (  
    QKeyEvent * e ) [override], [protected]
```

Catch key press events.

#### 6.1142.2.5 popup()

```
void Digikam::PanIconFrame::popup (
    const QPoint & pos )
```

Open the popup window at position pos.

#### 6.1142.2.6 resizeEvent()

```
void Digikam::PanIconFrame::resizeEvent (
    QResizeEvent * resize ) [override]
```

The resize event. Simply resizes the main widget to the whole widgets client size.

#### 6.1142.2.7 setMainWidget()

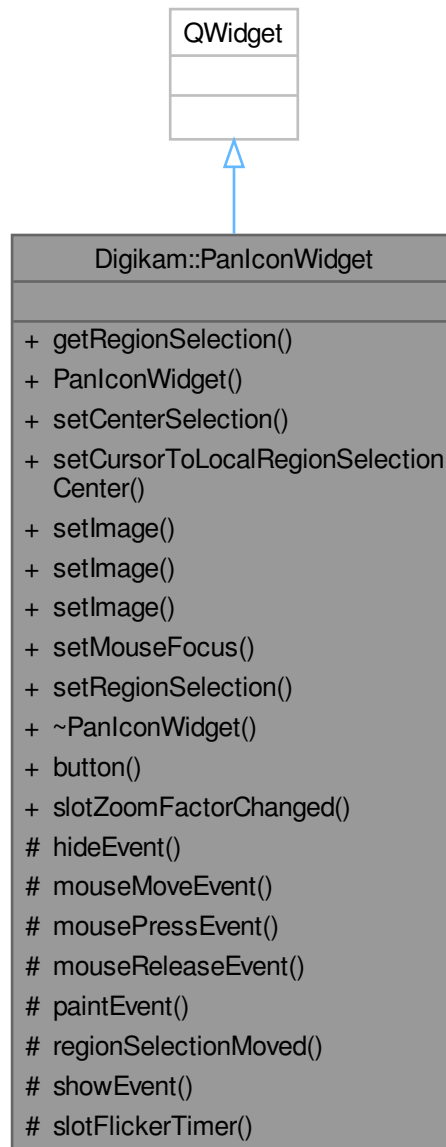
```
void Digikam::PanIconFrame::setMainWidget (
    QWidget *const main )
```

Set the main widget. You cannot set the main widget from the constructor, since it must be a child of the frame itselfes. Be careful: the size is set to the main widgets size. It is up to you to set the main widgets correct size before setting it as the main widget.



## 6.1143 Digikam::PanIconWidget Class Reference

Inheritance diagram for Digikam::PanIconWidget:



### Public Slots

- void **slotZoomFactorChanged** (double)

### Signals

- void **signalHidden** ()
- void **signalSelectionMoved** (const QRect &rect, bool targetDone)
- void **signalSelectionTakeFocus** ()

## Public Member Functions

- QRect **getRegionSelection** () const
- **PanIconWidget** (QWidget \*const parent=nullptr)
- void **setCenterSelection** ()
- void **setCursorToLocalRegionSelectionCenter** ()
- void **setImage** (const QImage &scaledPreviewImage, const QSize &fullImageSize)
- void **setImage** (int previewWidth, int previewHeight, const QImage &fullOriginalImage)
- void **setImage** (int previewWidth, int previewHeight, const QImage &fullOriginalImage)
- void **setMouseFocus** ()
- void **setRegionSelection** (const QRect &regionSelection)

## Static Public Member Functions

- static QPushButton \* **button** ()

## Protected Slots

- void **slotFlickerTimer** ()

## Protected Member Functions

- void **hideEvent** (QHideEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **regionSelectionMoved** (bool targetDone)
- void **showEvent** (QShowEvent \*) override

## 6.1143.1 Member Function Documentation

### 6.1143.1.1 regionSelectionMoved()

```
void Digikam::PanIconWidget::regionSelectionMoved (
    bool targetDone ) [protected]
```

Recalculate the target selection position and emit 'signalSelectionMoved'.

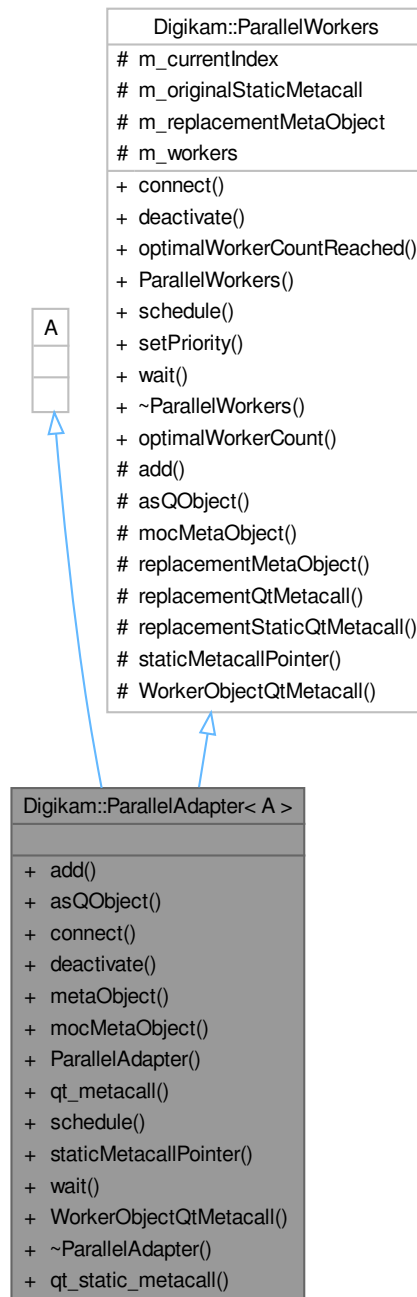
### 6.1143.1.2 signalSelectionMoved

```
void Digikam::PanIconWidget::signalSelectionMoved (
    const QRect & rect,
    bool targetDone ) [signal]
```

Emitted when selection have been moved with mouse. 'targetDone' boolean value is used for indicate if the mouse have been released.

## 6.1144 Digikam::ParallelAdapter< A > Class Template Reference

Inheritance diagram for Digikam::ParallelAdapter< A >:



### Public Member Functions

- void **add** (A \*const worker)
- QObject \* [asQObject](#) () override

- bool **connect** (const char \*signal, const QObject \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection) const
- void **deactivate** ([WorkerObject::DeactivatingMode](#) mode=[WorkerObject::FlushSignals](#))
- const QMetaObject \* **metaObject** () const override
- const QMetaObject \* **mocMetaObject** () const override
- [ParallelAdapter](#) ()=default
- int **qt\_metacall** (QMetaObject::Call \_c, int \_id, void \*\*\_a) override
- void **schedule** ()
- StaticMetacallFunction [staticMetacallPointer](#) () override
- void **wait** ()
- int [WorkerObjectQtMetacall](#) (QMetaObject::Call \_c, int \_id, void \*\*\_a) override

### Public Member Functions inherited from [Digikam::ParallelWorkers](#)

- bool **connect** (const char \*signal, const QObject \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection) const  
*Connects signals outbound from all workers to a given receiver.*
- void **deactivate** ([WorkerObject::DeactivatingMode](#) mode=[WorkerObject::FlushSignals](#))
- bool [optimalWorkerCountReached](#) () const
- [ParallelWorkers](#) ()=default
- void [schedule](#) ()
- void **setPriority** (QThread::Priority priority)
- void **wait** ()

### Static Public Member Functions

- static void **qt\_static\_metacall** (QObject \*o, QMetaObject::Call \_c, int \_id, void \*\*\_a)

### Static Public Member Functions inherited from [Digikam::ParallelWorkers](#)

- static int [optimalWorkerCount](#) ()

### Additional Inherited Members

### Protected Types inherited from [Digikam::ParallelWorkers](#)

- typedef void(\* **StaticMetacallFunction**) (QObject \*, QMetaObject::Call, int, void \*\*)

### Protected Member Functions inherited from [Digikam::ParallelWorkers](#)

- void **add** ([WorkerObject](#) \*const worker)
- const QMetaObject \* **replacementMetaObject** () const
- int [replacementQtMetacall](#) (QMetaObject::Call \_c, int \_id, void \*\*\_a)
- int [replacementStaticQtMetacall](#) (QMetaObject::Call \_c, int \_id, void \*\*\_a)

## Protected Attributes inherited from [Digikam::ParallelWorkers](#)

- int `m_currentIndex` = 0
- StaticMetacallFunction `m_originalStaticMetacall` = nullptr
- QMetaObject \* `m_replacementMetaObject` = nullptr
- QList< [WorkerObject](#) \* > `m_workers`

### 6.1144.1 Constructor & Destructor Documentation

#### 6.1144.1.1 ParallelAdapter()

```
template<class A >
Digikam::ParallelAdapter< A >::ParallelAdapter ( ) [default]
```

Instead of using a single [WorkerObject](#), create a [ParallelAdapter](#) for your worker object subclass, and add() individual WorkerObjects. The load will be evenly distributed. Note: unlike with [WorkerObject](#) directly, there is no need to call schedule(). For inbound connections (signals connected to a [WorkerObject](#)'s slot, to be processed, use a Qt::DirectConnection on the adapter. For outbound connections (signals emitted from the [WorkerObject](#)), use [ParallelAdapter](#)'s connect to have a connection from all added WorkerObjects.

### 6.1144.2 Member Function Documentation

#### 6.1144.2.1 asQObject()

```
template<class A >
QObject * Digikam::ParallelAdapter< A >::asQObject ( ) [inline], [override], [virtual]
```

Return the target QObject (double inheritance)

Implements [Digikam::ParallelWorkers](#).

#### 6.1144.2.2 mocMetaObject()

```
template<class A >
const QMetaObject * Digikam::ParallelAdapter< A >::mocMetaObject ( ) const [inline], [override], [virtual]
```

The moc-generated metaObject of the target object

Implements [Digikam::ParallelWorkers](#).

#### 6.1144.2.3 staticMetacallPointer()

```
template<class A >
StaticMetacallFunction Digikam::ParallelAdapter< A >::staticMetacallPointer ( ) [inline], [override], [virtual]
```

Implements [Digikam::ParallelWorkers](#).

#### 6.1144.2.4 WorkerObjectQtMetacall()

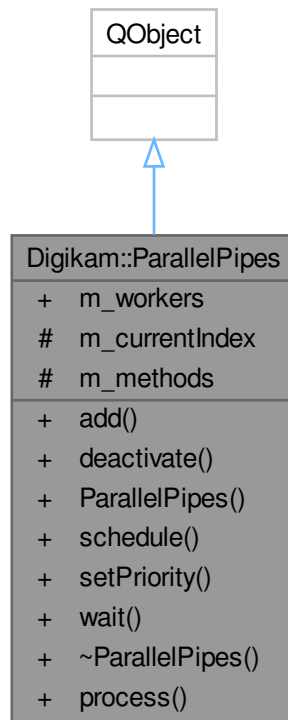
```
template<class A >
int Digikam::ParallelAdapter< A >::WorkerObjectQtMetacall (
    QMetaObject::Call _c,
    int _id,
    void ** _a ) [inline], [override], [virtual]
```

The qt\_metacall of [WorkerObject](#), one level above the target QObject

Implements [Digikam::ParallelWorkers](#).

### 6.1145 Digikam::ParallelPipes Class Reference

Inheritance diagram for Digikam::ParallelPipes:



#### Public Slots

- void **process** (const FacePipelineExtendedPackage::Ptr &package)

#### Signals

- void **processed** (const FacePipelineExtendedPackage::Ptr &package)

### Public Member Functions

- void **add** ([WorkerObject](#) \*const worker)
- void **deactivate** ([WorkerObject::DeactivatingMode](#) mode=[WorkerObject::FlushSignals](#))
- void **schedule** ()
- void **setPriority** ([QThread::Priority](#) priority)
- void **wait** ()

### Public Attributes

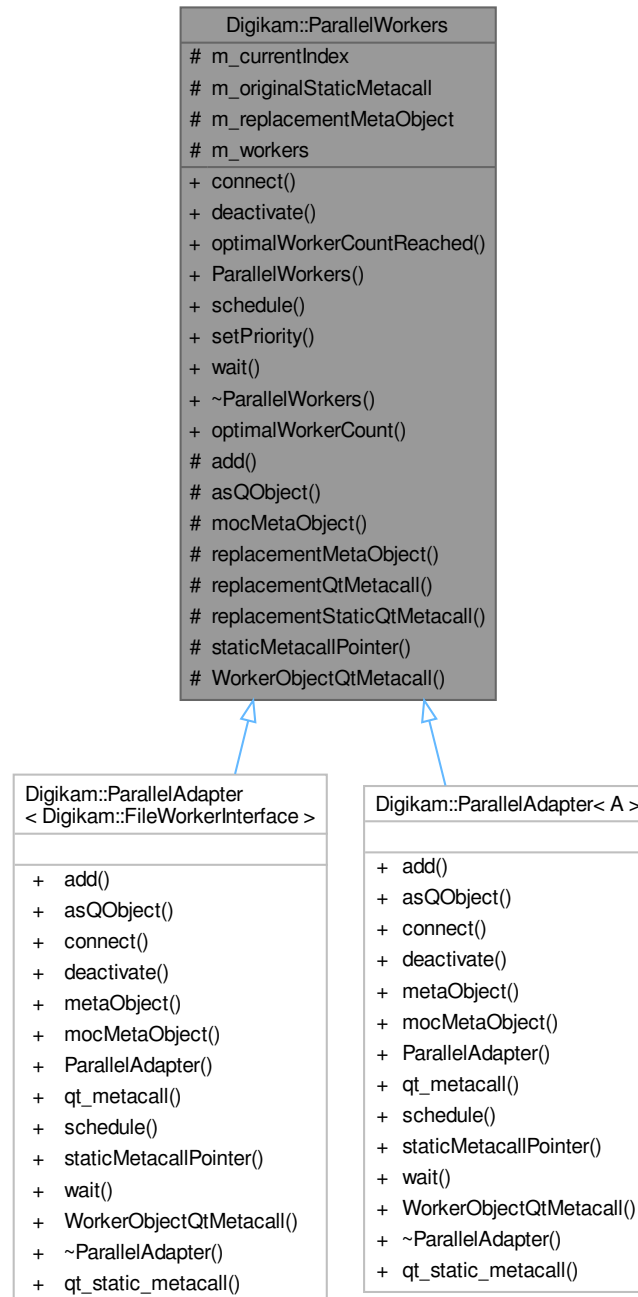
- [QList](#)< [WorkerObject](#) \* > **m\_workers**

### Protected Attributes

- int **m\_currentIndex** = 0
- [QList](#)< [QMetaMethod](#) > **m\_methods**

## 6.1146 Digikam::ParallelWorkers Class Reference

Inheritance diagram for Digikam::ParallelWorkers:



### Public Member Functions

- bool **connect** (const char \*signal, const QObject \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection) const



*Connects signals outbound from all workers to a given receiver.*

- void **deactivate** ([WorkerObject::DeactivatingMode](#) mode=[WorkerObject::FlushSignals](#))
- bool [optimalWorkerCountReached](#) () const
- [ParallelWorkers](#) ()=default
- void [schedule](#) ()
- void **setPriority** (QThread::Priority priority)
- void **wait** ()

### Static Public Member Functions

- static int [optimalWorkerCount](#) ()

### Protected Types

- typedef void(\* **StaticMetacallFunction**) (QObject \*, QMetaObject::Call, int, void \*\*)

### Protected Member Functions

- void **add** ([WorkerObject](#) \*const worker)
- virtual QObject \* [asQObject](#) ()=0
- virtual const QMetaObject \* [mocMetaObject](#) () const =0
- const QMetaObject \* **replacementMetaObject** () const
- int [replacementQtMetacall](#) (QMetaObject::Call \_c, int \_id, void \*\*\_a)
- int **replacementStaticQtMetacall** (QMetaObject::Call \_c, int \_id, void \*\*\_a)
- virtual StaticMetacallFunction **staticMetacallPointer** ()=0
- virtual int [WorkerObjectQtMetacall](#) (QMetaObject::Call \_c, int \_id, void \*\*\_a)=0

### Protected Attributes

- int **m\_currentIndex** = 0
- StaticMetacallFunction **m\_originalStaticMetacall** = nullptr
- QMetaObject \* **m\_replacementMetaObject** = nullptr
- QList< [WorkerObject](#) \* > **m\_workers**

## 6.1146.1 Constructor & Destructor Documentation

### 6.1146.1.1 ParallelWorkers()

```
Digikam::ParallelWorkers::ParallelWorkers ( ) [default]
```

[ParallelWorkers](#) is a helper class to distribute work over several identical workers objects. See [ParallelAdapter](#) for guidance how to use it.

## 6.1146.2 Member Function Documentation

### 6.1146.2.1 asQObject()

```
virtual QObject * Digikam::ParallelWorkers::asQObject ( ) [protected], [pure virtual]
```

Return the target QObject (double inheritance)

Implemented in [Digikam::ParallelAdapter< A >](#), and [Digikam::ParallelAdapter< Digikam::FileWorkerInterface >](#).

### 6.1146.2.2 mocMetaObject()

```
virtual const QMetaObject * Digikam::ParallelWorkers::mocMetaObject ( ) const [protected],  
[pure virtual]
```

The moc-generated metaObject of the target object

Implemented in [Digikam::ParallelAdapter< A >](#), and [Digikam::ParallelAdapter< Digikam::FileWorkerInterface >](#).

### 6.1146.2.3 optimalWorkerCount()

```
int Digikam::ParallelWorkers::optimalWorkerCount ( ) [static]
```

Regarding the number of logical CPUs on the current machine, returns the optimal count of concurrent workers

### 6.1146.2.4 optimalWorkerCountReached()

```
bool Digikam::ParallelWorkers::optimalWorkerCountReached ( ) const
```

Returns true if the current number of added workers has reached the [optimalWorkerCount\(\)](#)

### 6.1146.2.5 replacementQtMetacall()

```
int Digikam::ParallelWorkers::replacementQtMetacall (  
    QMetaObject::Call _c,  
    int _id,  
    void ** _a ) [protected]
```

Replaces slot call distribution of the target QObject

### 6.1146.2.6 schedule()

```
void Digikam::ParallelWorkers::schedule ( )
```

The corresponding methods of all added worker objects will be called

### 6.1146.2.7 WorkerObjectQtMetacall()

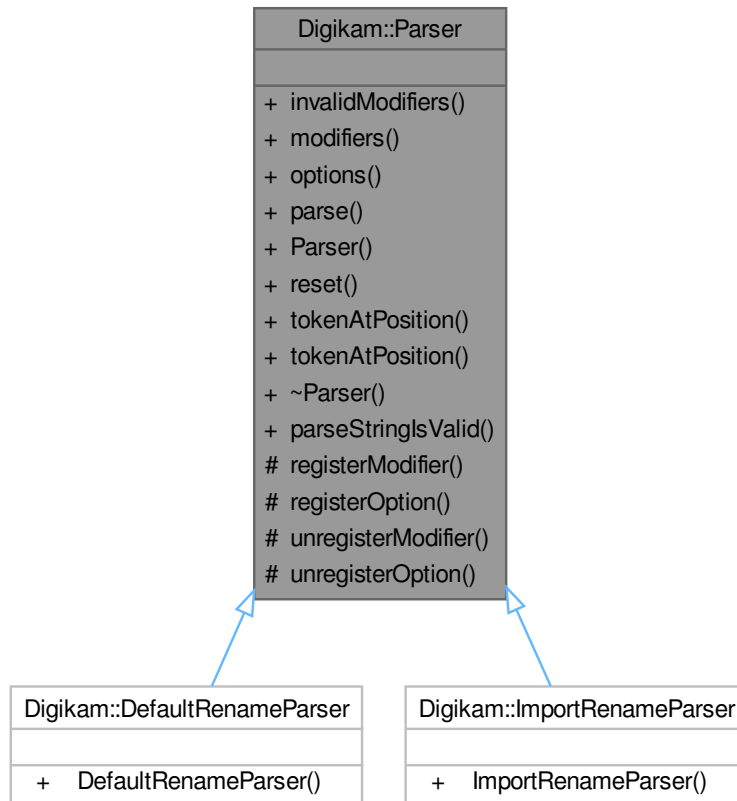
```
virtual int Digikam::ParallelWorkers::WorkerObjectQtMetacall (  
    QMetaObject::Call _c,  
    int _id,  
    void ** _a ) [protected], [pure virtual]
```

The qt\_metacall of [WorkerObject](#), one level above the target QObject

Implemented in [Digikam::ParallelAdapter< A >](#), and [Digikam::ParallelAdapter< Digikam::FileWorkerInterface >](#).

## 6.1147 Digikam::Parser Class Reference

Inheritance diagram for Digikam::Parser:



### Public Member Functions

- `ParseResults` **invalidModifiers** (`ParseSettings` &settings)
- `RulesList` **modifiers** () const
- `RulesList` **options** () const
- `QString` **parse** (`ParseSettings` &settings)
- void **reset** ()
- bool **tokenAtPosition** (`ParseSettings` &settings, int pos)
- bool **tokenAtPosition** (`ParseSettings` &settings, int pos, int &start, int &length)

### Static Public Member Functions

- static bool **parseStringlsValid** (const `QString` &str)

### Protected Member Functions

- void **registerModifier** (`Rule` \*modifier)
- void **registerOption** (`Rule` \*option)
- void **unregisterModifier** (const `Rule` \*modifier)
- void **unregisterOption** (const `Rule` \*option)

## 6.1147.1 Member Function Documentation

### 6.1147.1.1 parseStringIsValid()

```
bool Digikam::Parser::parseStringIsValid (
    const QString & str ) [static]
```

check if the given parse string is valid

#### Parameters

<i>str</i>	the parse string
------------	------------------

#### Returns

true if valid / can be parsed

## 6.1148 Digikam::ParseResults Class Reference

### Public Types

- typedef QPair< int, int > **ResultsKey**
- typedef QMultiMap< ResultsKey, ResultsValue > **ResultsMap**
- typedef QPair< QString, QString > **ResultsValue**

### Public Member Functions

- void **addEntry** (const ResultsKey &key, const ResultsValue &value)
- void **append** (const ParseResults &results)
- void **clear** ()
- void **debug** () const
- void **deleteEntry** (const ResultsKey &key)
- bool **hasKey** (const ResultsKey &key)
- bool **hasKeyAtApproximatePosition** (int pos) const
- bool **hasKeyAtPosition** (int pos) const
- bool **isEmpty** () const
- ResultsKey **keyAtApproximatePosition** (int pos) const
- ResultsKey **keyAtPosition** (int pos) const
- QList< ResultsKey > **keys** () const
- int **offset** (const ResultsKey &key) const
- QString **replaceTokens** (const QString &markedString) const
- QString **result** (const ResultsKey &key) const
- QString **resultValuesAsString** () const
- QString **token** (const ResultsKey &key) const
- QList< ResultsValue > **values** () const

## 6.1149 Digikam::ParseSettings Class Reference

### Public Member Functions

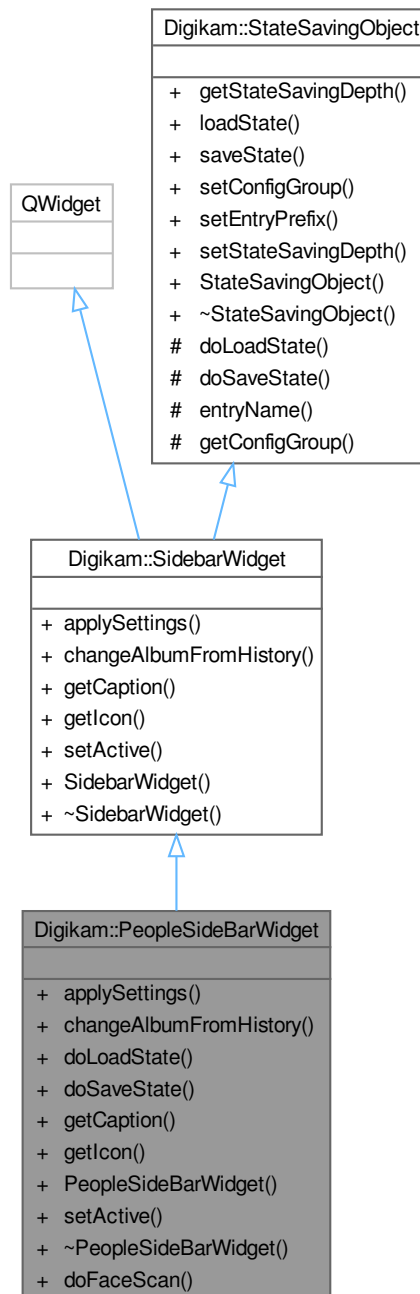
- bool **isValid** () const
- **ParseSettings** ()  
*default constructor*
- **ParseSettings** (const [ItemInfo](#) &info)  
*ItemInfo constructor.*
- **ParseSettings** (const QString &\_parseString)
- **ParseSettings** (const QString &\_parseString, const [ItemInfo](#) &info)

### Public Attributes

- QDateTime **creationTime**
- ParseResults::ResultsKey **currentResultsKey**
- int **cutFileName** = 0
- QUrl **fileUrl**
- [ParseResults](#) **invalidModifiers**
- [AdvancedRenameManager](#) \* **manager** = nullptr
- QString **parseString**
- [ParseResults](#) **results**
- int **startIndex** = 1
- QString **str2Modify**
- bool **useOriginalFileExtension** = true

## 6.1150 Digikam::PeopleSideBarWidget Class Reference

Inheritance diagram for Digikam::PeopleSideBarWidget:



### Signals

- void **requestFaceMode** (bool on)
- void **signalFindDuplicates** (const QList< [TAlbum](#) \* > &albums)

## Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

## Public Member Functions

- void [applySettings](#) () override
- void [changeAlbumFromHistory](#) (const QList< [Album](#) \* > &album) override
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- **PeopleSideBarWidget** (QWidget \*const parent, [TagModel](#) \*const tagModel, [SearchModificationHelper](#) \*const searchModificationHelper)
- void [setActive](#) (bool active) override

## Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Static Public Member Functions

- static void [doFaceScan](#) (const [FaceScanSettings](#) &faceScanSettings)

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.1150.1 Member Function Documentation

### 6.1150.1.1 applySettings()

```
void Digikam::PeopleSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).

### 6.1150.1.2 changeAlbumFromHistory()

```
void Digikam::PeopleSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.1150.1.3 doLoadState()

```
void Digikam::PeopleSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1150.1.4 doSaveState()

```
void Digikam::PeopleSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1150.1.5 getCaption()

```
const QString Digikam::PeopleSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).



### 6.1150.1.6 `getIcon()`

```
const QIcon Digikam::PeopleSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.1150.1.7 `setActive()`

```
void Digikam::PeopleSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

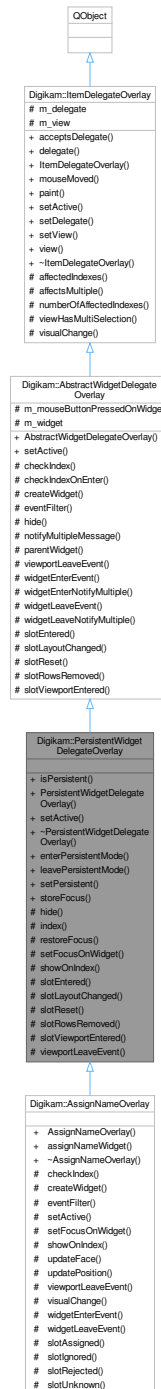
## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.1151 Digikam::PersistentWidgetDelegateOverlay Class Reference

Inheritance diagram for Digikam::PersistentWidgetDelegateOverlay:



**Public Slots**

- void **enterPersistentMode** ()
- void **leavePersistentMode** ()
- void **setPersistent** (bool persistent)
- void **storeFocus** ()

**Public Member Functions**

- bool **isPersistent** () const
- **PersistentWidgetDelegateOverlay** (QObject \*const parent)
- void **setActive** (bool active) override

**Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

**Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)**

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

**Protected Member Functions**

- void **hide** () override
- QModelIndex **index** () const
- void **restoreFocus** ()
- virtual void **setFocusOnWidget** ()
- virtual void **showOnIndex** (const QModelIndex &index)  
*see [slotEntered\(\)](#)*
- void **slotEntered** (const QModelIndex &index) override
- void **slotLayoutChanged** () override
- void **slotReset** () override
- void **slotRowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **slotViewportEntered** () override
- void **viewportLeaveEvent** (QObject \*obj, QEvent \*event) override

**Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)**

- virtual bool **checkIndex** (const QModelIndex &index) const
- bool **checkIndexOnEnter** (const QModelIndex &index) const
- virtual QWidget \* **createWidget** ()=0
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* **parentWidget** () const
- virtual void **widgetEnterEvent** ()
- void **widgetEnterNotifyMultiple** (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- `QList< QModelIndex > affectedIndexes` (const QModelIndex &index) const
- `bool affectsMultiple` (const QModelIndex &index) const
- `int numberOfAffectedIndexes` (const QModelIndex &index) const
- `bool viewHasMultiSelection` () const

## Additional Inherited Members

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- `void hideNotification` ()
- `void requestNotification` (const QModelIndex &index, const QString &message)
- `void update` (const QModelIndex &index)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

- virtual `void visualChange` ()

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- `bool m_mouseButtonPressedOnWidget` = false
- `QWidget * m_widget` = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- `QAbstractItemDelegate * m_delegate` = nullptr
- `QAbstractItemView * m_view` = nullptr

## 6.1151.1 Constructor & Destructor Documentation

### 6.1151.1.1 PersistentWidgetDelegateOverlay()

```
Digikam::PersistentWidgetDelegateOverlay::PersistentWidgetDelegateOverlay (
    QObject *const parent ) [explicit]
```

This class offers additional / modified behavior: When a "persistent" mode is entered, it will not move by mouse hover, but stay and only move on mouse click. If the overlay widget had focus, it will be restored on show.

## 6.1151.2 Member Function Documentation

### 6.1151.2.1 hide()

```
void Digikam::PersistentWidgetDelegateOverlay::hide ( ) [override], [protected], [virtual]
```

Called when the widget shall be hidden (mouse cursor left index, viewport, uninstalled etc.). Default implementation [hide\(\)](#)s `m_widget`.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1151.2.2 setActive()

```
void Digikam::PersistentWidgetDelegateOverlay::setActive (
    bool active ) [override], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1151.2.3 setFocusOnWidget()

```
void Digikam::PersistentWidgetDelegateOverlay::setFocusOnWidget ( ) [protected], [virtual]
```

Reimplement to set the focus on the correct subwidget. Default implementation sets focus on widget()

Reimplemented in [Digikam::AssignNameOverlay](#).

### 6.1151.2.4 setPersistent

```
void Digikam::PersistentWidgetDelegateOverlay::setPersistent (
    bool persistent ) [slot]
```

Enters persistent mode. The overlay is moved because of mouse hover.

### 6.1151.2.5 showOnIndex()

```
void Digikam::PersistentWidgetDelegateOverlay::showOnIndex (
    const QModelIndex & index ) [protected], [virtual]
```

Reimplemented in [Digikam::AssignNameOverlay](#).

### 6.1151.2.6 slotEntered()

```
void Digikam::PersistentWidgetDelegateOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Most overlays reimplement this slot to get the starting point for repositioning a widget etc. This class instead provides [showOnIndex\(\)](#) which you shall use for this purpose.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1151.2.7 slotLayoutChanged()

```
void Digikam::PersistentWidgetDelegateOverlay::slotLayoutChanged ( ) [override], [protected],
[virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1151.2.8 slotReset()

```
void Digikam::PersistentWidgetDelegateOverlay::slotReset ( ) [override], [protected], [virtual]
```

Default implementations of these three slots call [hide\(\)](#)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1151.2.9 slotRowsRemoved()

```
void Digikam::PersistentWidgetDelegateOverlay::slotRowsRemoved (
    const QModelIndex & parent,
    int start,
    int end ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1151.2.10 slotViewportEntered()

```
void Digikam::PersistentWidgetDelegateOverlay::slotViewportEntered ( ) [override], [protected],
[virtual]
```

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1151.2.11 viewportLeaveEvent()

```
void Digikam::PersistentWidgetDelegateOverlay::viewportLeaveEvent (
    QObject * obj,
    QEvent * event ) [override], [protected], [virtual]
```

Called when a `QEvent::Leave` of the viewport is received. The default implementation [hide\(\)](#)s.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

## 6.1152 Digikam::PhotoInfoContainer Class Reference

### Public Member Functions

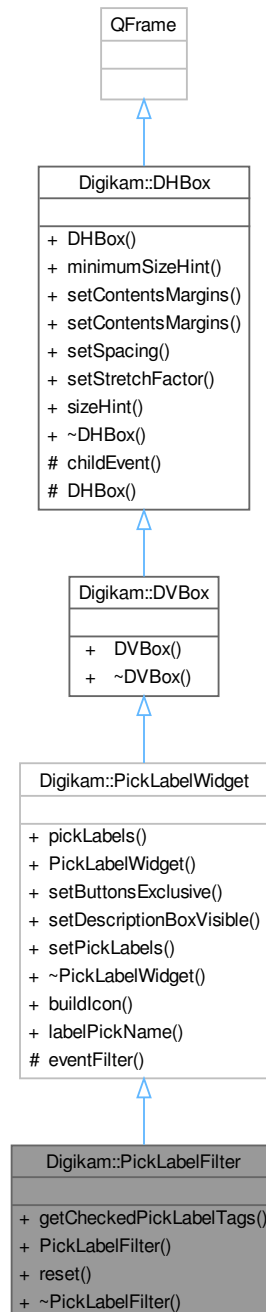
- `bool isEmpty () const`
- `bool isNull () const`
- `PhotoInfoContainer & operator= (const PhotoInfoContainer &)=default`
- `PhotoInfoContainer & operator= (PhotoInfoContainer &&)=default`
- `bool operator== (const PhotoInfoContainer &t) const`
- `PhotoInfoContainer (const PhotoInfoContainer &)=default`

### Public Attributes

- QString **aperture**
- QDateTime **dateTime**
- QString **exposureMode**
- QString **exposureProgram**
- QString **exposureTime**
- QString **flash**
- QString **focalLength**
- QString **focalLength35mm**
- bool **hasCoordinates** = false  
*true if GPS info are present*
- QString **lens**
- QString **make**
- QString **model**
- QString **sensitivity**
- QString **whiteBalance**

## 6.1153 Digikam::PickLabelFilter Class Reference

Inheritance diagram for Digikam::PickLabelFilter:



### Signals

- void **signalPickLabelSelectionChanged** (const QList< PickLabel > &)



## Signals inherited from [Digikam::PickLabelWidget](#)

- void **signalPickLabelChanged** (int)

## Public Member Functions

- QList< [TAlbum](#) \* > **getCheckedPickLabelTags** ()
- **PickLabelFilter** (QWidget \*const parent=nullptr)
- void **reset** ()

## Public Member Functions inherited from [Digikam::PickLabelWidget](#)

- QList< PickLabel > **pickLabels** () const
- **PickLabelWidget** (QWidget \*const parent=nullptr)
- void **setButtonsExclusive** (bool b)
- void **setDescriptionBoxVisible** (bool b)
- void **setPickLabels** (const QList< PickLabel > &list)

## Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::PickLabelWidget](#)

- static QIcon **buildIcon** (PickLabel label)
- static QString **labelPickName** (PickLabel label)

## Protected Member Functions inherited from [Digikam::PickLabelWidget](#)

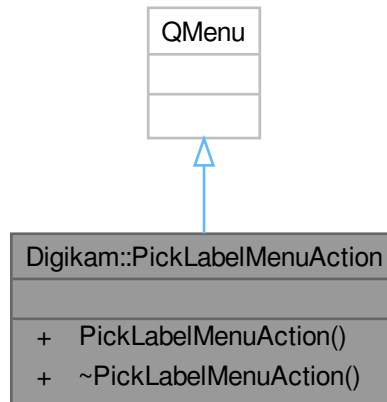
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1154 Digikam::PickLabelMenuAction Class Reference

Inheritance diagram for Digikam::PickLabelMenuAction:



### Signals

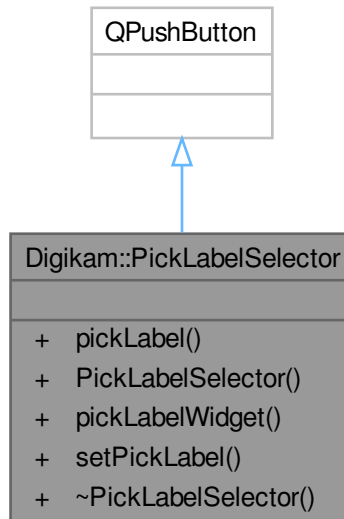
- void **signalPickLabelChanged** (int)

### Public Member Functions

- **PickLabelMenuAction** (QMenu \*const parent=nullptr)

## 6.1155 Digikam::PickLabelSelector Class Reference

Inheritance diagram for Digikam::PickLabelSelector:



### Signals

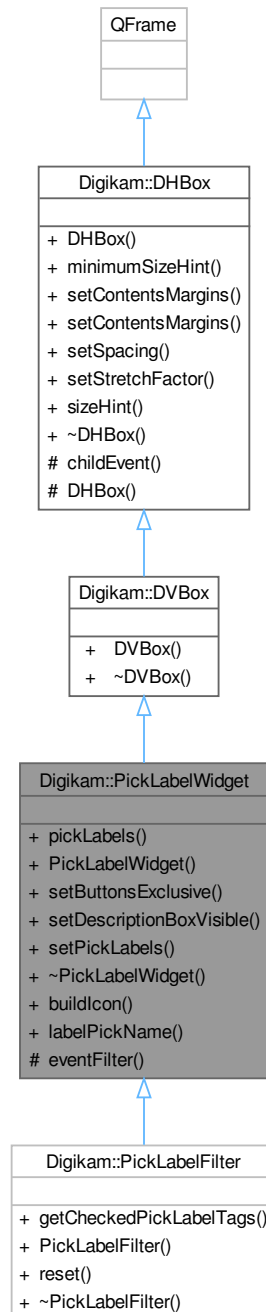
- void **signalPickLabelChanged** (int)

### Public Member Functions

- PickLabel **pickLabel** ()
- **PickLabelSelector** (QWidget \*const parent=nullptr)
- [PickLabelWidget](#) \* **pickLabelWidget** () const
- void **setPickLabel** (PickLabel label)

## 6.1156 Digikam::PickLabelWidget Class Reference

Inheritance diagram for Digikam::PickLabelWidget:



### Signals

- void **signalPickLabelChanged** (int)

## Public Member Functions

- `QList< PickLabel > pickLabels ()` const
- **PickLabelWidget** (`QWidget *const parent=nullptr`)
- void `setButtonsExclusive` (`bool b`)
- void `setDescriptionBoxVisible` (`bool b`)
- void `setPickLabels` (`const QList< PickLabel > &list`)

## Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (`QWidget *const parent=nullptr`)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (`QWidget *const parent=nullptr`)
- `QSize minimumSizeHint ()` const override
- void `setContentsMargins` (`const QMargins &margins`)
- void `setContentsMargins` (`int left, int top, int right, int bottom`)
- void `setSpacing` (`int space`)
- void `setStretchFactor` (`QWidget *const widget, int stretch`)
- `QSize sizeHint ()` const override

## Static Public Member Functions

- static `QIcon buildIcon` (`PickLabel label`)
- static `QString labelPickName` (`PickLabel label`)

## Protected Member Functions

- bool `eventFilter` (`QObject *obj, QEvent *ev`) override

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void `childEvent` (`QChildEvent *e`) override
- **DHBox** (`bool vertical, QWidget *const parent`)

## 6.1156.1 Member Function Documentation

### 6.1156.1.1 pickLabels()

```
QList< PickLabel > Digikam::PickLabelWidget::pickLabels ( ) const
```

Return the list of Pick Label buttons turned on or an empty list of none.

**6.1156.1.2 setButtonsExclusive()**

```
void Digikam::PickLabelWidget::setButtonsExclusive (
    bool b )
```

Set all Color Label buttons exclusive or not. Default is true as only one can be selected. Non-exclusive mode is dedicated for Advanced Search tool.

**6.1156.1.3 setDescriptionBoxVisible()**

```
void Digikam::PickLabelWidget::setDescriptionBoxVisible (
    bool b )
```

Show or not on the bottom view the description of label with shortcuts.

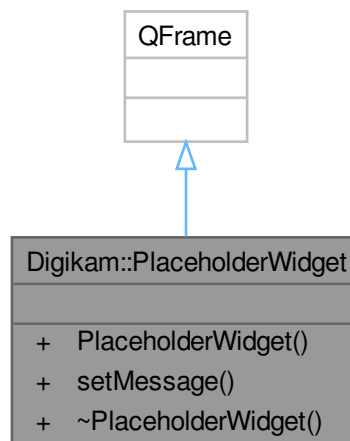
**6.1156.1.4 setPickLabels()**

```
void Digikam::PickLabelWidget::setPickLabels (
    const QList< PickLabel > & list )
```

Turn on Color Label buttons using list. Pass an empty list to clear all selection.

**6.1157 Digikam::PlaceholderWidget Class Reference**

Inheritance diagram for Digikam::PlaceholderWidget:

**Public Member Functions**

- **PlaceholderWidget** (QWidget \*const parent=nullptr)
- void **setMessage** (const QString &message)

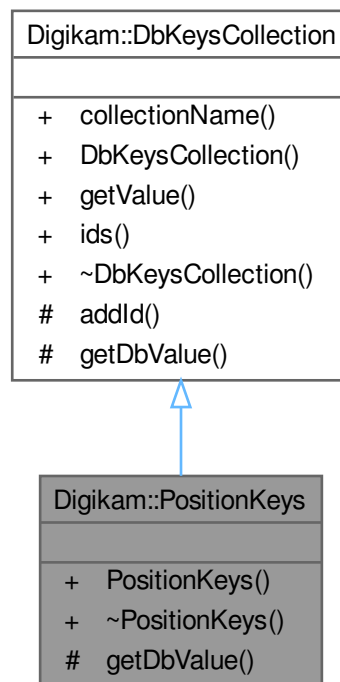
## 6.1158 Digikam::PointTransformAffine Class Reference

### Public Member Functions

- `const std::vector< float > & get_b () const`
- `const std::vector< std::vector< float > > & get_m () const`
- `const std::vector< float > operator() (const std::vector< float > &p) const`
- `PointTransformAffine (const std::vector< std::vector< float > > &m_)`
- `PointTransformAffine (const std::vector< std::vector< float > > &m_, const std::vector< float > &b_)`

## 6.1159 Digikam::PositionKeys Class Reference

Inheritance diagram for Digikam::PositionKeys:



### Public Member Functions

- [PositionKeys \(\)](#)

### Public Member Functions inherited from [Digikam::DbKeysCollection](#)

- `QString collectionName () const`
- `DbKeysCollection (const QString &n)`
- `QString getValue (const QString &key, ParseSettings &settings)`
- `DbKeyIdsMap ids () const`

### Protected Member Functions

- QString [getDbValue](#) (const QString &key, [ParseSettings](#) &settings) override

### Protected Member Functions inherited from [Digikam::DbKeysCollection](#)

- void [addId](#) (const QString &id, const QString &description)

## 6.1159.1 Constructor & Destructor Documentation

### 6.1159.1.1 PositionKeys()

```
Digikam::PositionKeys::PositionKeys ( )
```

## 6.1159.2 Member Function Documentation

### 6.1159.2.1 getDbValue()

```
QString Digikam::PositionKeys::getDbValue (
    const QString & key,
    ParseSettings & settings ) [override], [protected], [virtual]
```

Abstract method for retrieving the value from the database for the given key.

This method has to be implemented by all child classes. It is called by the [getValue\(\)](#) method.

#### Parameters

<i>key</i>	the key representing the value in the database
<i>settings</i>	the <a href="#">ParseSettings</a> object holding all relevant information about the image.

#### Returns

the value of the given database key

#### See also

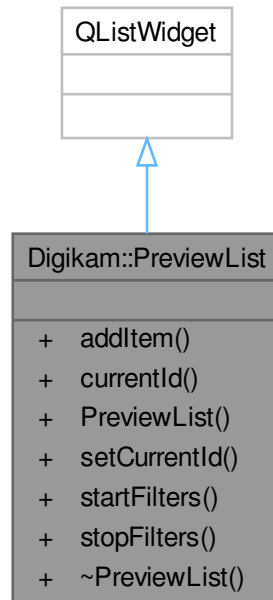
[DbKeysCollection::getValue\(\)](#)

Implements [Digikam::DbKeysCollection](#).



## 6.1160 Digikam::PreviewList Class Reference

Inheritance diagram for Digikam::PreviewList:

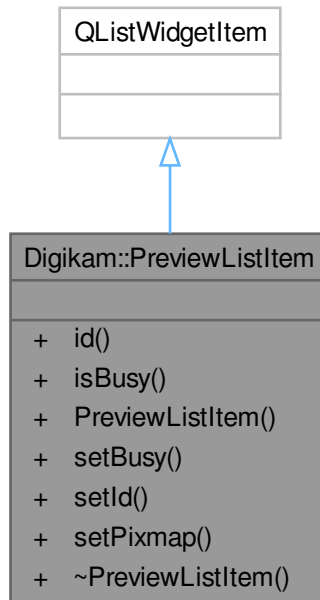


### Public Member Functions

- `PreviewListItem * addItem (DImgThreadedFilter *const filter, const QString &txt, int id)`
- `int currentId () const`
- `PreviewList (QWidget *const parent=nullptr)`
- `void setCurrentId (int id)`
- `void startFilters ()`
- `void stopFilters ()`

## 6.1161 Digikam::PreviewListItem Class Reference

Inheritance diagram for Digikam::PreviewListItem:

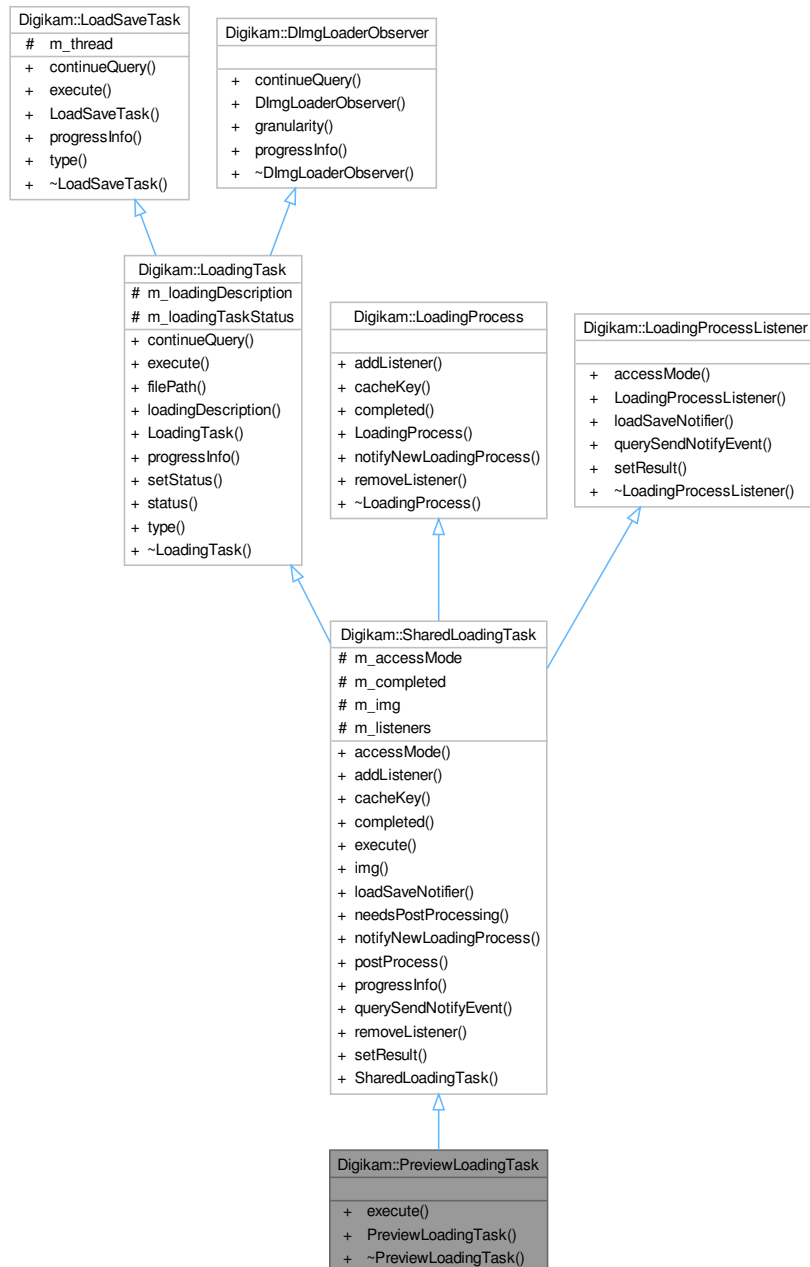


### Public Member Functions

- `int id () const`
- `bool isBusy () const`
- `PreviewListItem (QListWidgetItem *const parent=nullptr)`
- `void setBusy (bool b)`
- `void setId (int id)`
- `void setPixmap (const QPixmap &pix)`

## 6.1162 Digikam::PreviewLoadingTask Class Reference

Inheritance diagram for Digikam::PreviewLoadingTask:



### Public Member Functions

- void `execute()` override
- `PreviewLoadingTask` (`LoadSaveThread` \*const thread, const `LoadingDescription` &description)

## Public Member Functions inherited from [Digikam::SharedLoadingTask](#)

- [LoadSaveThread::AccessMode](#) `accessMode ()` const override
- void `addListener (LoadingProcessListener *const listener)` override
- QString `cacheKey ()` const override
- bool `completed ()` const override
- void `execute ()` override
- [DImg](#) `img ()` const
- [LoadSaveNotifier](#) \* `loadSaveNotifier ()` const override
- bool `needsPostProcessing ()` const
- void `notifyNewLoadingProcess (LoadingProcess *const process, const LoadingDescription &description)` override
- virtual void `postProcess ()`
- void `progressInfo (float progress)` override
- bool `querySendNotifyEvent ()` const override
- void `removeListener (LoadingProcessListener *const listener)` override
- void `setResult (const LoadingDescription &loadingDescription, const DImg &img)` override
- **SharedLoadingTask** ([LoadSaveThread](#) \*const thread, const [LoadingDescription](#) &description, [LoadSaveThread::AccessMode](#) mode=[LoadSaveThread::AccessModeReadWrite](#), [LoadingTaskStatus](#) loadingTaskStatus=[LoadingTaskStatusLoading](#))

## Public Member Functions inherited from [Digikam::LoadingTask](#)

- bool `continueQuery ()` override
- QString `filePath ()` const
- const [LoadingDescription](#) & `loadingDescription ()` const
- **LoadingTask** ([LoadSaveThread](#) \*const thread, const [LoadingDescription](#) &description, [LoadingTaskStatus](#) loadingTaskStatus=[LoadingTaskStatusLoading](#))
- void `setStatus (LoadingTaskStatus status)`
- [LoadingTaskStatus](#) `status ()` const
- [TaskType](#) `type ()` override

## Public Member Functions inherited from [Digikam::LoadSaveTask](#)

- **LoadSaveTask** ([LoadSaveThread](#) \*const thread)

## Public Member Functions inherited from [Digikam::DImgLoaderObserver](#)

- virtual float `granularity ()`

## Additional Inherited Members

## Public Types inherited from [Digikam::LoadingTask](#)

- enum **LoadingTaskStatus** { [LoadingTaskStatusLoading](#) , [LoadingTaskStatusPreloading](#) , [LoadingTaskStatusStopping](#) }

## Public Types inherited from [Digikam::LoadSaveTask](#)

- enum **TaskType** { [TaskTypeLoading](#) , [TaskTypeSaving](#) }

### Protected Attributes inherited from [Digikam::SharedLoadingTask](#)

- [LoadSaveThread::AccessMode](#) **m\_accessMode** = [LoadSaveThread::AccessModeReadWrite](#)
- volatile bool **m\_completed** = false
- [DImg](#) **m\_img**
- [QList](#)< [LoadingProcessListener](#) \* > **m\_listeners**

### Protected Attributes inherited from [Digikam::LoadingTask](#)

- [LoadingDescription](#) **m\_loadingDescription**
- volatile [LoadingTaskStatus](#) **m\_loadingTaskStatus** = [LoadingTaskStatusLoading](#)

### Protected Attributes inherited from [Digikam::LoadSaveTask](#)

- [LoadSaveThread](#) \* **m\_thread** = nullptr

## 6.1162.1 Member Function Documentation

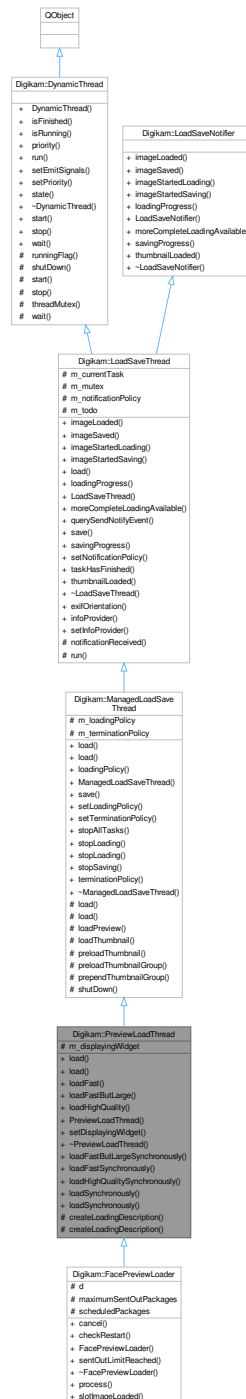
### 6.1162.1.1 execute()

```
void Digikam::PreviewLoadingTask::execute ( ) [override], [virtual]
```

Reimplemented from [Digikam::LoadingTask](#).

## 6.1163 Digikam::PreviewLoadThread Class Reference

Inheritance diagram for Digikam::PreviewLoadThread:



### Public Member Functions

- void [load](#) (const [LoadingDescription](#) &description)
- void [load](#) (const QString &filePath, const [PreviewSettings](#) &settings, int size=0)

- void [loadFast](#) (const QString &filePath, int size)
- void [loadFastButLarge](#) (const QString &filePath, int minimumSize)
- void [loadHighQuality](#) (const QString &filePath, PreviewSettings::RawLoading rawLoadingMode=PreviewSettings::RawPreviewAutomatic)
- [PreviewLoadThread](#) (QObject \*const parent=nullptr)
- void [setDisplayingWidget](#) (QWidget \*const widget)

### Public Member Functions inherited from [Digikam::ManagedLoadSaveThread](#)

- void [load](#) (const [LoadingDescription](#) &description)
- void [load](#) (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- [LoadingPolicy](#) [loadingPolicy](#) () const
- [ManagedLoadSaveThread](#) (QObject \*const parent=nullptr)
- void [save](#) (const [DImg](#) &image, const QString &filePath, const QString &format)
- void [setLoadingPolicy](#) ([LoadingPolicy](#) policy)
- void [setTerminationPolicy](#) ([TerminationPolicy](#) terminationPolicy)
- void [stopAllTasks](#) ()
- void [stopLoading](#) (const [LoadingDescription](#) &desc, [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopLoading](#) (const QString &filePath=QString(), [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopSaving](#) (const QString &filePath=QString())
- [TerminationPolicy](#) [terminationPolicy](#) () const

### Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- void [imageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img) override
- void [imageSaved](#) (const QString &filePath, bool success) override
- void [imageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription) override
- void [imageStartedSaving](#) (const QString &filePath) override
- void [load](#) (const [LoadingDescription](#) &description)
- void [loadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress) override
- [LoadSaveThread](#) (QObject \*const parent=nullptr)
- void [moreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription) override
- virtual bool [querySendNotifyEvent](#) () const
- void [save](#) (const [DImg](#) &image, const QString &filePath, const QString &format)
- void [savingProgress](#) (const QString &filePath, float progress) override
- void [setNotificationPolicy](#) ([NotificationPolicy](#) notificationPolicy)
- virtual void [taskHasFinished](#) ()
- void [thumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const QImage &img) override
- [~LoadSaveThread](#) () override

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static [DImg](#) **loadFastButLargeSynchronously** (const QString &filePath, int minimumSize, const [IccProfile](#) &profile=[IccProfile\(\)](#))
- static [DImg](#) **loadFastSynchronously** (const QString &filePath, int size, const [IccProfile](#) &profile=[IccProfile\(\)](#))
- static [DImg](#) **loadHighQualitySynchronously** (const QString &filePath, [PreviewSettings::RawLoading](#) rawLoadingMode=[PreviewSettings::RawPreviewAutomatic](#), const [IccProfile](#) &profile=[IccProfile\(\)](#))
- static [DImg](#) **loadSynchronously** (const [LoadingDescription](#) &description)
- static [DImg](#) **loadSynchronously** (const QString &filePath, const [PreviewSettings](#) &previewSettings, int size, const [IccProfile](#) &profile=[IccProfile\(\)](#))

### Static Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- static int [exifOrientation](#) (const QString &filePath, const [DMetadata](#) &metadata, bool isRaw, bool fromRawEmbeddedPreview)
- static [LoadSaveFileInfoProvider](#) \* **infoProvider** ()
- static void **setInfoProvider** ([LoadSaveFileInfoProvider](#) \*const infoProvider)

### Protected Member Functions

- [LoadingDescription](#) **createLoadingDescription** (const QString &filePath, const [PreviewSettings](#) &settings, int size)

### Protected Member Functions inherited from [Digikam::ManagedLoadSaveThread](#)

- void **load** (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void **load** (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [LoadingPolicy](#) policy, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void **loadPreview** (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- void **loadThumbnail** (const [LoadingDescription](#) &description)
- void **preloadThumbnail** (const [LoadingDescription](#) &description)
- void **preloadThumbnailGroup** (const QList< [LoadingDescription](#) > &descriptions)
- void **prependThumbnailGroup** (const QList< [LoadingDescription](#) > &descriptions)
- void **shutDown** ()

### Protected Member Functions inherited from [Digikam::LoadSaveThread](#)

- void **notificationReceived** ()
- void [run](#) () override

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void **wait** (QMutexLocker< QMutex > &locker)



### Static Protected Member Functions

- static [LoadingDescription](#) **createLoadingDescription** (const QString &filePath, const [PreviewSettings](#) &settings, int size, const [IccProfile](#) &profile)

### Protected Attributes

- QWidget \* **m\_displayingWidget** = nullptr

### Protected Attributes inherited from [Digikam::ManagedLoadSaveThread](#)

- [LoadingPolicy](#) **m\_loadingPolicy** = [LoadingPolicyAppend](#)
- [TerminationPolicy](#) **m\_terminationPolicy** = [TerminationPolicyTerminateLoading](#)

### Protected Attributes inherited from [Digikam::LoadSaveThread](#)

- [LoadSaveTask](#) \* **m\_currentTask** = nullptr
- QMutex **m\_mutex**
- [NotificationPolicy](#) **m\_notificationPolicy** = [NotificationPolicyTimeLimited](#)
- QList< [LoadSaveTask](#) \* > **m\_todo**

### Additional Inherited Members

### Public Types inherited from [Digikam::ManagedLoadSaveThread](#)

- enum [LoadingMode](#) { [LoadingModeNormal](#) , [LoadingModeShared](#) }
- enum [LoadingPolicy](#) { [LoadingPolicyFirstRemovePrevious](#) , [LoadingPolicyPrepend](#) , [LoadingPolicySimplePrepend](#) , [LoadingPolicyAppend](#) , [LoadingPolicySimpleAppend](#) , [LoadingPolicyPreload](#) }
- enum [LoadingTaskFilter](#) { [LoadingTaskFilterAll](#) , [LoadingTaskFilterPreloading](#) }
- enum [TerminationPolicy](#) { [TerminationPolicyTerminateLoading](#) , [TerminationPolicyTerminatePreloading](#) , [TerminationPolicyWait](#) , [TerminationPolicyTerminateAll](#) }

### Public Types inherited from [Digikam::LoadSaveThread](#)

- enum [AccessMode](#) { [AccessModeRead](#) , [AccessModeReadWrite](#) }
- enum [NotificationPolicy](#) { [NotificationPolicyDirect](#) , [NotificationPolicyTimeLimited](#) }

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

### Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::LoadSaveThread](#)

- void [signalImageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)
- void [signalImageSaved](#) (const [QString](#) &filePath, bool success)
- void [signalImageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription)
- void [signalImageStartedSaving](#) (const [QString](#) &filePath)
- void [signalLoadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress)
- void [signalMoreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription)
- void [signalSavingProgress](#) (const [QString](#) &filePath, float progress)
- void [signalThumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [QImage](#) &img)

## Signals inherited from [Digikam::DynamicThread](#)

- void [finished](#) ()
- void [starting](#) ()

## 6.1163.1 Constructor & Destructor Documentation

### 6.1163.1.1 [PreviewLoadThread\(\)](#)

```
Digikam::PreviewLoadThread::PreviewLoadThread (
    QObject *const parent = nullptr ) [explicit]
```

Creates a preview load thread. Provides three flavors of preview loading. The default loading policy, for the typical usage in a preview widget, always stops any previous tasks and loads the new task as soon as possible.

## 6.1163.2 Member Function Documentation

### 6.1163.2.1 [load\(\)](#) [1/2]

```
void Digikam::PreviewLoadThread::load (
    const LoadingDescription & description )
```

Load a preview. Loading description will not be touched.

### 6.1163.2.2 [load\(\)](#) [2/2]

```
void Digikam::PreviewLoadThread::load (
    const QString & filePath,
    const PreviewSettings & settings,
    int size = 0 )
```

Load a preview. Settings determine the loading mode. For fast loading, size is preview area size. For fast-but-large loading, it serves as a minimum size. For high quality loading, it is ignored

### 6.1163.2.3 loadFast()

```
void Digikam::PreviewLoadThread::loadFast (
    const QString & filePath,
    int size )
```

Load a preview that is optimized for fast loading. Raw decoding and color management settings will be adjusted.

### 6.1163.2.4 loadFastButLarge()

```
void Digikam::PreviewLoadThread::loadFastButLarge (
    const QString & filePath,
    int minimumSize )
```

Load a preview that is as large as possible without sacrificing speed for performance. Especially, raw previews are taken if larger than the given size. Raw decoding and color management settings will be adjusted.

### 6.1163.2.5 loadFastSynchronously()

```
DImg Digikam::PreviewLoadThread::loadFastSynchronously (
    const QString & filePath,
    int size,
    const IccProfile & profile = IccProfile() ) [static]
```

Synchronous versions of the above methods. These are safe to call from the non-UI thread, as the [IccProfile](#) either passed or deduced independent from a displaying widget

### 6.1163.2.6 loadHighQuality()

```
void Digikam::PreviewLoadThread::loadHighQuality (
    const QString & filePath,
    PreviewSettings::RawLoading rawLoadingMode = PreviewSettings::RawPreviewAutomatic
)
```

Load a preview with higher resolution, trading more quality for less speed. Raw decoding and color management settings will be adjusted.

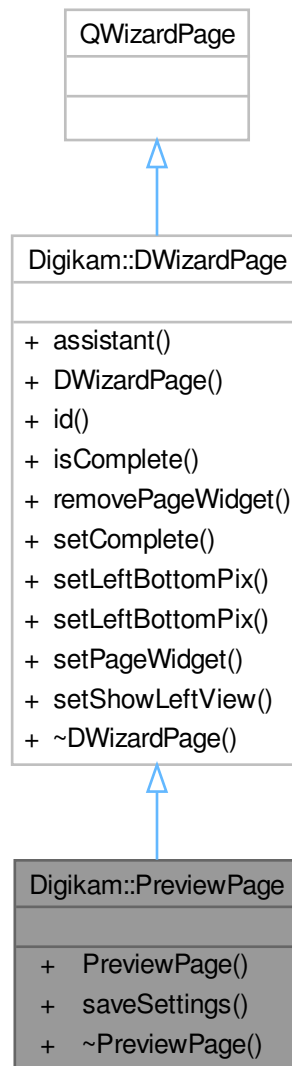
### 6.1163.2.7 setDisplayingWidget()

```
void Digikam::PreviewLoadThread::setDisplayingWidget (
    QWidget *const widget )
```

Optionally, set the displaying widget for color management

## 6.1164 Digikam::PreviewPage Class Reference

Inheritance diagram for Digikam::PreviewPage:



### Public Member Functions

- **PreviewPage** (QWizard \*const dlg)
- void **saveSettings** ()

### Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)

- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.1165 Digikam::PreviewSettings Class Reference

### Public Types

- enum [Quality](#) { [FastPreview](#) , [FastButLargePreview](#) , [HighQualityPreview](#) }
- enum [RawLoading](#) { [RawPreviewAutomatic](#) , [RawPreviewFromEmbeddedPreview](#) , [RawPreviewFromRawHalfSize](#) , [RawPreviewFromRawFullSize](#) }

### Public Member Functions

- bool **operator==** (const [PreviewSettings](#) &other) const
- [PreviewSettings](#) ([Quality](#) quality=[HighQualityPreview](#), [RawLoading](#) rawLoading=[RawPreviewAutomatic](#))

### Static Public Member Functions

- static [PreviewSettings](#) **fastPreview** ()
- static [PreviewSettings](#) **highQualityPreview** ()

### Public Attributes

- bool **convertToEightBit** = false
- [Quality](#) **quality**
- [RawLoading](#) **rawLoading**

## 6.1165.1 Member Enumeration Documentation

### 6.1165.1.1 Quality

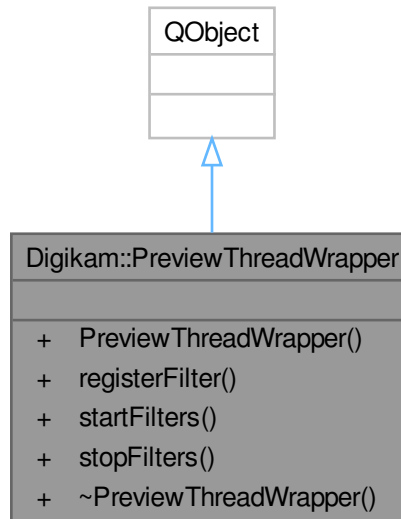
enum [Digikam::PreviewSettings::Quality](#)

#### Enumerator

FastPreview	A preview were loading time is most important. Preview can be reduced in size. Additionally specifying the size of the preview area may be appropriate
FastButLargePreview	Load a preview that is as large as possible without sacrificing speed for performance. Especially, raw previews are taken if larger than the given size. Raw decoding and color management settings will be adjusted.
HighQualityPreview	Load a high quality additional image. For normal images, loads the full data. For RAW, the additional settings below are taken into account

## 6.1166 Digikam::PreviewThreadWrapper Class Reference

Inheritance diagram for Digikam::PreviewThreadWrapper:



### Signals

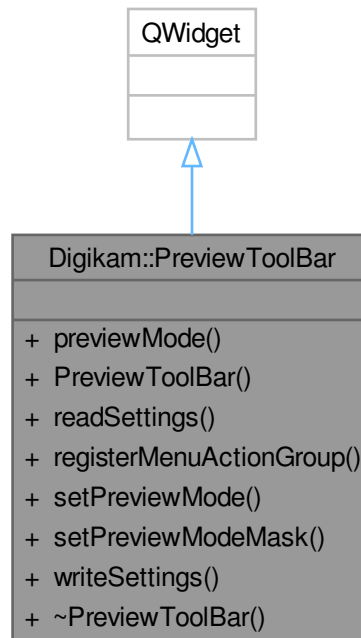
- void **signalFilterFinished** (int, const QPixmap &)
- void **signalFilterStarted** (int)

### Public Member Functions

- **PreviewThreadWrapper** (QObject \*const parent=nullptr)
- void **registerFilter** (int id, [DImgThreadedFilter](#) \*const filter)
- void **startFilters** ()
- void **stopFilters** ()

## 6.1167 Digikam::PreviewToolBar Class Reference

Inheritance diagram for Digikam::PreviewToolBar:



### Public Types

- enum `PreviewMode` {  
`PreviewOriginalImage` = 0x00000001 , `PreviewBothImagesHorz` = 0x00000002 , `PreviewBothImagesVert` = 0x00000004 , `PreviewBothImagesHorzCont` = 0x00000008 ,  
`PreviewBothImagesVertCont` = 0x00000010 , `PreviewTargetImage` = 0x00000020 , `PreviewToggleOnMouseOver` = 0x00000040 , `NoPreviewMode` = 0x00000080 ,  
**AllPreviewModes** , **UnSplitPreviewModes** = `PreviewOriginalImage` | `PreviewTargetImage` | `PreviewToggleOnMouseOver` }

### Signals

- void **signalPreviewModeChanged** (int)

### Public Member Functions

- void **PreviewMode previewMode** () const
- void **PreviewToolBar** (QWidget \*const parent=nullptr)
- void **readSettings** (const KConfigGroup &group)
- void **registerMenuActionGroup** (EditorWindow \*const editor)
- void **setPreviewMode** (PreviewMode mode)
- void **setPreviewModeMask** (int mask)
- void **writeSettings** (KConfigGroup &group)

## 6.1167.1 Member Enumeration Documentation

### 6.1167.1.1 PreviewMode

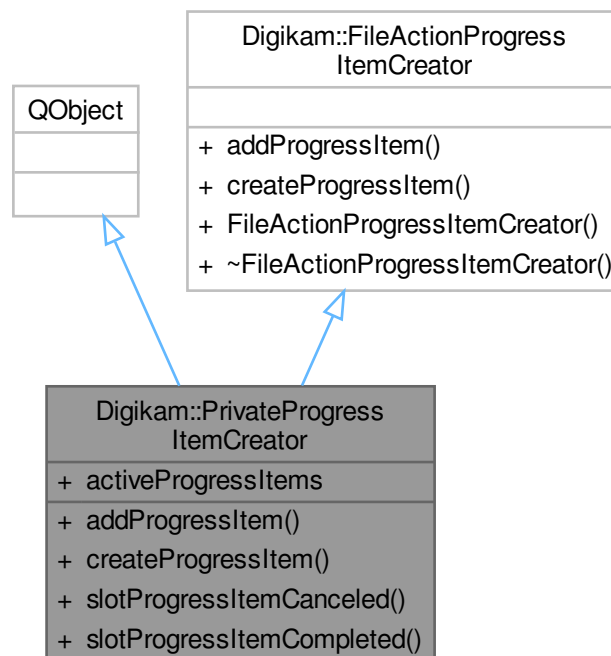
enum `Digikam::PreviewToolBar::PreviewMode`

Enumerator

<code>PreviewOriginalImage</code>	Original image only.
<code>PreviewBothImagesHorz</code>	Horizontal with original and target duplicated.
<code>PreviewBothImagesVert</code>	Vertical with original and target duplicated.
<code>PreviewBothImagesHorzCont</code>	Horizontal with original and target in contiguous.
<code>PreviewBothImagesVertCont</code>	Vertical with original and target in contiguous.
<code>PreviewTargetImage</code>	Target image only.
<code>PreviewToggleOnMouseOver</code>	Original image if mouse is over image area, else target image.
<code>NoPreviewMode</code>	Target image only without information displayed.

## 6.1168 Digikam::PrivateProgressItemCreator Class Reference

Inheritance diagram for `Digikam::PrivateProgressItemCreator`:





## Public Slots

- void **slotProgressItemCanceled** ([ProgressItem](#) \*item)
- void **slotProgressItemCompleted** ()

## Signals

- void **lastItemCompleted** ()

## Public Member Functions

- void **addProgressItem** ([ProgressItem](#) \*const item) override
- [ProgressItem](#) \* **createProgressItem** (const QString &action) const override

## Public Attributes

- QAtomicInt **activeProgressItems**

## 6.1168.1 Member Function Documentation

### 6.1168.1.1 addProgressItem()

```
void Digikam::PrivateProgressItemCreator::addProgressItem (  
    ProgressItem *const item ) [override], [virtual]
```

Implements [Digikam::FileActionProgressItemCreator](#).

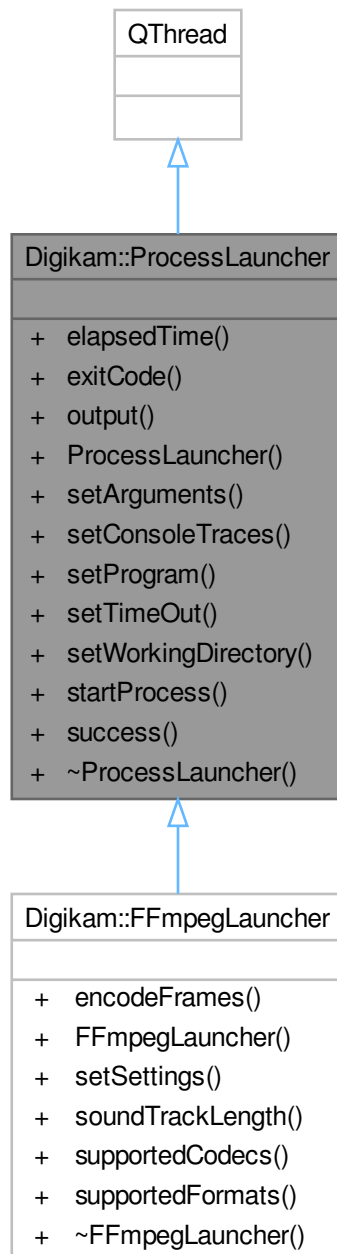
### 6.1168.1.2 createProgressItem()

```
ProgressItem * Digikam::PrivateProgressItemCreator::createProgressItem (  
    const QString & action ) const [override], [virtual]
```

Implements [Digikam::FileActionProgressItemCreator](#).

## 6.1169 Digikam::ProcessLauncher Class Reference

Inheritance diagram for Digikam::ProcessLauncher:



### Signals

- void **signalComplete** (bool `success`, int `exitCode`)

## Public Member Functions

- qint64 [elapsedTime](#) () const
- int [exitCode](#) () const
- QString [output](#) () const
- **ProcessLauncher** (QObject \*const parent=nullptr)
- void **setArguments** (const QStringList &args)
- void [setConsoleTraces](#) (bool b)
- void **setProgram** (const QString &prog)
- void **setTimeout** (int msec)
- void **setWorkingDirectory** (const QString &dir)
- void [startProcess](#) ()
- bool [success](#) () const

## 6.1169.1 Member Function Documentation

### 6.1169.1.1 [elapsedTime\(\)](#)

```
qint64 Digikam::ProcessLauncher::elapsedTime ( ) const
```

Return the elapsed time in ms to run the process.

### 6.1169.1.2 [exitCode\(\)](#)

```
int Digikam::ProcessLauncher::exitCode ( ) const
```

Return the exit code from the process.

### 6.1169.1.3 [output\(\)](#)

```
QString Digikam::ProcessLauncher::output ( ) const
```

Return the process output as string.

### 6.1169.1.4 [setConsoleTraces\(\)](#)

```
void Digikam::ProcessLauncher::setConsoleTraces (
    bool b )
```

If turned on, all traces from the process are printed on the console.

### 6.1169.1.5 [startProcess\(\)](#)

```
void Digikam::ProcessLauncher::startProcess ( )
```

Start the process.

### 6.1169.1.6 success()

```
bool Digikam::ProcessLauncher::success ( ) const
```

Return true if the process is started and completed without error.

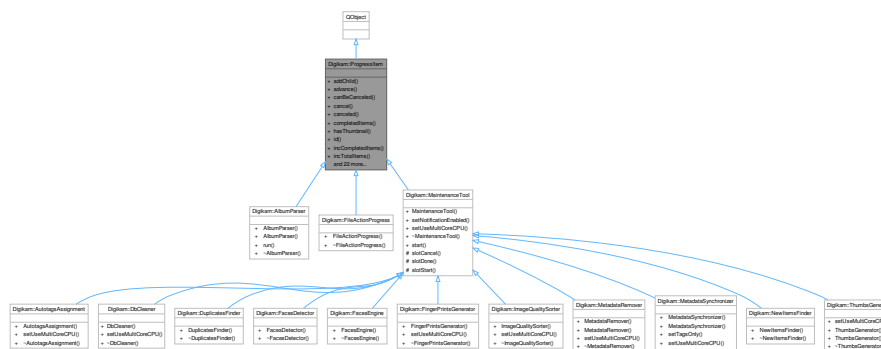
## 6.1170 Digikam::ProgressEntry Class Reference

### Public Attributes

- bool **canCancel** = false
- QString **message**
- float **progress** = 0.0

## 6.1171 Digikam::ProgressItem Class Reference

Inheritance diagram for Digikam::ProgressItem:



### Signals

- void **progressItemAdded** (ProgressItem \*item)  
*Emitted when a new ProgressItem is added.*
- void **progressItemCanceled** (ProgressItem \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a ProgressManager::slotStandardCancelHandler which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const QString &id)
- void **progressItemCompleted** (ProgressItem \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void **progressItemLabel** (ProgressItem \*item, const QString &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void **progressItemProgress** (ProgressItem \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void **progressItemStatus** (ProgressItem \*item, const QString &mess)

*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*

- void [progressItemThumbnail](#) ([ProgressItem](#) \*item, const QPixmap &thumb)

*Emitted when the thumbnail data must be set in item.*

- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*item, bool value)

*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

## Public Member Functions

- void **addChild** ([ProgressItem](#) \*const kiddo)

- bool [advance](#) (unsigned int v)

*Advance total items processed by n values and update percentage in progressbar.*

- bool [canBeCanceled](#) () const

- void **cancel** ()

- bool **canceled** () const

- unsigned int **completedItems** () const

- bool [hasThumbnail](#) () const

- const QString & [id](#) () const

- bool **incCompletedItems** (unsigned int v=1)

- void **incTotalItems** (unsigned int v=1)

- const QString & [label](#) () const

- [ProgressItem](#) \* [parent](#) () const

- unsigned int [progress](#) () const

- [ProgressItem](#) ([ProgressItem](#) \*const [parent](#), const QString &[id](#), const QString &[label](#), const QString &[status](#), bool [canBeCanceled](#), bool [hasThumb](#))

- void **removeChild** ([ProgressItem](#) \*const kiddo)

- void **reset** ()

*Reset the progress value of this item to 0 and the status string to the empty string.*

- void **setComplete** ()

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*

- bool **setCompletedItems** (unsigned int v)

- void [setLabel](#) (const QString &v)

- void [setProgress](#) (unsigned int v)

*Set the progress (percentage of completion) value of this item.*

- void [setShowAtStart](#) (bool [showAtStart](#))

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*

- void [setStatus](#) (const QString &v)

*Set the string to be used for showing this item's current status.*

- void [setThumbnail](#) (const QIcon &icon)

*Sets whether this item has a thumbnail.*

- void **setTotalItems** (unsigned int v)

- void [setUsesBusyIndicator](#) (bool useBusyIndicator)

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*

- bool [showAtStart](#) () const

- const QString & [status](#) () const

- bool **totalCompleted** () const

- unsigned int **totalItems** () const

- void **updateProgress** ()

*Recalculate progress according to total/completed items and update.*

- bool [usesBusyIndicator](#) () const

## 6.1171.1 Member Function Documentation

### 6.1171.1.1 advance()

```
bool Digikam::ProgressItem::advance (
    unsigned int v )
```

#### Parameters

v	The value to advance.
---	-----------------------

#### Returns

true if totalCompleted()

### 6.1171.1.2 canBeCanceled()

```
bool Digikam::ProgressItem::canBeCanceled ( ) const
```

#### Returns

Whether this item can be canceled.

### 6.1171.1.3 hasThumbnail()

```
bool Digikam::ProgressItem::hasThumbnail ( ) const
```

#### Returns

whether this item has a thumbnail.

### 6.1171.1.4 id()

```
const QString & Digikam::ProgressItem::id ( ) const
```

#### Returns

The id string which uniquely identifies the operation represented by this item.

### 6.1171.1.5 label()

```
const QString & Digikam::ProgressItem::label ( ) const
```

#### Returns

The user visible string to be used to represent this item.

### 6.1171.1.6 parent()

```
ProgressItem * Digikam::ProgressItem::parent ( ) const
```

#### Returns

The parent item of this one, if there is one.

### 6.1171.1.7 progress()

```
unsigned int Digikam::ProgressItem::progress ( ) const
```

#### Returns

The current progress value of this item in percent.

### 6.1171.1.8 progressItemAdded

```
void Digikam::ProgressItem::progressItemAdded (
    ProgressItem * item ) [signal]
```

#### Parameters

<i>item</i>	The <a href="#">ProgressItem</a> that was added.
-------------	--

### 6.1171.1.9 progressItemCanceled

```
void Digikam::ProgressItem::progressItemCanceled (
    ProgressItem * item ) [signal]
```

#### Parameters

<i>item</i>	The canceled item;
-------------	--------------------

### 6.1171.1.10 progressItemCompleted

```
void Digikam::ProgressItem::progressItemCompleted (
    ProgressItem * item ) [signal]
```

#### Parameters

<i>item</i>	The completed item.
-------------	---------------------

**6.1171.1.11 progressItemLabel**

```
void Digikam::ProgressItem::progressItemLabel (
    ProgressItem * item,
    const QString & label ) [signal]
```

**Parameters**

<i>item</i>	The updated item.
<i>label</i>	The new label.

**6.1171.1.12 progressItemProgress**

```
void Digikam::ProgressItem::progressItemProgress (
    ProgressItem * item,
    unsigned int v ) [signal]
```

**Parameters**

<i>item</i>	The item which got a new value.
<i>v</i>	The value, for convenience.

**6.1171.1.13 progressItemStatus**

```
void Digikam::ProgressItem::progressItemStatus (
    ProgressItem * item,
    const QString & mess ) [signal]
```

**Parameters**

<i>item</i>	The updated item.
<i>mess</i>	The new message.

**6.1171.1.14 progressItemThumbnail**

```
void Digikam::ProgressItem::progressItemThumbnail (
    ProgressItem * item,
    const QPixmap & thumb ) [signal]
```

**Parameters**

<i>item</i>	The updated item
<i>thumb</i>	thumbnail data



### 6.1171.1.15 progressItemUsesBusyIndicator

```
void Digikam::ProgressItem::progressItemUsesBusyIndicator (
    ProgressItem * item,
    bool value ) [signal]
```

#### Parameters

<i>item</i>	The updated item
<i>value</i>	True if the item uses a busy indicator now, false otherwise

### 6.1171.1.16 setLabel()

```
void Digikam::ProgressItem::setLabel (
    const QString & v )
```

#### Parameters

<i>v</i>	Set the user visible string identifying this item.
----------	--

### 6.1171.1.17 setProgress()

```
void Digikam::ProgressItem::setProgress (
    unsigned int v )
```

#### Parameters

<i>v</i>	The percentage value.
----------	-----------------------

### 6.1171.1.18 setShowAtStart()

```
void Digikam::ProgressItem::setShowAtStart (
    bool showAtStart )
```

#### Parameters

<i>showAtStart</i>	The flag to turn on this property.
--------------------	------------------------------------

### 6.1171.1.19 setStatus()

```
void Digikam::ProgressItem::setStatus (
    const QString & v )
```

**Parameters**

v	The status string.
---	--------------------

**6.1171.1.20 setThumbnail()**

```
void Digikam::ProgressItem::setThumbnail (
    const QIcon & icon )
```

**Parameters**

<i>icon</i>	The icon to use as thumbnail.
-------------	-------------------------------

**6.1171.1.21 setUsesBusyIndicator()**

```
void Digikam::ProgressItem::setUsesBusyIndicator (
    bool useBusyIndicator )
```

param useBusyIndicator The flag to indicate busy state.

**6.1171.1.22 showAtStart()**

```
bool Digikam::ProgressItem::showAtStart ( ) const
```

**Returns**

true if item must be pop-up when it's added in progress manager.

**6.1171.1.23 status()**

```
const QString & Digikam::ProgressItem::status ( ) const
```

**Returns**

The string to be used for showing this item's current status.

**6.1171.1.24 usesBusyIndicator()**

```
bool Digikam::ProgressItem::usesBusyIndicator ( ) const
```

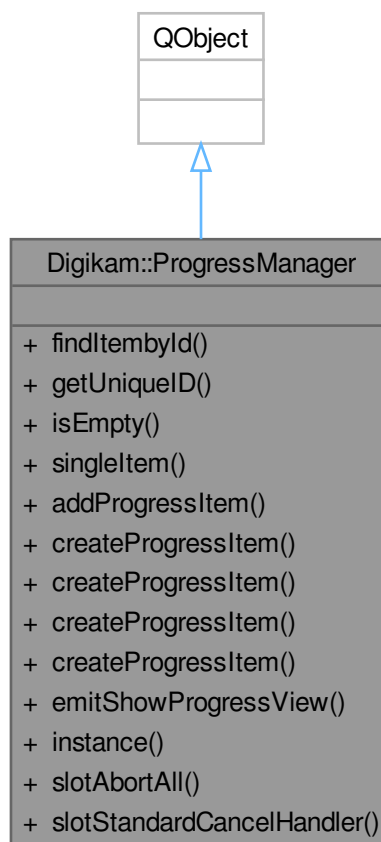
**Returns**

whether this item uses a busy indicator instead of real progress display

## 6.1172 Digikam::ProgressManager Class Reference

The [ProgressManager](#) singleton keeps track of all ongoing transactions and notifies observers (progress dialogs) when their progress percent value changes, when they are completed (by their owner), and when they are canceled. Each [ProgressItem](#) emits those signals individually and the singleton broadcasts them. Use the [createProgressItem\(\)](#) statics to acquire an item and then call `->setProgress( int percent )` on it every time you want to update the item and `->setComplete()` when the operation is done. This will delete the item. Connect to the item's [progressItemCanceled\(\)](#) signal to be notified when the user cancels the transaction using one of the observing progress dialogs or by calling `item->cancel()` in some other way. The owner is responsible for calling `setComplete()` on the item, even if it is canceled. Use the `standardCancelHandler()` slot if that is all you want to do on cancel.

Inheritance diagram for Digikam::ProgressManager:



### Public Slots

- void **slotAbortAll** ()  
*Aborts all running jobs. Bound to "Esc".*
- void **slotStandardCancelHandler** ([ProgressItem](#) \*item)  
*Calls `setCompleted()` on the item, to make sure it goes away. Provided for convenience.*

## Signals

- void **completeTransactionDeferred** ([ProgressItem](#) \*item)
- void [progressItemAdded](#) ([ProgressItem](#) \*)
- void [progressItemCanceled](#) ([ProgressItem](#) \*)
- void [progressItemCompleted](#) ([ProgressItem](#) \*)
- void [progressItemLabel](#) ([ProgressItem](#) \*, const [QString](#) &)
- void [progressItemProgress](#) ([ProgressItem](#) \*, unsigned int)
- void [progressItemStatus](#) ([ProgressItem](#) \*, const [QString](#) &)
- void [progressItemThumbnail](#) ([ProgressItem](#) \*, const [QPixmap](#) &)
- void [progressItemUsesBusyIndicator](#) ([ProgressItem](#) \*, bool)
- void **showProgressView** ()

*Emitted when an operation requests the listeners to be shown. Use [emitShowProgressView\(\)](#) to trigger it.*

## Public Member Functions

- [ProgressItem](#) \* [findItemById](#) (const [QString](#) &id) const
- [QString](#) [getUniqueId](#) ()
  - Use this to acquire a unique id number which can be used to discern an operation from all others going on at the same time. Use that number as the id string for your progressItem to ensure it is unique.*
- bool [isEmpty](#) () const
- [ProgressItem](#) \* [singleItem](#) () const

## Static Public Member Functions

- static bool [addProgressItem](#) ([ProgressItem](#) \*const t, [ProgressItem](#) \*const parent=nullptr)
  - Add a created progressItem outside manager with the given parent.*
- static [ProgressItem](#) \* [createProgressItem](#) (const [QString](#) &id, const [QString](#) &label, const [QString](#) &status=[QString](#)(), bool canBeCanceled=true, bool hasThumb=false)
  - Use this version if you have the id string of the parent but without the parent instance.*
- static [ProgressItem](#) \* [createProgressItem](#) (const [QString](#) &label, const [QString](#) &status=[QString](#)(), bool canBeCanceled=true, bool hasThumb=false)
  - Creates a [ProgressItem](#) with a unique id and the given label. This is the simplest way to acquire a progress item. It will not have a parent.*
- static [ProgressItem](#) \* [createProgressItem](#) (const [QString](#) &parent, const [QString](#) &id, const [QString](#) &label, const [QString](#) &status=[QString](#)(), bool canBeCanceled=true, bool hasThumb=false)
  - Use this version if you have the id string of the parent and want to add a subjob to it.*
- static [ProgressItem](#) \* [createProgressItem](#) ([ProgressItem](#) \*const parent, const [QString](#) &id, const [QString](#) &label, const [QString](#) &status=[QString](#)(), bool canBeCanceled=true, bool hasThumb=false)
  - Creates a new progressItem with the given parent, id, label and initial status.*
- static void **emitShowProgressView** ()
  - Ask all listeners to show the progress dialog, because there is something that wants to be shown.*
- static [ProgressManager](#) \* [instance](#) ()

## Friends

- class **ProgressManagerCreator**

## 6.1172.1 Detailed Description

### Note

if you request an item with a certain id and there is already one with that id, there will not be a new one created but the existing one will be returned. This is convenient for accessing items that are needed regularly without the to store a pointer to them or to add child items to parents by id.

## 6.1172.2 Member Function Documentation

### 6.1172.2.1 addProgressItem()

```
bool Digikam::ProgressManager::addProgressItem (
    ProgressItem *const t,
    ProgressItem *const parent = nullptr ) [static]
```

#### Parameters

<i>t</i>	The process to add on manager.
<i>parent</i>	Specify an already existing item as the parent of this one (can be null).

#### Returns

true if [ProgressItem](#) have been added to manager, else false.

### 6.1172.2.2 createProgressItem() [1/4]

```
ProgressItem * Digikam::ProgressManager::createProgressItem (
    const QString & id,
    const QString & label,
    const QString & status = QString(),
    bool canBeCanceled = true,
    bool hasThumb = false ) [static]
```

#### Parameters

<i>id</i>	Used to identify this operation for cancel and progress info.
<i>label</i>	The text to be displayed by progress handlers
<i>status</i>	Additional text to be displayed for the item.
<i>canBeCanceled</i>	can the user cancel this operation? Cancelling the parent will cancel the children as well (if they can be canceled) and ongoing children prevent parents from finishing.
<i>hasThumb</i>	flag to indicate if progress item has a thumbnail.

#### Returns

The [ProgressItem](#) representing the operation.

**6.1172.2.3 createProgressItem() [2/4]**

```
ProgressItem * Digikam::ProgressManager::createProgressItem (
    const QString & label,
    const QString & status = QString(),
    bool canBeCanceled = true,
    bool hasThumb = false ) [static]
```

**Parameters**

<i>label</i>	The text to be displayed by progress handlers
<i>status</i>	Additional text to be displayed for the item.
<i>canBeCanceled</i>	Can the user cancel this operation? Cancelling the parent will cancel the children as well (if they can be canceled) and ongoing children prevent parents from finishing.
<i>hasThumb</i>	flag to indicate if progress item has a thumbnail.

**Returns**

The [ProgressItem](#) representing the operation.

**6.1172.2.4 createProgressItem() [3/4]**

```
ProgressItem * Digikam::ProgressManager::createProgressItem (
    const QString & parent,
    const QString & id,
    const QString & label,
    const QString & status = QString(),
    bool canBeCanceled = true,
    bool hasThumb = false ) [static]
```

**Parameters**

<i>parent</i>	Specify an already existing item as the parent of this one.
<i>id</i>	Used to identify this operation for cancel and progress info.
<i>label</i>	The text to be displayed by progress handlers
<i>status</i>	Additional text to be displayed for the item.
<i>canBeCanceled</i>	can the user cancel this operation? Cancelling the parent will cancel the children as well (if they can be canceled) and ongoing children prevent parents from finishing.
<i>hasThumb</i>	flag to indicate if progress item has a thumbnail.

**Returns**

The [ProgressItem](#) representing the operation.

**6.1172.2.5 createProgressItem() [4/4]**

```
ProgressItem * Digikam::ProgressManager::createProgressItem (
    ProgressItem *const parent,
```

```

const QString & id,
const QString & label,
const QString & status = QString(),
bool canBeCanceled = true,
bool hasThumb = false ) [static]

```

**Parameters**

<i>parent</i>	Specify an already existing item as the parent of this one.
<i>id</i>	Used to identify this operation for cancel and progress info.
<i>label</i>	The text to be displayed by progress handlers
<i>status</i>	Additional text to be displayed for the item.
<i>canBeCanceled</i>	can the user cancel this operation? Cancelling the parent will cancel the children as well (if they can be canceled) and ongoing children prevent parents from finishing.
<i>hasThumb</i>	flag to indicate if progress item has a thumbnail.

**Returns**

The [ProgressItem](#) representing the operation.

**6.1172.2.6 findItembyId()**

```

ProgressItem * Digikam::ProgressManager::findItembyId (
    const QString & id ) const

```

**Returns**

the progressitem for this

**Parameters**

<i>id</i>	if it exist, else null.
-----------	-------------------------

**6.1172.2.7 getUniqueID()**

```

QString Digikam::ProgressManager::getUniqueID ( )

```

**Returns**

The string with the unique ID number.

**6.1172.2.8 instance()**

```

ProgressManager * Digikam::ProgressManager::instance ( ) [static]

```

**Returns**

The singleton instance of this class.

### 6.1172.2.9 isEmpty()

```
bool Digikam::ProgressManager::isEmpty ( ) const
```

#### Returns

true when there are no more progress items.

### 6.1172.2.10 progressItemAdded

```
void Digikam::ProgressManager::progressItemAdded (
    ProgressItem * ) [signal]
```

#### See also

[ProgressItem::progressItemAdded\(\)](#)

### 6.1172.2.11 progressItemCanceled

```
void Digikam::ProgressManager::progressItemCanceled (
    ProgressItem * ) [signal]
```

#### See also

[ProgressItem::progressItemCanceled\(\)](#)

### 6.1172.2.12 progressItemCompleted

```
void Digikam::ProgressManager::progressItemCompleted (
    ProgressItem * ) [signal]
```

#### See also

[ProgressItem::progressItemCompleted\(\)](#)

### 6.1172.2.13 progressItemLabel

```
void Digikam::ProgressManager::progressItemLabel (
    ProgressItem * ,
    const QString & ) [signal]
```

#### See also

[ProgressItem::progressItemLabel\(\)](#)



### 6.1172.2.14 progressItemProgress

```
void Digikam::ProgressManager::progressItemProgress (
    ProgressItem * ,
    unsigned int ) [signal]
```

See also

[ProgressItem::progressItemProgress\(\)](#)

### 6.1172.2.15 progressItemStatus

```
void Digikam::ProgressManager::progressItemStatus (
    ProgressItem * ,
    const QString & ) [signal]
```

See also

[ProgressItem::progressItemStatus\(\)](#)

### 6.1172.2.16 progressItemThumbnail

```
void Digikam::ProgressManager::progressItemThumbnail (
    ProgressItem * ,
    const QPixmap & ) [signal]
```

See also

[ProgressItem::progressItemThumbnail](#)

### 6.1172.2.17 progressItemUsesBusyIndicator

```
void Digikam::ProgressManager::progressItemUsesBusyIndicator (
    ProgressItem * ,
    bool ) [signal]
```

See also

[ProgressItem::progressItemUsesBusyIndicator](#)

### 6.1172.2.18 singleItem()

```
ProgressItem * Digikam::ProgressManager::singleItem ( ) const
```

Returns

the only top level progressitem when there's only one. Returns 0 if there is no item, or more than one top level item. Since this is used to calculate the overall progress, it will also return 0 if there is an item which uses a busy indicator, since that will invalidate the overall progress.

### 6.1172.2.19 slotStandardCancelHandler

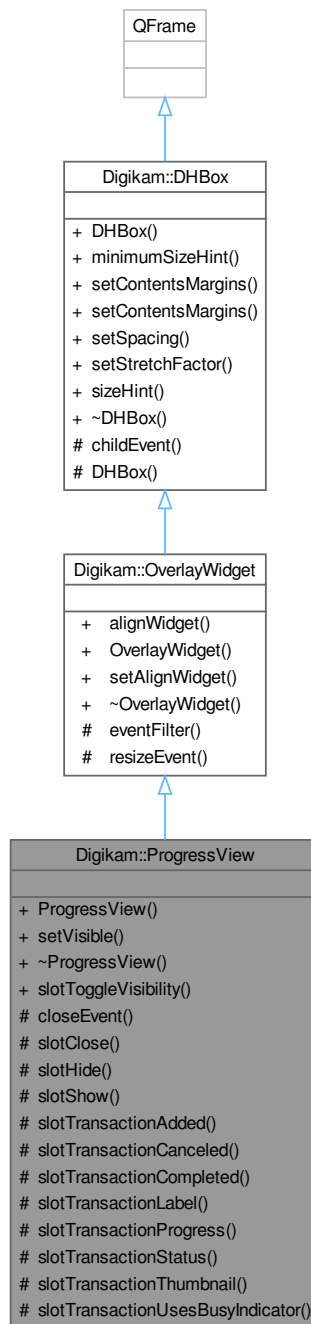
```
void Digikam::ProgressManager::slotStandardCancelHandler (
    ProgressItem * item ) [slot]
```

## Parameters

<i>item</i>	the canceled item.
-------------	--------------------

## 6.1173 Digikam::ProgressView Class Reference

Inheritance diagram for Digikam::ProgressView:



### Public Slots

- void **slotToggleVisibility** ()

### Signals

- void **visibilityChanged** (bool)

### Public Member Functions

- **ProgressView** (QWidget \*const alignWidget, QWidget \*const parent, const QString &name=QString())
- void **setVisible** (bool b) override

### Public Member Functions inherited from [Digikam::OverlayWidget](#)

- QWidget \* **alignWidget** () const
- **OverlayWidget** (QWidget \*const alignWidget, QWidget \*const parent, const QString &name=QString())
- void **setAlignWidget** (QWidget \*const alignWidget)

### Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentMargins** (const QMargins &margins)
- void **setContentMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

### Protected Slots

- void **slotClose** ()
- void **slotHide** ()
- void **slotShow** ()
- void **slotTransactionAdded** ([ProgressItem](#) \*)
- void **slotTransactionCanceled** ([ProgressItem](#) \*)
- void **slotTransactionCompleted** ([ProgressItem](#) \*)
- void **slotTransactionLabel** ([ProgressItem](#) \*, const QString &)
- void **slotTransactionProgress** ([ProgressItem](#) \*, unsigned int progress)
- void **slotTransactionStatus** ([ProgressItem](#) \*, const QString &)
- void **slotTransactionThumbnail** ([ProgressItem](#) \*, const QPixmap &)
- void **slotTransactionUsesBusyIndicator** ([ProgressItem](#) \*, bool)

### Protected Member Functions

- void **closeEvent** (QCloseEvent \*) override

**Protected Member Functions inherited from [Digikam::OverlayWidget](#)**

- bool **eventFilter** (QObject \*o, QEvent \*e) override
- void **resizeEvent** (QResizeEvent \*ev) override

**Protected Member Functions inherited from [Digikam::DHBox](#)**

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1174 Digikam::ProxyClickLineEdit Class Reference

Inheritance diagram for Digikam::ProxyClickLineEdit:



### Signals

- void `leftClicked` ()

### Signals inherited from [Digikam::ProxyLineEdit](#)

- void `signalClearButtonPressed` ()

**Public Member Functions**

- [ProxyClickLineEdit](#) (QWidget \*const parent=nullptr)

**Public Member Functions inherited from [Digikam::ProxyLineEdit](#)**

- [ProxyLineEdit](#) (QWidget \*const parent=nullptr)
- void **setClearButtonShown** (bool show)
- virtual void [setWidget](#) (QWidget \*widget)

**Protected Member Functions**

- void **mouseReleaseEvent** (QMouseEvent \*event) override

**Protected Member Functions inherited from [Digikam::ProxyLineEdit](#)**

- void **changeEvent** (QEvent \*event) override
- void **contextMenuEvent** (QContextMenuEvent \*event) override
- void **dragEnterEvent** (QDragEnterEvent \*event) override
- void **dragLeaveEvent** (QDragLeaveEvent \*e) override
- void **dragMoveEvent** (QDragMoveEvent \*e) override
- void **dropEvent** (QDropEvent \*event) override
- void **focusInEvent** (QFocusEvent \*event) override
- void **focusOutEvent** (QFocusEvent \*event) override
- void **inputMethodEvent** (QInputMethodEvent \*event) override
- void **keyPressEvent** (QKeyEvent \*event) override
- QSize **minimumSizeHint** () const override
- void **mouseDoubleClickEvent** (QMouseEvent \*event) override
- void [mouseMoveEvent](#) (QMouseEvent \*event) override
- void [mousePressEvent](#) (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- void **paintEvent** (QPaintEvent \*event) override
- QSize **sizeHint** () const override

**Additional Inherited Members****Protected Attributes inherited from [Digikam::ProxyLineEdit](#)**

- QVBoxLayout \* **m\_layout** = nullptr
- QWidget \* **m\_widget** = nullptr

**6.1174.1 Constructor & Destructor Documentation****6.1174.1.1 [ProxyClickLineEdit\(\)](#)**

```
Digikam::ProxyClickLineEdit::ProxyClickLineEdit (
    QWidget *const parent = nullptr ) [explicit]
```

A [ProxyLineEdit](#) that emits leftClicked() on mouse press event. Press on the held widget will result in the signal if the widget does not accept() them.

## 6.1175 Digikam::ProxyLineEdit Class Reference

Inheritance diagram for Digikam::ProxyLineEdit:



### Signals

- void **signalClearButtonPressed** ()

### Public Member Functions

- [ProxyLineEdit](#) (QWidget \*const parent=nullptr)
- void **setClearButtonShown** (bool show)
- virtual void [setWidget](#) (QWidget \*widget)

## Protected Member Functions

- void **changeEvent** (QEvent \*event) override
- void **contextMenuEvent** (QContextMenuEvent \*event) override
- void **dragEnterEvent** (QDragEnterEvent \*event) override
- void **dragLeaveEvent** (QDragLeaveEvent \*e) override
- void **dragMoveEvent** (QDragMoveEvent \*e) override
- void **dropEvent** (QDropEvent \*event) override
- void **focusInEvent** (QFocusEvent \*event) override
- void **focusOutEvent** (QFocusEvent \*event) override
- void **inputMethodEvent** (QInputMethodEvent \*event) override
- void **keyPressEvent** (QKeyEvent \*event) override
- QSize **minimumSizeHint** () const override
- void **mouseDoubleClickEvent** (QMouseEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- void **paintEvent** (QPaintEvent \*event) override
- QSize **sizeHint** () const override

## Protected Attributes

- QVBoxLayout \* **m\_layout** = nullptr
- QWidget \* **m\_widget** = nullptr

## 6.1175.1 Constructor & Destructor Documentation

### 6.1175.1.1 ProxyLineEdit()

```
Digikam::ProxyLineEdit::ProxyLineEdit (
    QWidget *const parent = nullptr ) [explicit]
```

This class will not act as a QLineEdit at all, but present another widget (any kind of widget) instead in the space assigned to the QLineEdit. Use this class if you need to pass a QLineEdit but want actually to use a different widget.

## 6.1175.2 Member Function Documentation

### 6.1175.2.1 mouseMoveEvent()

```
void Digikam::ProxyLineEdit::mouseMoveEvent (
    QMouseEvent * event ) [override], [protected]
```

We just re-implement all relevant QWidget event handlers and call the QWidget implementation, not the QLineEdit one.

### 6.1175.2.2 mousePressEvent()

```
void Digikam::ProxyLineEdit::mousePressEvent (
    QMouseEvent * event ) [override], [protected]
```

NOTE: see bug #326718: We need to use QLineEdit parent class with these methods to have clear button working fine.



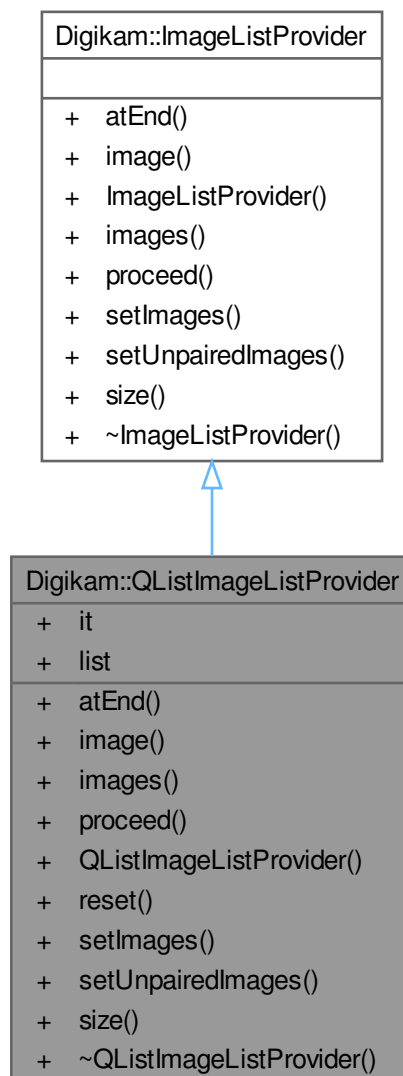
## 6.1175.2.3 setWidget()

```
void Digikam::ProxyLineEdit::setWidget (
    QWidget * widget ) [virtual]
```

After constructing, set the actual widget here

## 6.1176 Digikam::QListImageListProvider Class Reference

Inheritance diagram for Digikam::QListImageListProvider:



## Public Member Functions

- bool [atEnd](#) () const override
- QPair< QImage \*, QString > [image](#) () override
- QList< QPair< QImage \*, QString > > [images](#) () override
- void [proceed](#) (int steps=1) override
- void [reset](#) ()
- void [setImage](#)s (const QList< QPair< QImage \*, QString > > &) override
- void [setUnpairedImages](#) (const QList< QImage \* > &) override
- int [size](#) () const override

## Public Attributes

- QList< QPair< QImage \*, QString > >::const\_iterator [it](#)
- QList< QPair< QImage \*, QString > > [list](#)

### 6.1176.1 Detailed Description

A wrapper implementation for [ImageListProvider](#) if you have a QList of QImages

### 6.1176.2 Member Function Documentation

#### 6.1176.2.1 atEnd()

```
bool Digikam::QListImageListProvider::atEnd ( ) const [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

#### 6.1176.2.2 image()

```
QPair< QImage *, QString > Digikam::QListImageListProvider::image ( ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

#### 6.1176.2.3 images()

```
QList< QPair< QImage *, QString > > Digikam::QListImageListProvider::images ( ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

#### 6.1176.2.4 proceed()

```
void Digikam::QListImageListProvider::proceed (
    int steps = 1 ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

**6.1176.2.5 setImages()**

```
void Digikam::QListImageListProvider::setImages (
    const QList< QPair< QImage *, QString > > & lst ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

**6.1176.2.6 setUnpairedImages()**

```
void Digikam::QListImageListProvider::setUnpairedImages (
    const QList< QImage * > & lst ) [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

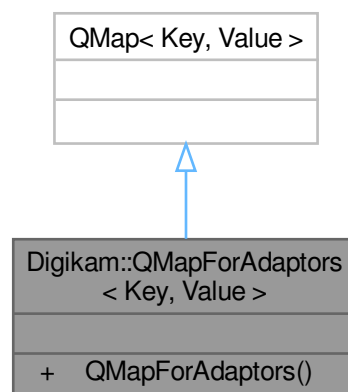
**6.1176.2.7 size()**

```
int Digikam::QListImageListProvider::size ( ) const [override], [virtual]
```

Implements [Digikam::ImageListProvider](#).

**6.1177 Digikam::QMapForAdaptors< Key, Value > Class Template Reference**

Inheritance diagram for Digikam::QMapForAdaptors< Key, Value >:

**Public Types**

- typedef Value **data\_type**
- typedef Key **key\_type**
- typedef std::pair< const Key, Value > **value\_type**

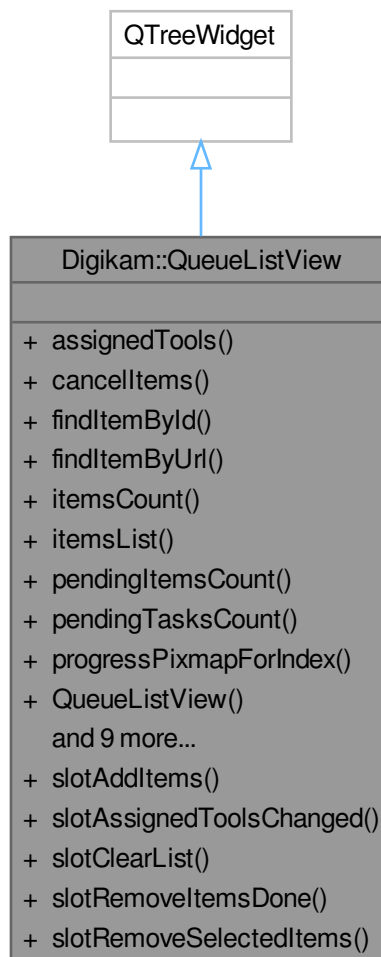
### 6.1177.1 Detailed Description

```
template<typename Key, typename Value>
class Digikam::QMapForAdaptors< Key, Value >
```

Adds the necessary typedefs so that `associative_property_map` accepts a `QMap`, and it can be used as a Boost Property Map.

## 6.1178 Digikam::QueueListView Class Reference

Inheritance diagram for `Digikam::QueueListView`:



### Public Types

- enum `ItemListType` { `Pending` = 0 , `Selected` , `All` }

## Public Slots

- void **slotAddItems** (const [ItemInfoList](#) &)
- void **slotAssignedToolsChanged** (const [AssignedBatchTools](#) &)
- void **slotClearList** ()
- void **slotRemoveItemsDone** ()
- void **slotRemoveSelectedItems** ()

## Signals

- void **signalQueueContentsChanged** ()

## Public Member Functions

- [AssignedBatchTools](#) **assignedTools** () const
- void **cancellItems** ()
- [QueueListViewItem](#) \* **findItemById** (qulonglong id)
- [QueueListViewItem](#) \* **findItemByUrl** (const [QUrl](#) &url)
- int **itemsCount** ()
- [ItemInfoList](#) **itemsList** ([ItemListType](#) type)
- int **pendingItemsCount** ()
- int **pendingTasksCount** ()
- [QPixmap](#) **progressPixmapForIndex** (int index) const
- [QueueListView](#) ([QWidget](#) \*const parent)
- void **reloadThumbs** (const [QUrl](#) &url)
- void **removeItemById** (qulonglong id)
- void **removeItemByInfo** (const [ItemInfo](#) &info)
- void **setAssignedTools** (const [AssignedBatchTools](#) &tools)
- void **setEnabledToolTips** (bool val)
- void **setItemBusy** (qulonglong id)
- void **setSettings** (const [QueueSettings](#) &settings)
- [QueueSettings](#) **settings** () const

## 6.1178.1 Member Enumeration Documentation

### 6.1178.1.1 ItemListType

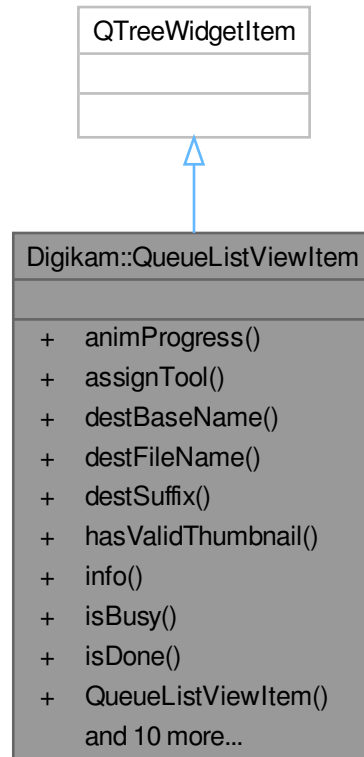
```
enum Digikam::QueueListView::ItemListType
```

#### Enumerator

Pending	Items from the list not yet processed.
Selected	Items from the list selected.
All	All items from the list.

## 6.1179 Digikam::QueueListViewItem Class Reference

Inheritance diagram for Digikam::QueueListViewItem:



### Public Member Functions

- void **animProgress** ()
- void **assignTool** (int index, const [BatchToolSet](#) &set)
- QString **destBaseName** () const
- QString **destFileName** () const
- QString **destSuffix** () const
- bool **hasValidThumbnail** () const
- [ItemInfo](#) **info** () const
- bool **isBusy** () const
- bool **isDone** () const
- [QueueListViewItem](#) ([QueueListView](#) \*const view, const [ItemInfo](#) &info)
- void **reset** ()
- void **setBusy** ()
- void **setCanceled** ()
- void **setDestFileName** (const QString &str)
- void **setDone** ()
- void **setFailed** ()
- void **setInfo** (const [ItemInfo](#) &info)
- void **setThumb** (const QPixmap &pix, bool hasThumb=true)
- void **unassignTool** (int index)

## 6.1180 Digikam::QueueMgrWindow Class Reference

Inheritance diagram for Digikam::QueueMgrWindow:



### Classes

- class [Private](#)

### Public Slots

- void **slotAssignQueueSettings** (const QString &)
- void **slotRun** ()
- void **slotRunAll** ()
- void **slotStop** ()
- void **slotUpdateQueueSettings** (const QString &)

### Signals

- void **signalBqmlsBusy** (bool)
- void **signalWindowHasMoved** ()

### Public Member Functions

- void **addNewQueue** ()
- void **applySettings** ()
- int **currentQueueId** () const
- [DInfoInterface](#) \* **infoface** ([DPluginAction](#) \*const) override
- bool **isBusy** () const
- void **loadItemInfos** (const [ItemInfoList](#) &list, int queueId)
- void **loadItemInfosToCurrentQueue** (const [ItemInfoList](#) &list)
- void **loadItemInfosToNewQueue** (const [ItemInfoList](#) &list)
- bool **queryClose** () override
- [QueuePool](#) \* **queuePool** () const
- QMap< int, QString > **queuesMap** () const
- void **refreshView** ()

### Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- QList< QAction \* > **allActions** () const
- void **cleanupActions** ()
- QString **configGroupName** () const
- void **createFullScreenAction** (const QString &name)
- void **createHelpActions** (const QString &handbookSection, bool coreOptions=true)
- void **createSettingsActions** ()
- void **createSidebarActions** ()
- [DXmlGuiWindow](#) (QWidget \*const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())
- bool **fullScreensActive** () const
- void **readFullScreenSettings** (const KConfigGroup &group)
- virtual void **registerExtraPluginsActions** (QString &)
- void **registerPluginsActions** ()
- void **setConfigGroupName** (const QString &name)
- void **setFullScreenOptions** (int options)
- void **unminimizeAndActivateWindow** ()

### Static Public Member Functions

- static [QueueMgrWindow](#) \* **queueManagerWindow** ()
- static bool **queueManagerWindowCreated** ()



## Static Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- static QAction \* **buildStdAction** (StdActionType type, const QObject \*const recvr, const char \*const slot, QObject \*const parent)
- static QString **configFullScreenHideSideBarsEntry** ()
- static QString **configFullScreenHideStatusBarEntry** ()
- static QString **configFullScreenHideThumbBarEntry** ()
- static QString **configFullScreenHideToolBarsEntry** ()
- static void **restoreWindowSize** (QWindow \*const win, const KConfigGroup &group)
- static void **saveWindowSize** (QWindow \*const win, KConfigGroup &group)
- static void **setGoodDefaultWindowSize** (QWindow \*const win)
- static void **setupIconTheme** ()

## Protected Member Functions

- void **moveEvent** (QMoveEvent \*e) override

## Protected Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- void **closeEvent** (QCloseEvent \*e) override
- void **editKeyboardShortcuts** (KActionCollection \*const extraac=nullptr, const QString &actitle=QString())
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override
- void **keyPressEvent** (QKeyEvent \*e) override
- QAction \* **showMenuBarAction** () const
- virtual void **showSideBars** (bool visible)
- QAction \* **showStatusBarAction** () const
- virtual void **showThumbBar** (bool visible)
- virtual bool **thumbbarVisibility** () const

## Additional Inherited Members

## Protected Slots inherited from [Digikam::DXmlGuiWindow](#)

- bool **slotClose** ()

## Protected Attributes inherited from [Digikam::DXmlGuiWindow](#)

- [DLogoAction](#) \* **m\_animLogo** = nullptr

## 6.1180.1 Member Function Documentation

### 6.1180.1.1 infoface()

```
DInfoInterface * Digikam::QueueMgrWindow::infoIface (
    DPluginAction * const ac ) [override], [virtual]
```

Return the interface instance to access to items information.

Implements [Digikam::DXmlGuiWindow](#).

### 6.1180.1.2 queuesMap()

```
QMap< int, QString > Digikam::QueueMgrWindow::queuesMap ( ) const
```

Return a map of all queues available from pool (index and title).

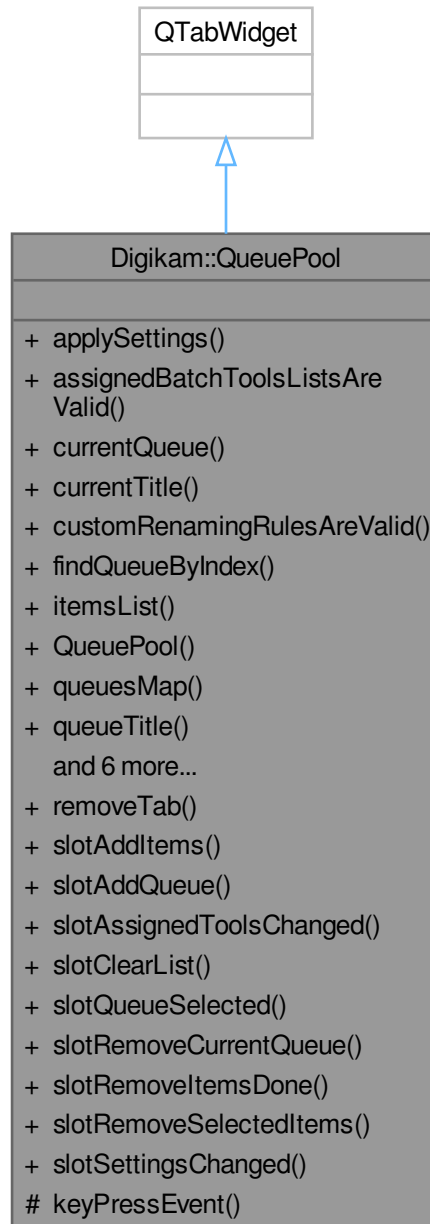
## 6.1181 Digikam::QueueMgrWindow::Private Class Reference

### Public Attributes

- [AssignedListView](#) \* **assignedList** = nullptr
- [BatchToolsFactory](#) \* **batchToolsMgr** = nullptr
- const QString **BOTTOM\_SPLITTER\_CONFIG\_KEY** = QLatin1String("BqmBottomSplitter")
- [SidebarSplitter](#) \* **bottomSplitter** = nullptr
- bool **busy** = false
- QAction \* **clearQueueAction** = nullptr
- QAction \* **clearToolsAction** = nullptr
- QAction \* **contributeAction** = nullptr
- int **currentQueueToProcess** = 0
- QAction \* **donateMoneyAction** = nullptr
- QAction \* **moveDownToolAction** = nullptr
- QAction \* **moveUpToolAction** = nullptr
- QAction \* **newQueueAction** = nullptr
- bool **processingAllQueues** = false
- [QueuePool](#) \* **queuePool** = nullptr
- [QueueSettingsView](#) \* **queueSettingsView** = nullptr
- QAction \* **rawCameraListAction** = nullptr
- QAction \* **removeItemsDoneAction** = nullptr
- QAction \* **removeItemsSelAction** = nullptr
- QAction \* **removeQueueAction** = nullptr
- QAction \* **removeToolAction** = nullptr
- QAction \* **runAction** = nullptr
- QAction \* **runAllAction** = nullptr
- QAction \* **saveQueueAction** = nullptr
- QLabel \* **statusLabel** = nullptr
- [StatusProgressBar](#) \* **statusProgressBar** = nullptr
- QAction \* **stopAction** = nullptr
- [ActionThread](#) \* **thread** = nullptr
- [ToolSettingsView](#) \* **toolSettings** = nullptr
- [ToolsView](#) \* **toolsView** = nullptr
- const QString **TOP\_SPLITTER\_CONFIG\_KEY** = QLatin1String("BqmTopSplitter")
- [SidebarSplitter](#) \* **topSplitter** = nullptr
- const QString **VERTICAL\_SPLITTER\_CONFIG\_KEY** = QLatin1String("BqmVerticalSplitter")
- [SidebarSplitter](#) \* **verticalSplitter** = nullptr

## 6.1182 Digikam::QueuePool Class Reference

Inheritance diagram for Digikam::QueuePool:



### Public Slots

- void **removeTab** (int index)
- void **slotAddItems** (const [ItemInfoList](#) &, int queueId)
- void **slotAddQueue** ()

- void **slotAssignedToolsChanged** (const [AssignedBatchTools](#) &)
- void **slotClearList** ()
- void **slotQueueSelected** (int)
- void **slotRemoveCurrentQueue** ()
- void **slotRemoveItemsDone** ()
- void **slotRemoveSelectedItems** ()
- void **slotSettingsChanged** (const [QueueSettings](#) &)

## Signals

- void **signalItemSelectionChanged** ()
- void **signalQueueContentsChanged** ()
- void **signalQueuePoolChanged** ()
- void **signalQueueSelected** (int id, const [QueueSettings](#) &, const [AssignedBatchTools](#) &)

## Public Member Functions

- void [applySettings](#) ()
- bool **assignedBatchToolsListsAreValid** () const
- [QueueListView](#) \* **currentQueue** () const
- QString **currentTitle** () const
- bool **customRenamingRulesAreValid** () const
- [QueueListView](#) \* **findQueueByIndex** (int index) const
- [QueuePoolItemsList](#) **itemsList** (int index, int type) const
- [QueuePool](#) (QWidget \*const parent)
- QMap< int, QString > **queuesMap** () const
- QString **queueTitle** (int index) const
- bool **saveWorkflow** () const
- void **setBusy** (bool b)
- void **setItemBusy** (qulonglong id)
- int **totalPendingItems** () const
- int **totalPendingTasks** () const

## Protected Member Functions

- void **keyPressEvent** (QKeyEvent \*event) override

## 6.1182.1 Member Function Documentation

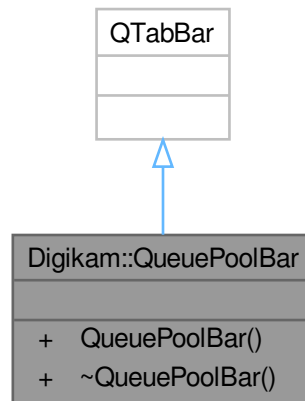
### 6.1182.1.1 [applySettings\(\)](#)

```
void Digikam::QueuePool::applySettings ( )
```

Apply settings changes to all queues settings container when something have been changed in digiKam setup dialog.

## 6.1183 Digikam::QueuePoolBar Class Reference

Inheritance diagram for Digikam::QueuePoolBar:



### Signals

- void **signalTestCanDecode** (const `QDragMoveEvent *`, bool &)

### Public Member Functions

- **QueuePoolBar** (`QWidget *const parent`)

## 6.1184 Digikam::QueueSettings Class Reference

### Public Types

- enum **RawLoadingRule** { `USEEMBEDEDJPEG = 0` , `DEMOSAICING` }
- enum **RenamingRule** { `USEORIGINAL = 0` , `CUSTOMIZE` }

### Public Attributes

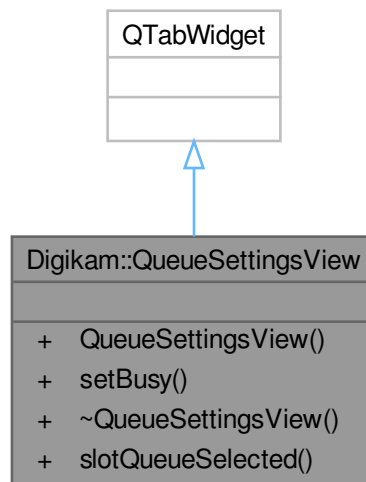
- `FileSaveConflictBox::ConflictRule` **conflictRule** = `FileSaveConflictBox::DIFFNAME`
- bool **exifSetOrientation** = true  
*Setting managed through Metadata control panel.*
- [IOFileSettings](#) **ioFileSettings**
- [DRawDecoderSettings](#) **rawDecodingSettings**
- `RawLoadingRule` **rawLoadingRule** = `DEMOSAICING`
- `QString` **renamingParser**
- `RenamingRule` **renamingRule** = `USEORIGINAL`
- bool **saveAsNewVersion** = true
- bool **useMultiCoreCPU** = false
- bool **useOrgAlbum** = true  
*If true, original file dir will be used to process queue items.*
- `QUrl` **workingUrl**

### 6.1184.1 Detailed Description

This container host all common settings used by a queue, not including assigned batch tools

## 6.1185 Digikam::QueueSettingsView Class Reference

Inheritance diagram for Digikam::QueueSettingsView:



### Public Slots

- void **slotQueueSelected** (int, const [QueueSettings](#) &, const [AssignedBatchTools](#) &)

### Signals

- void **signalSettingsChanged** (const [QueueSettings](#) &)

### Public Member Functions

- **QueueSettingsView** (QWidget \*const parent=nullptr)
- void **setBusy** (bool b)

## 6.1186 Digikam::QueueToolTip Class Reference

Inheritance diagram for Digikam::QueueToolTip:



### Public Member Functions

- `QueueToolTip` ([QueueListView](#) \*const view)
- void `setQueueItem` ([QueueListViewItem](#) \*const item)

### Public Member Functions inherited from [Digikam::DItemToolTip](#)

- `DItemToolTip` (`QWidget` \*const parent=nullptr)

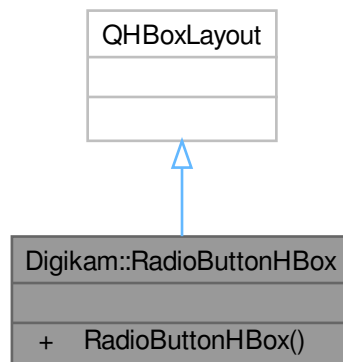
### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DItemToolTip](#)

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- void **resizeEvent** (QResizeEvent \*) override
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()

## 6.1187 Digikam::RadioButtonHBox Class Reference

Inheritance diagram for Digikam::RadioButtonHBox:



### Public Member Functions

- **RadioButtonHBox** (QWidget \*const left, QWidget \*const right, Qt::LayoutDirection dir)



## 6.1188 Digikam::RainDropFilter Class Reference

Inheritance diagram for Digikam::RainDropFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- **RainDropFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, int drop=80, int amount=150, int coeff=30, const QRect &selection=QRect(0, 0, 0, 0))
- **RainDropFilter** (QObject \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.1188.1 Member Function Documentation

### 6.1188.1.1 filterAction()

`FilterAction` Digikam::RainDropFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1188.1.2 filterIdentifier()

`QString` Digikam::RainDropFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1188.1.3 readParameters()

```
void Digikam::RainDropFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1189 Digikam::RandomNumberGenerator Class Reference

### Public Member Functions

- quint32 `currentSeed` ( ) const
- double `number` (double min, double max)
- int `number` (int min, int max)
- [RandomNumberGenerator](#) ( )
- void `reseed` ( )
- void `seed` (quint32 seed)
- quint32 `seedByTime` ( )
- quint32 `seedNonDeterministic` ( )
- bool `yesOrNo` (double p)

### Static Public Member Functions

- static quint32 `nonDeterministicSeed` ( )
- static quint32 `timeSeed` ( )

## 6.1189.1 Detailed Description

This class differs from standard pseudo random number generators (`rand()`) in these points:

- it uses a specified, independently implemented algorithm identical across platforms
- provides access to the used seed
- it can thus guarantee replayable sequences
- it provides convenient seeding of varying quality

## 6.1189.2 Constructor & Destructor Documentation

### 6.1189.2.1 RandomNumberGenerator()

```
Digikam::RandomNumberGenerator::RandomNumberGenerator ( ) [explicit]
```

Constructs a random number generator that is seeded with a constant value. It is recommended to call a `seed` method after construction.

## 6.1189.3 Member Function Documentation

### 6.1189.3.1 currentSeed()

```
quint32 Digikam::RandomNumberGenerator::currentSeed ( ) const
```

Retrieves the current seed. Can be used for `seed(quint32)` to replay the results again.

### 6.1189.3.2 nonDeterministicSeed()

```
quint32 Digikam::RandomNumberGenerator::nonDeterministicSeed ( ) [static]
```

Produces a non-deterministic seed, as used by `seedNonDeterministic()`

### 6.1189.3.3 number() [1/2]

```
double Digikam::RandomNumberGenerator::number (
    double min,
    double max )
```

Returns a random double in the interval `[min, max)` (including min, excluding max) Warning: this method is non re-entrant.

#### 6.1189.3.4 number() [2/2]

```
int Digikam::RandomNumberGenerator::number (
    int min,
    int max )
```

Returns a random integer in the interval [min, max] (including min and max). Warning: this method is non re-entrant.

#### 6.1189.3.5 reseed()

```
void Digikam::RandomNumberGenerator::reseed ( )
```

Seeds the generator again with the [currentSeed\(\)](#). This is not a no-op, rather, the sequence of random numbers starts again from its beginning after each re-seed. Equivalent to `seed(currentSeed())`

#### 6.1189.3.6 seed()

```
void Digikam::RandomNumberGenerator::seed (
    quint32 seed )
```

Seeds the generator with the given value. This is not meant to be called with a constant value, but with a value retrieved from [currentSeed\(\)](#) on a previous run. Across platforms, the same sequence of random numbers will be generated for the same seed.

#### 6.1189.3.7 seedByTime()

```
quint32 Digikam::RandomNumberGenerator::seedByTime ( )
```

Seeds the generator by current time. This is common practice and good enough for most purposes. Returns the new [currentSeed\(\)](#).

#### 6.1189.3.8 seedNonDeterministic()

```
quint32 Digikam::RandomNumberGenerator::seedNonDeterministic ( )
```

Seeds the generator from a non-deterministic random number generator. This is the most secure seeding method. Returns the new [currentSeed\(\)](#).

#### 6.1189.3.9 timeSeed()

```
quint32 Digikam::RandomNumberGenerator::timeSeed ( ) [static]
```

Produces a seed that includes at least the time as source of random data

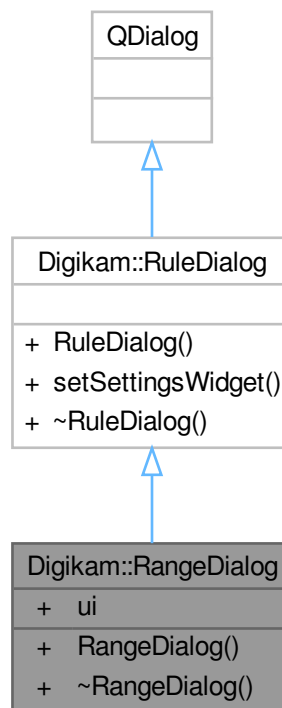
## 6.1189.3.10 yesOrNo()

```
bool Digikam::RandomNumberGenerator::yesOrNo (
    double p )
```

Returns true with a probability of  $p$  (where  $p$  shall be in the interval  $[0, 1)$ ) Warning: this method is non re-entrant.

## 6.1190 Digikam::RangeDialog Class Reference

Inheritance diagram for Digikam::RangeDialog:



## Public Member Functions

- **RangeDialog** ([Rule](#) \*const parent)

Public Member Functions inherited from [Digikam::RuleDialog](#)

- **RuleDialog** ([Rule](#) \*const parent)
- void **setSettingsWidget** (QWidget \*const settingsWidget)

## Public Attributes

- `Ui::RangeModifierDialogWidget *const ui = nullptr`

## 6.1191 Digikam::RangeModifier Class Reference

Inheritance diagram for Digikam::RangeModifier:





## Public Member Functions

- QString [parseOperation](#) ([ParseSettings](#) &settings, const QRegularExpressionMatch &match) override

## Public Member Functions inherited from [Digikam::Modifier](#)

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- [ParseResults](#) **parse** ([ParseSettings](#) &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from [Digikam::Rule](#)

- void **signalTokenTriggered** (const QString &)

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool **addToken** (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void **setUseTokenMenu** (bool value)

## 6.1191.1 Member Function Documentation

### 6.1191.1.1 parseOperation()

```
QString Digikam::RangeModifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [virtual]
```

TODO: describe me

#### Parameters

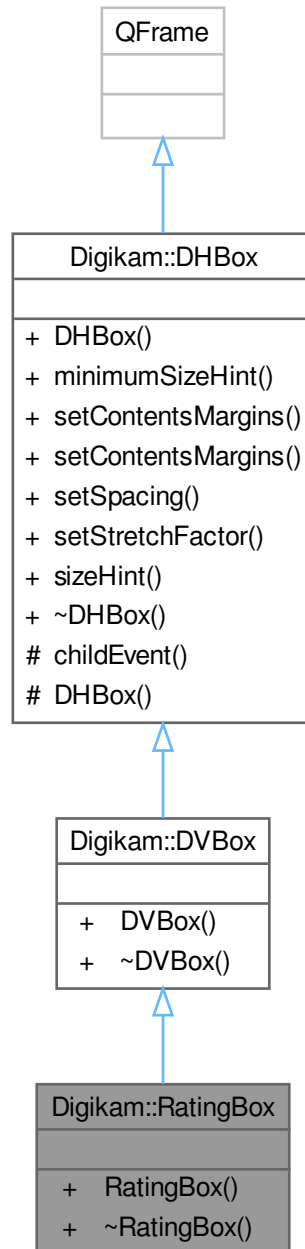
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in Option::parse()

#### Returns

Implements [Digikam::Modifier](#).

## 6.1192 Digikam::RatingBox Class Reference

Inheritance diagram for Digikam::RatingBox:



### Signals

- void **signalRatingChanged** (int)

**Public Member Functions**

- **RatingBox** (QWidget \*const parent)

**Public Member Functions inherited from [Digikam::DVBox](#)**

- **DVBox** (QWidget \*const parent=nullptr)

**Public Member Functions inherited from [Digikam::DHBox](#)**

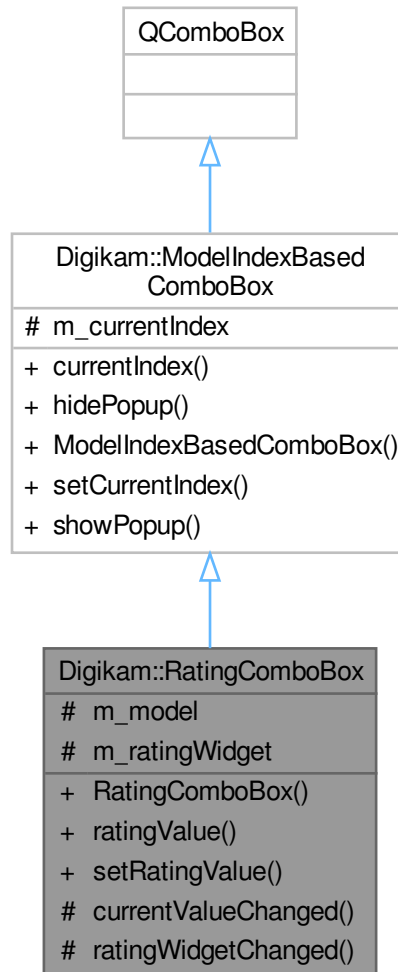
- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentMargins** (const QMargins &margins)
- void **setContentMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

**Additional Inherited Members****Protected Member Functions inherited from [Digikam::DHBox](#)**

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1193 Digikam::RatingComboBox Class Reference

Inheritance diagram for Digikam::RatingComboBox:



### Public Types

- enum `RatingValue` {  
`Null` = -2 , `NoRating` = -1 , `Rating0` = 0 , `Rating1` = 1 ,  
`Rating2` = 2 , `Rating3` = 3 , `Rating4` = 4 , `Rating5` = 5 }

### Signals

- void `ratingValueChanged` (int value)

**Public Member Functions**

- **RatingComboBox** (QWidget \*const parent=nullptr)
- **RatingValue ratingValue** () const
- void **setRatingValue** (RatingValue value)

**Public Member Functions inherited from Digikam::ModelIndexBasedComboBox**

- QModelIndex **currentIndex** () const
- void **hidePopup** () override
- **ModelIndexBasedComboBox** (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

**Protected Slots**

- void **currentValueChanged** (const QModelIndex &current, const QModelIndex &previous)
- void **ratingWidgetChanged** (int)

**Protected Attributes**

- **RatingComboBoxModel** \* **m\_model** = nullptr
- **RatingComboBoxWidget** \* **m\_ratingWidget** = nullptr

**Protected Attributes inherited from Digikam::ModelIndexBasedComboBox**

- QPersistentModelIndex **m\_currentIndex**

**6.1193.1 Member Enumeration Documentation****6.1193.1.1 RatingValue**

enum `Digikam::RatingComboBox::RatingValue`

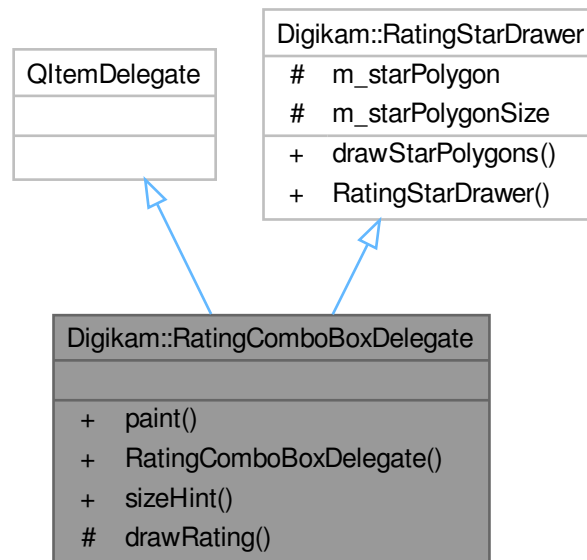
An advanced widget for entering a rating, including support for Null and NoRating values

**Enumerator**

Null	The rating value. All values except Null correspond to the integers used by the database.
------	---

## 6.1194 Digikam::RatingComboBoxDelegate Class Reference

Inheritance diagram for Digikam::RatingComboBoxDelegate:



### Public Member Functions

- void **paint** (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- **RatingComboBoxDelegate** (QObject \*const parent=nullptr)
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &index) const override

### Public Member Functions inherited from [Digikam::RatingStarDrawer](#)

- QRect **drawStarPolygons** (QPainter \*p, int numberOfStars) const

### Protected Member Functions

- void **drawRating** (QPainter \*painter, const QRect &rect, int rating, bool selectable) const

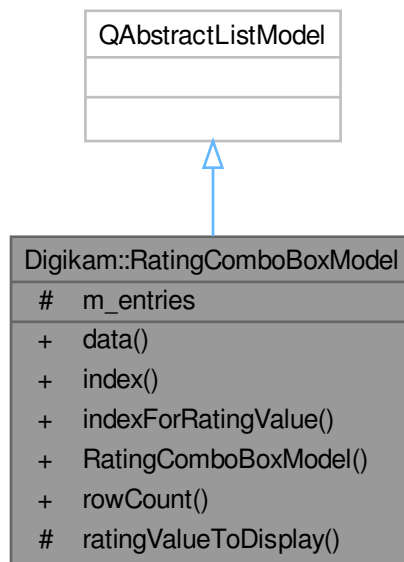
### Additional Inherited Members

### Protected Attributes inherited from [Digikam::RatingStarDrawer](#)

- QPolygon **m\_starPolygon** = [RatingWidget::starPolygon](#)()
- QSize **m\_starPolygonSize** = QSize(15, 15)

## 6.1195 Digikam::RatingComboBoxModel Class Reference

Inheritance diagram for Digikam::RatingComboBoxModel:



### Public Types

- enum **CustomRoles** { **RatingRole** = Qt::UserRole }

### Public Member Functions

- QVariant **data** (const QModelIndex &index, int role) const override
- QModelIndex **index** (int row, int column=0, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex **indexForRatingValue** ([RatingComboBox::RatingValue](#) value) const
- **RatingComboBoxModel** (QObject \*const parent=nullptr)
- int **rowCount** (const QModelIndex &parent) const override

### Protected Member Functions

- QVariant **ratingValueToDisplay** ([RatingComboBox::RatingValue](#) value) const

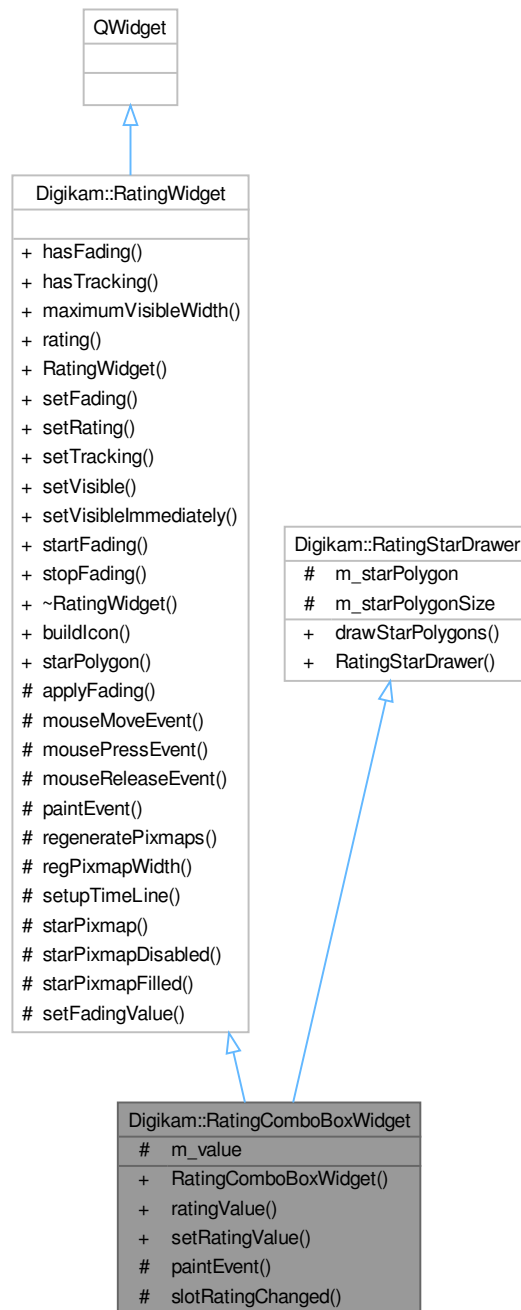
### Protected Attributes

- QList< [RatingComboBox::RatingValue](#) > **m\_entries**



## 6.1196 Digikam::RatingComboBoxWidget Class Reference

Inheritance diagram for Digikam::RatingComboBoxWidget:



### Signals

- void **ratingValueChanged** (int value)

## Signals inherited from [Digikam::RatingWidget](#)

- void **signalRatingChanged** (int)
- void **signalRatingModified** (int)

*Not managed by tracking properties.*

## Public Member Functions

- [RatingComboBoxWidget](#) (QWidget \*const parent=nullptr)
- [RatingComboBox::RatingValue](#) **ratingValue** () const
- void **setRatingValue** ([RatingComboBox::RatingValue](#) value)

## Public Member Functions inherited from [Digikam::RatingWidget](#)

- bool **hasFading** () const
- bool **hasTracking** () const
- int **maximumVisibleWidth** () const
- int **rating** () const
- [RatingWidget](#) (QWidget \*const parent)
- void **setFading** (bool fading)
- void **setRating** (int val)
- void **setTracking** (bool tracking)
- void **setVisible** (bool visible) override
- void **setVisibleImmediately** ()
- void **startFading** ()
- void **stopFading** ()

## Public Member Functions inherited from [Digikam::RatingStarDrawer](#)

- QRect **drawStarPolygons** (QPainter \*p, int numberOfStars) const

## Protected Slots

- void **slotRatingChanged** (int)

## Protected Slots inherited from [Digikam::RatingWidget](#)

- void **setFadingValue** (int value)

## Protected Member Functions

- void **paintEvent** (QPaintEvent \*) override

## Protected Member Functions inherited from [Digikam::RatingWidget](#)

- void **applyFading** (QPixmap &pix)
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **regeneratePixmap** ()
- int **regPixmapWidth** () const
- void **setupTimeLine** ()
- QPixmap **starPixmap** () const
- QPixmap **starPixmapDisabled** () const
- QPixmap **starPixmapFilled** () const

## Protected Attributes

- [RatingComboBox::RatingValue](#) **m\_value** = [RatingComboBox::Null](#)

## Protected Attributes inherited from [Digikam::RatingStarDrawer](#)

- QPolygon **m\_starPolygon** = [RatingWidget::starPolygon](#)()
- QSize **m\_starPolygonSize** = QSize(15, 15)

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::RatingWidget](#)

- static QIcon **buildIcon** (int rate, int size)
- static QPolygon **starPolygon** ()

## 6.1196.1 Constructor & Destructor Documentation

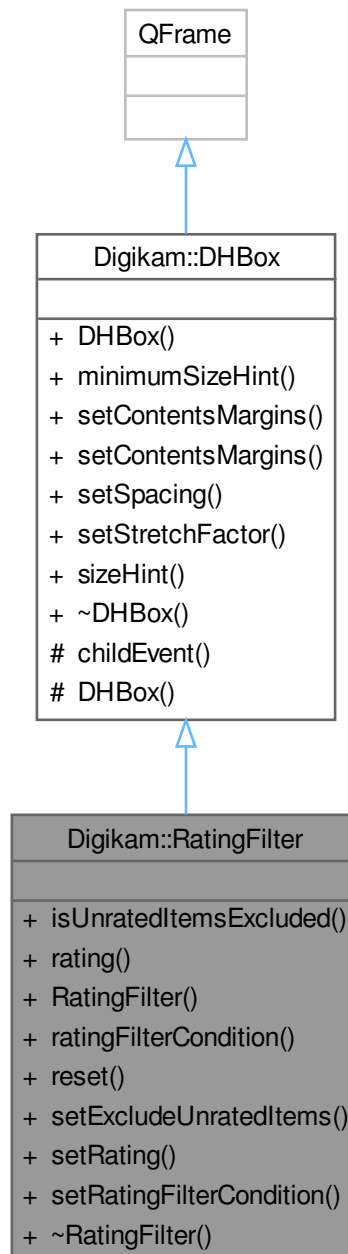
### 6.1196.1.1 [RatingComboBoxWidget](#)()

```
Digikam::RatingComboBoxWidget::RatingComboBoxWidget (  
    QWidget *const parent = nullptr ) [explicit]
```

Internal sub-classing the classic [RatingWidget](#), this provides support for the Null and NoRating states.

## 6.1197 Digikam::RatingFilter Class Reference

Inheritance diagram for Digikam::RatingFilter:



### Signals

- void **signalRatingFilterChanged** (int, [ItemFilterSettings::RatingCondition](#), bool)

### Public Member Functions

- bool **isUnratedItemsExcluded** ()
- int **rating** () const
- **RatingFilter** (QWidget \*const parent)
- [ItemFilterSettings::RatingCondition](#) **ratingFilterCondition** ()
- void **reset** ()
- void **setExcludeUnratedItems** (bool excluded)
- void **setRating** (int val)
- void **setRatingFilterCondition** ([ItemFilterSettings::RatingCondition](#) cond)

### Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1198 Digikam::RatingFilterWidget Class Reference

Inheritance diagram for Digikam::RatingFilterWidget:



### Signals

- void **signalRatingFilterChanged** (int, [ItemFilterSettings::RatingCondition](#), bool)

## Signals inherited from [Digikam::RatingWidget](#)

- void **signalRatingChanged** (int)
- void **signalRatingModified** (int)

*Not managed by tracking properties.*

## Public Member Functions

- bool **isUnratedItemsExcluded** ()
- [ItemFilterSettings::RatingCondition](#) **ratingFilterCondition** ()
- **RatingFilterWidget** (QWidget \*const parent)
- void **setExcludeUnratedItems** (bool excluded)
- void **setRatingFilterCondition** ([ItemFilterSettings::RatingCondition](#) cond)

## Public Member Functions inherited from [Digikam::RatingWidget](#)

- bool **hasFading** () const
- bool **hasTracking** () const
- int **maximumVisibleWidth** () const
- int **rating** () const
- **RatingWidget** (QWidget \*const parent)
- void **setFading** (bool fading)
- void **setRating** (int val)
- void **setTracking** (bool tracking)
- void **setVisible** (bool visible) override
- void **setVisibleImmediately** ()
- void **startFading** ()
- void **stopFading** ()

## Protected Member Functions

- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override

## Protected Member Functions inherited from [Digikam::RatingWidget](#)

- void **applyFading** (QPixmap &pix)
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **regeneratePixmap** ()
- int **regPixmapWidth** () const
- void **setupTimeLine** ()
- QPixmap **starPixmap** () const
- QPixmap **starPixmapDisabled** () const
- QPixmap **starPixmapFilled** () const

### Additional Inherited Members

### Static Public Member Functions inherited from [Digikam::RatingWidget](#)

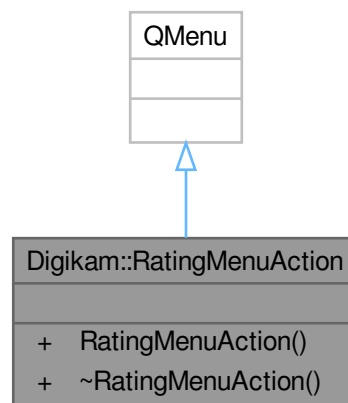
- static QIcon **buildIcon** (int rate, int size)
- static QPolygon **starPolygon** ()

### Protected Slots inherited from [Digikam::RatingWidget](#)

- void **setFadingValue** (int value)

## 6.1199 Digikam::RatingMenuAction Class Reference

Inheritance diagram for Digikam::RatingMenuAction:



### Signals

- void **signalRatingChanged** (int)

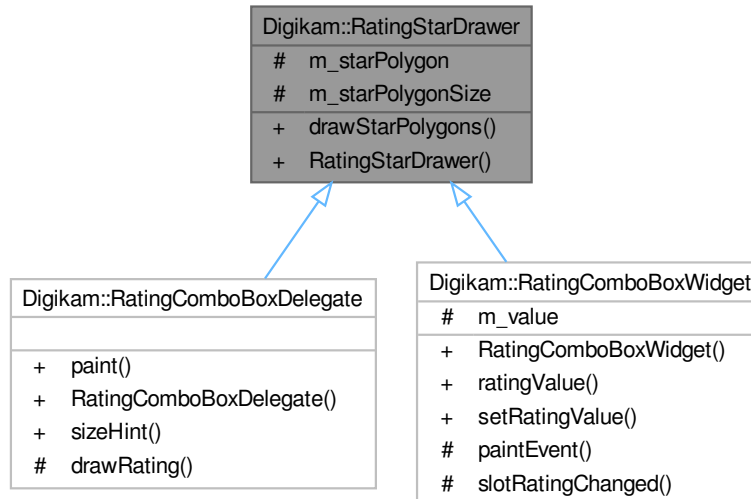
### Public Member Functions

- **RatingMenuAction** (QMenu \*const parent=nullptr)



## 6.1200 Digikam::RatingStarDrawer Class Reference

Inheritance diagram for Digikam::RatingStarDrawer:



### Public Member Functions

- QRect **drawStarPolygons** (QPainter \*p, int numberOfStars) const

### Protected Attributes

- QPolygon **m\_starPolygon** = [RatingWidget::starPolygon\(\)](#)
- QSize **m\_starPolygonSize** = QSize(15, 15)

## 6.1201 Digikam::RatingWidget Class Reference

Inheritance diagram for Digikam::RatingWidget:



### Signals

- void **signalRatingChanged** (int)
- void **signalRatingModified** (int)

*Not managed by tracking properties.*

## Public Member Functions

- bool **hasFading** () const
- bool **hasTracking** () const
- int **maximumVisibleWidth** () const
- int **rating** () const
- **RatingWidget** (QWidget \*const parent)
- void **setFading** (bool fading)
- void **setRating** (int val)
- void **setTracking** (bool tracking)
- void **setVisible** (bool visible) override
- void **setVisibleImmediately** ()
- void **startFading** ()
- void **stopFading** ()

## Static Public Member Functions

- static QIcon **buildIcon** (int rate, int size)
- static QPolygon **starPolygon** ()

## Protected Slots

- void **setFadingValue** (int value)

## Protected Member Functions

- void **applyFading** (QPixmap &pix)
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **regeneratePixmaps** ()
- int **regPixmapWidth** () const
- void **setupTimeLine** ()
- QPixmap **starPixmap** () const
- QPixmap **starPixmapDisabled** () const
- QPixmap **starPixmapFilled** () const

## 6.1201.1 Member Function Documentation

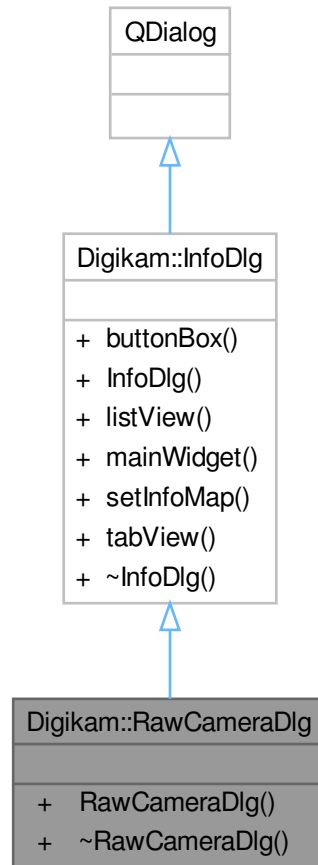
### 6.1201.1.1 starPolygon()

QPolygon Digikam::RatingWidget::starPolygon ( ) [static]

Pre-computed star polygon for a 15x15 pixmap.

## 6.1202 Digikam::RawCameraDlg Class Reference

Inheritance diagram for Digikam::RawCameraDlg:



### Public Member Functions

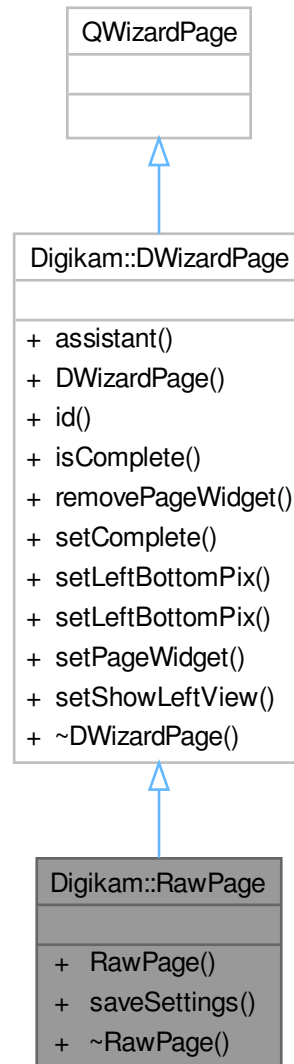
- **RawCameraDlg** (QWidget \*const parent)

### Public Member Functions inherited from [Digikam::InfoDlg](#)

- QDialogButtonBox \* **buttonBox** () const
- **InfoDlg** (QWidget \*const parent)
- QTreeWidget \* **listView** () const
- QWidget \* **mainWidget** () const
- virtual void **setInfoMap** (const QMap< QString, QString > &list)
- QTabWidget \* **tabView** () const

## 6.1203 Digikam::RawPage Class Reference

Inheritance diagram for Digikam::RawPage:



### Public Member Functions

- **RawPage** (QWizard \*const dlg)
- void **saveSettings** ()

### Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)

- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.1204 Digikam::RawProcessingFilter Class Reference

Inheritance diagram for Digikam::RawProcessingFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- [RawProcessingFilter](#) (const [DRawDecoding](#) &settings, [DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const [QString](#) &name=QString())
- [RawProcessingFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [DRawDecoding](#) &settings, const [QString](#) &name=QString())
- [RawProcessingFilter](#) ([QObject](#) \*const parent=nullptr)
- void [readParameters](#) (const [FilterAction](#) &action) override
- void [setObserver](#) ([DImgLoaderObserver](#) \*const observer, int progressBegin, int progressEnd)
- void [setOutputProfile](#) (const [IccProfile](#) &profile)
- void [setSettings](#) (const [DRawDecoding](#) &settings)
- [DRawDecoding](#) **settings** () const

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=QString())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=QString())
- const [QString](#) & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static [QString](#) **DisplayableName** ()
- static [QString](#) **FilterIdentifier** ()
- static [QList](#)< int > **SupportedVersions** ()



**Protected Member Functions**

- bool **continueQuery** () const
- void **filterImage** () override
- void **postProgress** (int)

**Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)**

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

**Protected Member Functions inherited from [Digikam::DynamicThread](#)**

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

**Protected Attributes**

- **IccProfile** **m\_customOutputProfile**
- **DImgLoaderObserver** \* **m\_observer** = nullptr
- **DRawDecoding** **m\_settings**

**Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)**

- **DImg** **m\_destImage**
- **DImgThreadedFilter** \* **m\_master** = nullptr
- QString **m\_name**
- **DImg** **m\_orgImage**
- int **m\_progressBegin** = 0
- int **m\_progressCurrent** = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* **m\_slave** = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

### Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

### Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

### Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## 6.1204.1 Detailed Description

This is a special filter. It implements RAW post processing. Additionally, it provides some facilities for use from the [DImg](#) Raw loader.

The original image shall come from RawEngine without further modification.

## 6.1204.2 Constructor & Destructor Documentation

### 6.1204.2.1 RawProcessingFilter() [1/3]

```
Digikam::RawProcessingFilter::RawProcessingFilter (
    QObject *const parent = nullptr ) [explicit]
```

Default constructor. You need to call [setSettings\(\)](#) and [setOriginalImage\(\)](#) before starting the filter.

### 6.1204.2.2 RawProcessingFilter() [2/3]

```
Digikam::RawProcessingFilter::RawProcessingFilter (
    DImg *const orgImage,
    QObject *const parent,
    const DRawDecoding & settings,
    const QString & name = QString() )
```

Traditional constructor

### 6.1204.2.3 RawProcessingFilter() [3/3]

```
Digikam::RawProcessingFilter::RawProcessingFilter (
    const DRawDecoding & settings,
    DImgThreadedFilter *const master,
    const DImg & orgImage,
    const DImg & destImage,
    int progressBegin = 0,
    int progressEnd = 100,
    const QString & name = QString() )
```

For use with a master filter. Computation is started immediately.

## 6.1204.3 Member Function Documentation

### 6.1204.3.1 filterAction()

```
FilterAction Digikam::RawProcessingFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1204.3.2 filterIdentifier()

```
QString Digikam::RawProcessingFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1204.3.3 filterImage()

```
void Digikam::RawProcessingFilter::filterImage ( ) [override], [protected], [virtual]
```

Main image filter method. Override in subclass.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1204.3.4 readParameters()

```
void Digikam::RawProcessingFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

### 6.1204.3.5 setObserver()

```
void Digikam::RawProcessingFilter::setObserver (
    DImgLoaderObserver *const observer,
    int progressBegin,
    int progressEnd )
```

Normally, filters post progress and are cancelled by [DynamicThread](#) facilities. Here, as an alternative, a [DImgLoaderObserver](#) is set. Its `continueQuery` is called and progress is posted in the given interval.

### 6.1204.3.6 setOutputProfile()

```
void Digikam::RawProcessingFilter::setOutputProfile (
    const IccProfile & profile )
```

As additional and first post-processing step, convert the image's color space to the specified profile.

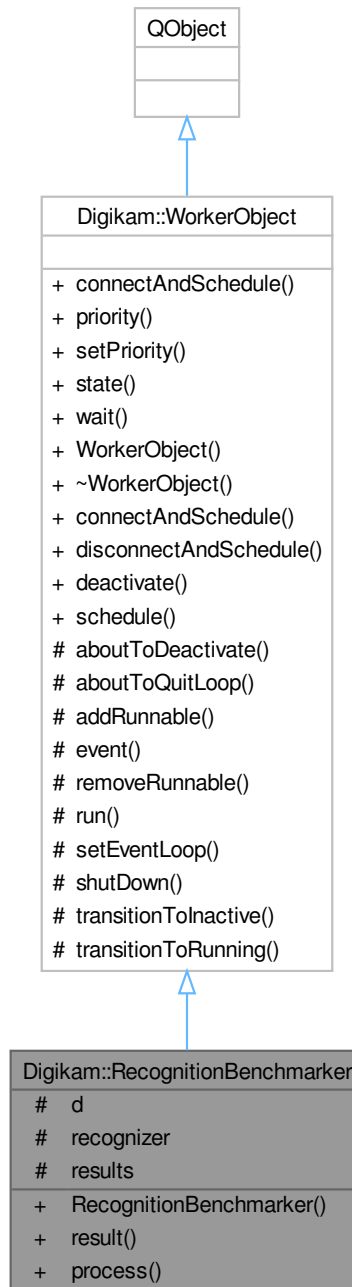
### 6.1204.3.7 setSettings()

```
void Digikam::RawProcessingFilter::setSettings (
    const DRawDecoding & settings )
```

Set the raw decoding settings. The post processing is carried out here, the libraw settings are needed to construct the [FilterAction](#).

## 6.1205 Digikam::RecognitionBenchmarker Class Reference

Inheritance diagram for Digikam::RecognitionBenchmarker:



### Classes

- class [Statistics](#)

### Public Slots

- void **process** (const FacePipelineExtendedPackage::Ptr &package)

### Public Slots inherited from [Digikam::WorkerObject](#)

- void **deactivate** ([DeactivatingMode](#) mode=[FlushSignals](#))
- void **schedule** ()

### Signals

- void **processed** (const FacePipelineExtendedPackage::Ptr &package)

### Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

### Public Member Functions

- **RecognitionBenchmark** ([FacePipeline::Private](#) \*const dd)
- QString **result** () const

### Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const char \*method, Qt::↔ ConnectionType type=Qt::AutoConnection) const
- QThread::Priority **priority** () const
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

### Protected Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- [FacialRecognitionWrapper](#) **recognizer**
- QMap< int, [Statistics](#) > **results**

### Additional Inherited Members

### Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

### 6.1205.1 Member Function Documentation

#### 6.1205.1.1 result()

```
QString Digikam::RecognitionBenchmarker::result ( ) const
```

NOTE: Bench performance code. No need i18n here

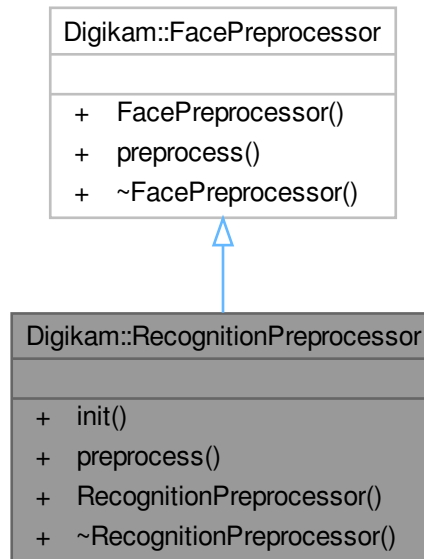
## 6.1206 Digikam::RecognitionBenchmarker::Statistics Class Reference

### Public Attributes

- int **correctlyRecognized** = 0
- int **knownFaces** = 0

## 6.1207 Digikam::RecognitionPreprocessor Class Reference

Inheritance diagram for Digikam::RecognitionPreprocessor:



### Public Member Functions

- void **init** (PreprocessorSelection mode)
- cv::Mat **preprocess** (const cv::Mat &image) const override

### 6.1207.1 Member Function Documentation

#### 6.1207.1.1 preprocess()

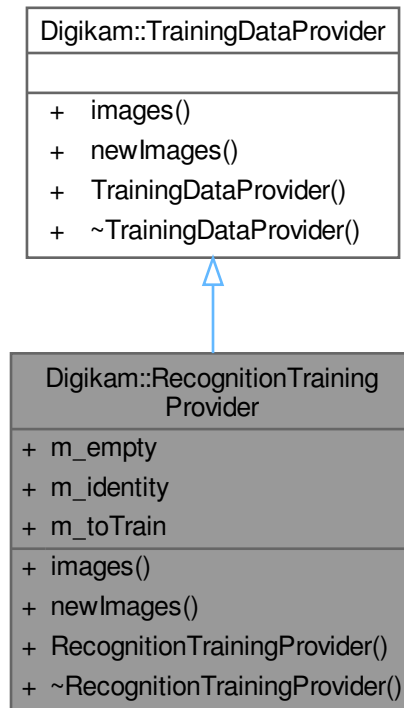
```
cv::Mat Digikam::RecognitionPreprocessor::preprocess (
    const cv::Mat & image ) const [override], [virtual]
```

Implements [Digikam::FacePreprocessor](#).



## 6.1208 Digikam::RecognitionTrainingProvider Class Reference

Inheritance diagram for Digikam::RecognitionTrainingProvider:



### Public Member Functions

- `ImageListProvider * images` (const `Identity &`) override
- `ImageListProvider * newImages` (const `Identity &id`) override
- **`RecognitionTrainingProvider`** (const `Identity &identity`, const `QList< QPair< QImage *, QString > >` &`newImages`)

### Public Attributes

- `QListImageListProvider m_empty`
- `Identity m_identity`
- `QListImageListProvider m_toTrain`

### 6.1208.1 Detailed Description

A simple QImage training data container used by `RecognitionDatabase::train(Identity, QImage, QString)`.

## 6.1208.2 Member Function Documentation

### 6.1208.2.1 images()

```
ImageListProvider * Digikam::RecognitionTrainingProvider::images (
    const Identity & identity ) [override], [virtual]
```

Provides all images known for the given identity. Ownership of the returned object stays with the [TrainingDataProvider](#).

Implements [Digikam::TrainingDataProvider](#).

### 6.1208.2.2 newImages()

```
ImageListProvider * Digikam::RecognitionTrainingProvider::newImages (
    const Identity & identity ) [override], [virtual]
```

Provides those images for the given identity that have not yet been supplied for training. Ownership of the returned object stays with the [TrainingDataProvider](#).

Implements [Digikam::TrainingDataProvider](#).

## 6.1209 Digikam::RecognitionTrainingUpdateQueue Class Reference

### Public Member Functions

- QString **endSignal** ()
- QString **front** ()
- QString **pop\_front** ()
- void **push** (const QString &hash)
- void **registerReaderThread** (const QThread \*thread)
- void **unregisterReaderThread** (const QThread \*thread)

## 6.1210 Digikam::RecognitionWorker Class Reference

Inheritance diagram for Digikam::RecognitionWorker:



### Public Slots

- void [process](#) (const FacePipelineExtendedPackage::Ptr &package)

- void **setAccuracyAndModel** (int detectAccuracy, [FaceScanSettings::FaceDetectionModel](#) detectModel, [FaceScanSettings::FaceDetectionSize](#) detectSize, int recognizeAccuracy, [FaceScanSettings::FaceRecognitionModel](#) recognizeModel)
- void **setThreshold** (int threshold, bool)

### Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

### Signals

- void **processed** (const [FacePipelineExtendedPackage::Ptr](#) &package)

### Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

### Public Member Functions

- **RecognitionWorker** ([FacePipeline::Private](#) \*const dd)

### Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const char \*method, [Qt::](#)↔[ConnectionType](#) type=[Qt::AutoConnection](#)) const
- [QThread::Priority](#) **priority** () const
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

### Protected Member Functions

- void [aboutToDeactivate](#) () override

### Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** ([WorkerObjectRunnable](#) \*loop)
- bool **event** ([QEvent](#) \*e) override
- void **removeRunnable** ([WorkerObjectRunnable](#) \*loop)
- void **run** ()
- void **setEventLoop** ([QEventLoop](#) \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

### Protected Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- [FaceltemRetriever](#) **imageRetriever**
- [FacialRecognitionWrapper](#) **recognizer**

### Additional Inherited Members

### Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum **State** { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

### Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## 6.1210.1 Member Function Documentation

### 6.1210.1.1 [aboutToDeactivate\(\)](#)

```
void Digikam::RecognitionWorker::aboutToDeactivate ( ) [override], [protected], [virtual]
```

Called from [deactivate\(\)](#), typically from a different thread than the worker thread, possibly the UI thread. You can stop any extra controlled threads here. Immediately afterwards, an event will be sent to the working thread which will cause the event loop to quit. ([aboutToQuitLoop\(\)](#))

Reimplemented from [Digikam::WorkerObject](#).

### 6.1210.1.2 [process](#)

```
void Digikam::RecognitionWorker::process (
    const FacePipelineExtendedPackage::Ptr & package ) [slot]
```

TODO: investigate this method.

## 6.1211 Digikam::RedEye::RegressionTree Struct Reference

### Public Member Functions

- unsigned long **num\_leaves** () const
- const std::vector< float > & **operator()** (const std::vector< float > &feature\_pixel\_values, unsigned long &i) const

## Public Attributes

- `std::vector< std::vector< float > >` **leaf\_values**
- `std::vector< SplitFeature >` **splits**

## 6.1211.1 Member Function Documentation

### 6.1211.1.1 `operator()`

```
const std::vector< float > & Digikam::RedEye::RegressionTree::operator() (
    const std::vector< float > & feature_pixel_values,
    unsigned long & i ) const
```

requires

- All the index values in splits are less than

#### Parameters

<code>feature_pixel_values</code>	size.
-----------------------------------	-------

- `leaf_values.size()` is a power of 2. (i.e. we require a tree with all the levels fully filled out.
- `leaf_values.size() == splits.size()+1` (i.e. there needs to be the right number of leaves given the number of splits in the tree) ensures runs through the tree and returns the vector at the leaf we end up in.

#### Parameters

<code>i</code>	egal the selected leaf node index.
----------------	------------------------------------

## 6.1212 Digikam::RedEye::ShapePredictor Class Reference

### Public Member Functions

- unsigned long **num\_features** () const
- unsigned long **num\_parts** () const
- [FullObjectDetection](#) **operator()** (const cv::Mat &img, const cv::Rect &rect) const

### Public Attributes

- `std::vector< std::vector< unsigned long > >` **anchor\_idx**
- `std::vector< std::vector< std::vector< float > > >` **deltas**
- `std::vector< std::vector< RedEye::RegressionTree > >` **forests**
- `std::vector< float >` **initial\_shape**

## 6.1213 Digikam::RedEye::SplitFeature Struct Reference

### Public Attributes

- quint64 **idx1** = 0
- quint64 **idx2** = 0
- float **thresh** = 0.0F

## 6.1214 Digikam::RedEyeCorrectionContainer Class Reference

### Public Member Functions

- bool **isDefault** () const
- bool **operator==** (const [RedEyeCorrectionContainer](#) &other) const
- void **writeToFilterAction** ([FilterAction](#) &action, const QString &prefix=QString()) const

### Static Public Member Functions

- static [RedEyeCorrectionContainer](#) **fromFilterAction** (const [FilterAction](#) &action, const QString &prefix=QString())

### Public Attributes

- double **m\_redToAvgRatio** = 2.1

## 6.1215 Digikam::RedEyeCorrectionFilter Class Reference

Inheritance diagram for Digikam::RedEyeCorrectionFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override



- [RedEyeCorrectionFilter](#) (const [RedEyeCorrectionContainer](#) &settings, [DImgThreadedFilter](#) \*const parent←  
Filter, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100)
- **RedEyeCorrectionFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [RedEyeCorrectionContainer](#)  
&settings=[RedEyeCorrectionContainer](#)())
- **RedEyeCorrectionFilter** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0,  
int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.1215.1 Constructor & Destructor Documentation

### 6.1215.1.1 RedEyeCorrectionFilter()

```
Digikam::RedEyeCorrectionFilter::RedEyeCorrectionFilter (
    const RedEyeCorrectionContainer & settings,
    DImgThreadedFilter *const parentFilter,
    const DImg & orgImage,
    const DImg & destImage,
    int progressBegin = 0,
    int progressEnd = 100 ) [explicit]
```

Constructor for slave mode: execute immediately in current thread with specified master filter

## 6.1215.2 Member Function Documentation

### 6.1215.2.1 filterAction()

```
FilterAction Digikam::RedEyeCorrectionFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1215.2.2 filterIdentifier()

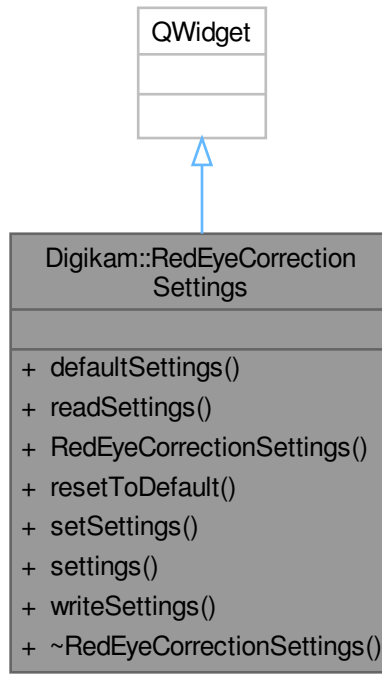
```
QString Digikam::RedEyeCorrectionFilter::filterIdentifier ( ) const [inline], [override],
[virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

## 6.1216 Digikam::RedEyeCorrectionSettings Class Reference

Inheritance diagram for Digikam::RedEyeCorrectionSettings:



### Signals

- void **signalSettingsChanged** ()

### Public Member Functions

- [RedEyeCorrectionContainer](#) **defaultSettings** () const
- void **readSettings** (const KConfigGroup &group)
- [RedEyeCorrectionSettings](#) (QWidget \*const parent=nullptr)
- void **resetToDefault** ()
- void **setSettings** (const [RedEyeCorrectionContainer](#) &settings)
- [RedEyeCorrectionContainer](#) **settings** () const
- void **writeSettings** (KConfigGroup &group)

## 6.1217 Digikam::RefocusFilter Class Reference

Inheritance diagram for Digikam::RefocusFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- void [readParameters](#) (const [FilterAction](#) &action) override
- **RefocusFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, int matrixSize=5, double radius=0.9, double gauss=0.0, double correlation=0.5, double noise=0.01)
- **RefocusFilter** (QObject \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static int **maxMatrixSize** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.1217.1 Member Function Documentation

### 6.1217.1.1 filterAction()

`FilterAction` Digikam::RefocusFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1217.1.2 filterIdentifier()

`QString` Digikam::RefocusFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1217.1.3 readParameters()

```
void Digikam::RefocusFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1218 Digikam::RefocusMatrix Class Reference

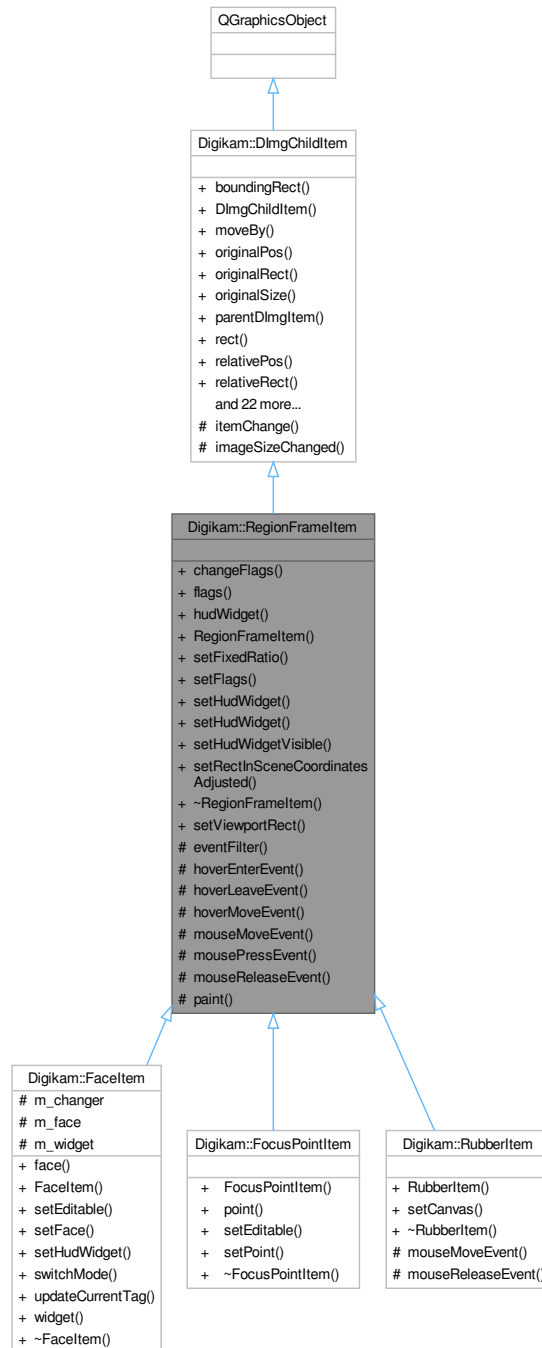
### Static Public Member Functions

- static double `c_mat_elt` (const [CMat](#) \*const mat, const int col, const int row)
- static [CMat](#) \* `compute_g_matrix` (const [CMat](#) \*const convolution, const int m, const double gamma, const double noise\_factor, const double musq, const bool symmetric)
- static void `convolve_star_mat` (const [CMat](#) \*const result, const [CMat](#) \*const mata, const [CMat](#) \*const matb)
- static void `fill_matrix` ([CMat](#) \*const matrix, const int m, double f(const int, const int, const double), const double fun\_arg)
- static void `fill_matrix2` ([CMat](#) \*const matrix, const int m, double f(const int, const int, const double, const double), const double fun\_arg1, const double fun\_arg2)
- static void `finish_and_free_matrix` ([Mat](#) \*const mat)
- static void `finish_c_mat` ([CMat](#) \*const mat)
- static void `finish_matrix` ([Mat](#) \*const mat)
- static void `init_c_mat` ([CMat](#) \*const mat, const int radius)
- static void `make_circle_convolution` (const double radius, [CMat](#) \*const convolution, const int m)
- static void `make_gaussian_convolution` (const double alpha, [CMat](#) \*const convolution, const int m)
- static double `mat_elt` (const [Mat](#) \*const mat, const int r, const int c)



## 6.1219 Digikam::RegionFrameItem Class Reference

Inheritance diagram for Digikam::RegionFrameItem:



### Public Types

- enum **Flag** { **NoFlags** = 0 , **ShowResizeHandles** = 1 << 0 , **MoveByDrag** = 1 << 1 , **GeometryEditable** = ShowResizeHandles | MoveByDrag }

### Public Slots

- void [setViewportRect](#) (const QRectF &[rect](#))

### Signals

- void [geometryEdited](#) ()

### Signals inherited from [Digikam::DImgChildItem](#)

- void [geometryChanged](#) ()
- void [geometryOnImageChanged](#) ()
- void [positionChanged](#) ()
- void [positionOnImageChanged](#) ()
- void [sizeChanged](#) ()
- void [sizeOnImageChanged](#) ()

### Public Member Functions

- void [changeFlags](#) (Flags flags, bool addOrRemove)
- Flags [flags](#) () const
- QGraphicsWidget \* [hudWidget](#) () const
- [RegionFrameItem](#) (QGraphicsItem \*const parent)
- void [setFixedRatio](#) (double ratio)
- void [setFlags](#) (Flags flags)
- void [setHudWidget](#) (QGraphicsWidget \*const hudWidget)
- void [setHudWidget](#) (QWidget \*const widget, Qt::WindowFlags wFlags=Qt::WindowFlags())
- void [setHudWidgetVisible](#) (bool visible)
- void [setRectInSceneCoordinatesAdjusted](#) (const QRectF &[rect](#))

### Public Member Functions inherited from [Digikam::DImgChildItem](#)

- QRectF [boundingRect](#) () const override
- [DImgChildItem](#) (QGraphicsItem \*const parent=nullptr)
- void [moveBy](#) (qreal dx, qreal dy)
- QPoint [originalPos](#) () const
- QRect [originalRect](#) () const
- QSize [originalSize](#) () const
- [GraphicsDImgItem](#) \* [parentDImgItem](#) () const
- QRectF [rect](#) () const
- QPointF [relativePos](#) () const
- QRectF [relativeRect](#) () const
- QSizeF [relativeSize](#) () const
- void [setOriginalPos](#) (const QPointF &posInOriginal)
- void [setOriginalPos](#) (qreal x, qreal y)
- void [setOriginalRect](#) (const QRectF &[rect](#))
- void [setOriginalRect](#) (qreal x, qreal y, qreal width, qreal height)
- void [setOriginalSize](#) (const QSizeF &sizeInOriginal)
- void [setOriginalSize](#) (qreal width, qreal height)
- void [setPos](#) (const QPointF &zoomedPos)
- void [setPos](#) (qreal x, qreal y)

- void **setRect** (const QRectF &rect)
- void **setRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRectInSceneCoordinates** (const QRectF &rect)
- void **setRelativePos** (const QPointF &relativePosition)
- void **setRelativePos** (qreal x, qreal y)
- void **setRelativeRect** (const QRectF &rect)
- void **setRelativeRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRelativeSize** (const QSizeF &relativeSize)
- void **setRelativeSize** (qreal width, qreal height)
- void **setSize** (const QSizeF &zoomedSize)
- void **setSize** (qreal width, qreal height)
- QSizeF **size** () const

### Protected Member Functions

- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- void **hoverEnterEvent** (QGraphicsSceneHoverEvent \*event) override
- void **hoverLeaveEvent** (QGraphicsSceneHoverEvent \*event) override
- void **hoverMoveEvent** (QGraphicsSceneHoverEvent \*event) override
- void **mouseMoveEvent** (QGraphicsSceneMouseEvent \*) override
- void **mousePressEvent** (QGraphicsSceneMouseEvent \*) override
- void **mouseReleaseEvent** (QGraphicsSceneMouseEvent \*) override
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget=nullptr) override

### Protected Member Functions inherited from [Digikam::DImgChildItem](#)

- QVariant **itemChange** (GraphicsItemChange change, const QVariant &value) override

### Additional Inherited Members

### Protected Slots inherited from [Digikam::DImgChildItem](#)

- void **imageSizeChanged** (const QSizeF &)

## 6.1219.1 Member Function Documentation

### 6.1219.1.1 setHudWidget()

```
void Digikam::RegionFrameItem::setHudWidget (
    QGraphicsWidget *const hudWidget )
```

Sets a widget item as HUD item. A HUD item will be positioned relative to this item, and repositioned on position changes or resizing. Ownership of the item is taken, and it is made a child item of this item. You can also add QWidget directly. It will be wrapped in a proxy item.

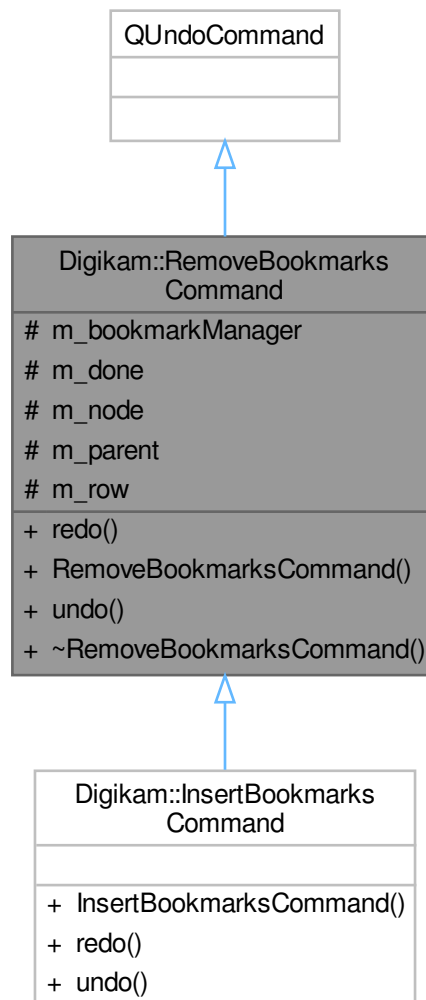
### 6.1219.1.2 setViewportRect

```
void Digikam::RegionFrameItem::setViewportRect (
    const QRectF & rect ) [slot]
```

The associated HUD item is dynamically moved to be visible. This can only be done for *one* region at a time. Set the current primary view region of the scene by this method to dynamically reposition the HUD inside this region. The rect given is in scene coordinates.

## 6.1220 Digikam::RemoveBookmarksCommand Class Reference

Inheritance diagram for Digikam::RemoveBookmarksCommand:



### Public Member Functions

- void **redo** () override
- **RemoveBookmarksCommand** ([BookmarksManager](#) \*const mgr, [BookmarkNode](#) \*const parent, int row)
- void **undo** () override

### Protected Attributes

- [BookmarksManager](#) \* **m\_bookmarkManager** = nullptr
- bool **m\_done** = false
- [BookmarkNode](#) \* **m\_node** = nullptr
- [BookmarkNode](#) \* **m\_parent** = nullptr
- int **m\_row** = 0

## 6.1221 Digikam::RemoveDoublesModifier Class Reference

Inheritance diagram for Digikam::RemoveDoublesModifier:



### Public Member Functions

- QString `parseOperation` (`ParseSettings` &settings, const QRegularExpressionMatch &match) override

## Public Member Functions inherited from Digikam::Modifier

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from Digikam::Rule

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- ParseResults **parse** (ParseSettings &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Additional Inherited Members

## Public Types inherited from Digikam::Rule

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from Digikam::Rule

- void **signalTokenTriggered** (const QString &)

## Static Public Member Functions inherited from Digikam::Rule

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from Digikam::Rule

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions inherited from Digikam::Rule

- bool **addToken** (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void **setUseTokenMenu** (bool value)

## 6.1221.1 Member Function Documentation

### 6.1221.1.1 parseOperation()

```
QString Digikam::RemoveDoublesModifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [virtual]
```

TODO: describe me

## Parameters

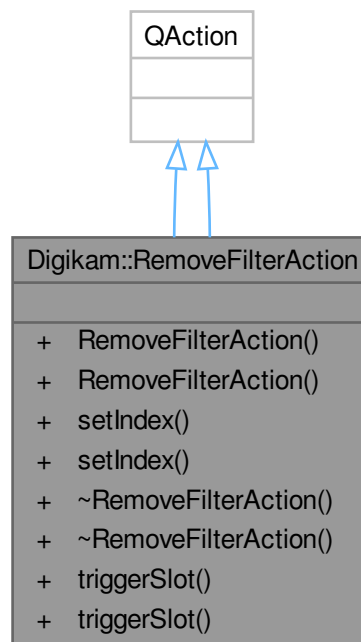
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Modifier](#).

## 6.1222 Digikam::RemoveFilterAction Class Reference

Inheritance diagram for Digikam::RemoveFilterAction:



## Public Slots

- void **triggerSlot** ()
- void **triggerSlot** ()

## Signals

- void **actionTriggered** (QModelIndex index)
- void **actionTriggered** (QModelIndex index)

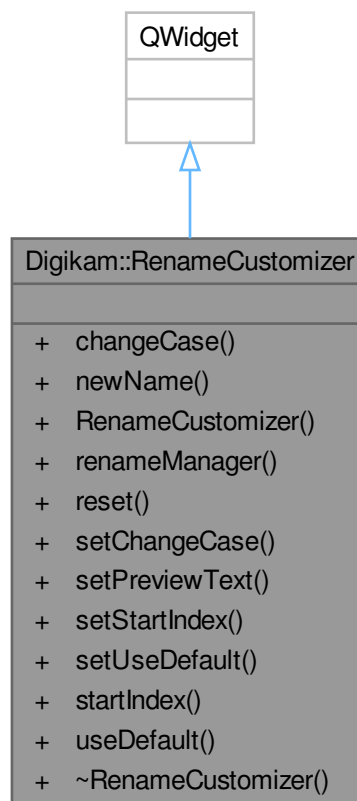


### Public Member Functions

- **RemoveFilterAction** (const QString &label, const QModelIndex &index, QObject \*const parent=nullptr)
- **RemoveFilterAction** (const QString &label, const QModelIndex &index, QObject \*const parent=nullptr)
- void **setIndex** (const QModelIndex &index)
- void **setIndex** (const QModelIndex &index)

## 6.1223 Digikam::RenameCustomizer Class Reference

Inheritance diagram for Digikam::RenameCustomizer:



### Public Types

- enum **Case** { **NONE** = 0 , **UPPER** , **LOWER** }

### Signals

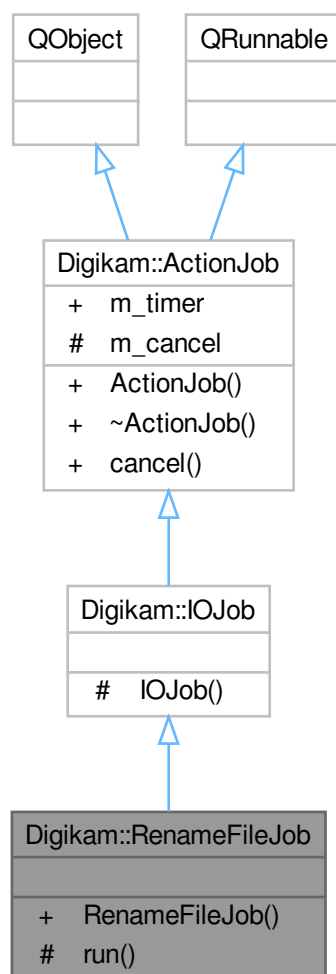
- void **signalChanged** ()

### Public Member Functions

- Case **changeCase** () const
- QString **newName** (const QString &fileName) const
- **RenameCustomizer** (QWidget \*const parent, const QString &cameraTitle)
- [AdvancedRenameManager](#) \* **renameManager** () const
- void **reset** ()
- void **setChangeCase** (Case val)
- void **setPreviewText** (const QString &txt)
- void **setStartIndex** (int startIndex)
- void **setUseDefault** (bool val)
- int **startIndex** () const
- bool **useDefault** () const

## 6.1224 Digikam::RenameFileJob Class Reference

Inheritance diagram for Digikam::RenameFileJob:



## Signals

- void **signalRenameFailed** (const QUrl &url)

## Signals inherited from [Digikam::IOJob](#)

- void **signalError** (const QString &errMsg)
- void **signalOneProcessed** (const QUrl &url)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Member Functions

- **RenameFileJob** ([IOJobData](#) \*const data)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void **run** () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

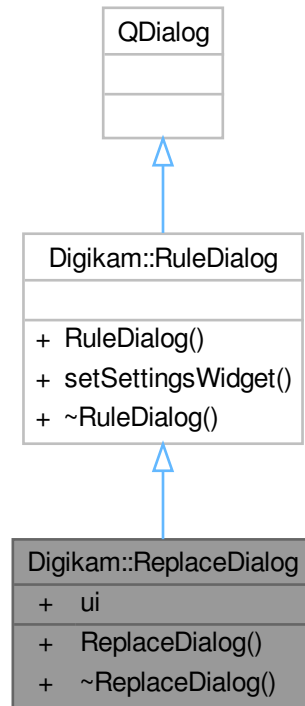
- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.1225 Digikam::ReplaceDialog Class Reference

Inheritance diagram for Digikam::ReplaceDialog:



### Public Member Functions

- **ReplaceDialog** ([Rule](#) \*const parent)

### Public Member Functions inherited from [Digikam::RuleDialog](#)

- **RuleDialog** ([Rule](#) \*const parent)
- void **setSettingsWidget** (QWidget \*const settingsWidget)

### Public Attributes

- Ui::ReplaceModifierDialogWidget \*const **ui** = nullptr

## 6.1226 Digikam::ReplaceModifier Class Reference

Inheritance diagram for Digikam::ReplaceModifier:



### Public Member Functions

- QString [parseOperation](#) ([ParseSettings](#) &settings, const QRegularExpressionMatch &match) override

## Public Member Functions inherited from [Digikam::Modifier](#)

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- [ParseResults](#) **parse** ([ParseSettings](#) &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from [Digikam::Rule](#)

- void **signalTokenTriggered** (const QString &)

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool **addToken** (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void **setUseTokenMenu** (bool value)

## 6.1226.1 Member Function Documentation

### 6.1226.1.1 `parseOperation()`

```
QString Digikam::ReplaceModifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [virtual]
```

TODO: describe me

## Parameters

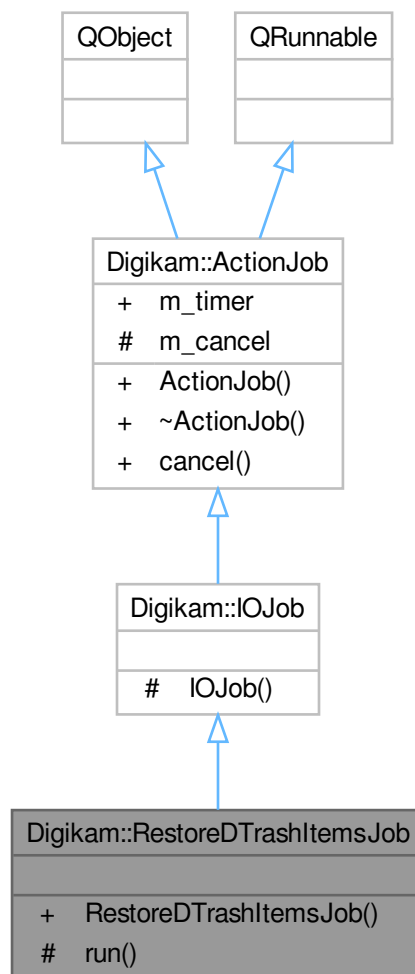
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in Option::parse()

## Returns

Implements [Digikam::Modifier](#).

## 6.1227 Digikam::RestoreDTrashItemsJob Class Reference

Inheritance diagram for Digikam::RestoreDTrashItemsJob:



### Public Member Functions

- **RestoreDTrashItemsJob** ([IOJobData](#) \*const data)

### Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

### Protected Member Functions

- void **run** () override

### Additional Inherited Members

### Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

### Signals inherited from [Digikam::IOJob](#)

- void **signalError** (const QString &errMsg)
- void **signalOneProcessed** (const QUrl &url)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

### Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false



## 6.1228 Digikam::RGBackend Class Reference

This class is a base class for Open Street Map and Geonames backends.

Inheritance diagram for Digikam::RGBackend:



### Signals

- void **signalRGReady** (const QList< [RGInfo](#) > &)  
*Emitted whenever some items are ready.*

### Public Member Functions

- virtual QString [backendName](#) ()
- virtual void [callRGBackend](#) (const QList< [RGInfo](#) > &, const QString &)=0
- virtual void **cancelRequests** ()=0
- virtual QString [getErrorMessage](#) ()
- [RGBackend](#) (QObject \*const parent)

## 6.1228.1 Constructor & Destructor Documentation

### 6.1228.1.1 RGBackend()

```
Digikam::RGBackend::RGBackend (
    QObject *const parent ) [explicit]
```

Constructor

## 6.1228.2 Member Function Documentation

### 6.1228.2.1 backendName()

```
QString Digikam::RBackend::backendName ( ) [virtual]
```

Reimplemented in [Digikam::BackendGeonamesRG](#), [Digikam::BackendGeonamesUSRG](#), and [Digikam::BackendOsmRG](#).

### 6.1228.2.2 callRBackend()

```
virtual void Digikam::RBackend::callRBackend (
    const QList< RInfo > & ,
    const QString & ) [pure virtual]
```

Implemented in [Digikam::BackendGeonamesRG](#), [Digikam::BackendGeonamesUSRG](#), and [Digikam::BackendOsmRG](#).

### 6.1228.2.3 getErrorMessage()

```
QString Digikam::RBackend::getErrorMessage ( ) [virtual]
```

Reimplemented in [Digikam::BackendGeonamesRG](#), [Digikam::BackendGeonamesUSRG](#), and [Digikam::BackendOsmRG](#).

## 6.1229 Digikam::RInfo Class Reference

This class contains data needed in reverse geocoding process.

### Public Member Functions

- [RInfo \(\)](#)=default
- [~RInfo \(\)](#)=default

### Public Attributes

- [GeoCoordinates](#) `coordinates`
- [QPersistentModelIndex](#) `id`
- [QMap< QString, QString >](#) `rgData`

## 6.1229.1 Constructor & Destructor Documentation

### 6.1229.1.1 RInfo()

```
Digikam::RInfo::RInfo ( ) [default]
```

Constructor

### 6.1229.1.2 ~RGInfo()

```
Digikam::RGInfo::~RGInfo ( ) [default]
```

Destructor

## 6.1229.2 Member Data Documentation

### 6.1229.2.1 coordinates

```
GeoCoordinates Digikam::RGInfo::coordinates
```

The coordinates of current image.

### 6.1229.2.2 id

```
QPersistentModelIndex Digikam::RGInfo::id
```

The image index.

### 6.1229.2.3 rgData

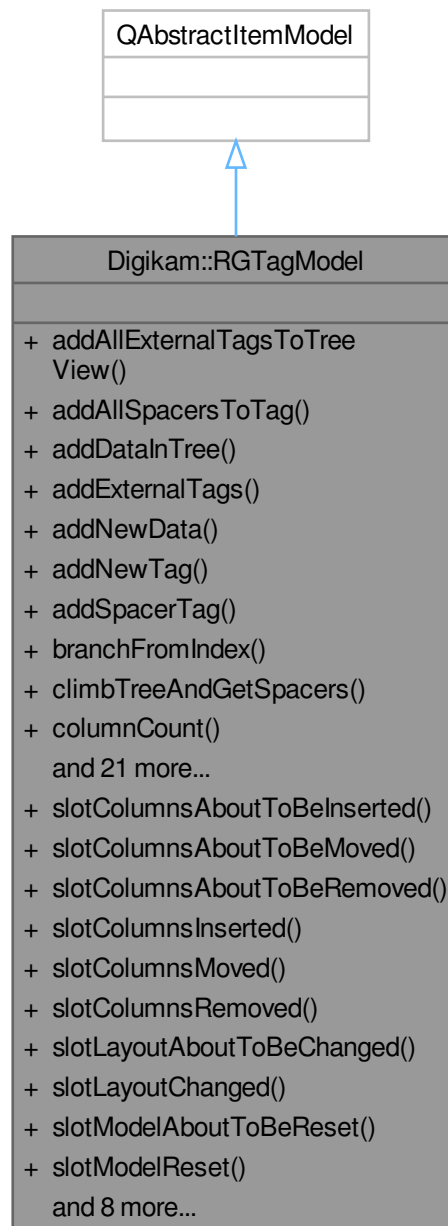
```
QMap<QString, QString> Digikam::RGInfo::rgData
```

The address elements and their names.

## 6.1230 Digikam::RGTagModel Class Reference

The model that holds data for the tag tree displayed in ReverseGeocodingWidget.

Inheritance diagram for Digikam::RGTAGModel:



### Public Slots

- void **slotColumnsAboutToBeInserted** (const QModelIndex &parent, int start, int end)
- void **slotColumnsAboutToBeMoved** (const QModelIndex &sourceParent, int sourceStart, int sourceEnd, const QModelIndex &destinationParent, int destinationColumn)
- void **slotColumnsAboutToBeRemoved** (const QModelIndex &parent, int start, int end)
- void **slotColumnsInserted** ()
- void **slotColumnsMoved** ()

- void **slotColumnsRemoved** ()
- void **slotLayoutAboutToBeChanged** ()
- void **slotLayoutChanged** ()
- void **slotModelAboutToBeReset** ()
- void **slotModelReset** ()
- void **slotRowsAboutToBeInserted** (const QModelIndex &parent, int start, int end)
- void **slotRowsAboutToBeMoved** (const QModelIndex &sourceParent, int sourceStart, int sourceEnd, const QModelIndex &destinationParent, int destinationRow)
- void **slotRowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end)
- void **slotRowsInserted** ()
- void **slotRowsMoved** ()
- void **slotRowsRemoved** ()
- void **slotSourceDataChanged** (const QModelIndex &topLeft, const QModelIndex &bottomRight)
- void **slotSourceHeaderDataChanged** (const Qt::Orientation orientation, int first, int last)

### Public Member Functions

- void **addAllExternalTagsToTreeView** ()  
*Add all external tags to the tag tree.*
- void **addAllSpacersToTag** (const QModelIndex &currentIndex, const QStringList &spacerList, int spacer↵ListIndex)  
*Adds all spacers found in spacerList to the tag tree.*
- void **addDataInTree** (TreeBranch \*currentBranch, int currentRow, const QStringList &addressElements, const QStringList &elementsData)  
*The function starts to scan the tree starting with currentBranch. When it finds a spacer containing an address element, it looks to see if the address element is found in addressElements list. If it's found, a new tag is added.*
- void **addExternalTags** (TreeBranch \*parentBranch, int currentRow)  
*Add tags from host application to the tag tree.*
- QList< QList< TagData > > **addNewData** (const QStringList &elements, const QStringList &resultedData)  
*Add new tags to tag tree. The function starts to scan the tree from root level. When it finds a spacer containing an address element, it looks to see if the address element is found in elements list. If it's found, a new tag is added.*
- QPersistentModelIndex **addNewTag** (const QModelIndex &parent, const QString &newTagName, const QString &newElement)  
*Adds a tag containing data returned from backends.*
- void **addSpacerTag** (const QModelIndex &parent, const QString &spacerName)  
*Adds a spacer tag.*
- TreeBranch \* **branchFromIndex** (const QModelIndex &index) const  
*Returns the branch found at index.*
- void **climbTreeAndGetSpacers** (const TreeBranch \*currentBranch)  
*Gets the spacers addresses below currentBranch. Address means the path from rootTag to currentBranch.*
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override  
*QAbstractItemModel.*
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- void **deleteAllSpacersOrNewTags** (const QModelIndex &currentIndex, Type whatShouldRemove)  
*Deletes all spacers or all new tags.*
- void **deleteTag** (const QModelIndex &currentIndex)  
*Deletes a tag.*
- void **findAndDeleteSpacersOrNewTags** (TreeBranch \*currentBranch, int currentRow, Type whatShould↵Remove)  
*Deletes all spacers or all new tags below current branch.*
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- QModelIndex **fromSourceIndex** (const QModelIndex &externalTagModelIndex) const

*Local functions.*

- `QList< TagData > getSpacerAddress (TreeBranch *currentBranch)`  
*Gets the address of a spacer. Address means the path from rootTag to currentBranch.*
- `QList< QList< TagData > > getSpacers ()`  
*Gets all spacers.*
- `QList< TagData > getTagAddress ()`  
*Gets the address of a tag.*
- `Type getTagType (const QModelIndex &index) const`  
*Gets the type of a tag found at index.*
- `QVariant headerData (int section, Qt::Orientation orientation, int role) const override`
- `QModelIndex index (int row, int column, const QModelIndex &parent=QModelIndex()) const override`
- `QModelIndex parent (const QModelIndex &index) const override`
- `void readNewTags (const QList< QList< TagData > > &tagAddressList)`  
*Takes each tag contained in tagAddressList and adds it to the tag tree.*
- `void readTag (TreeBranch *&currentBranch, int currentRow, const QList< TagData > &tagAddressElements, int currentAddressElementIndex)`  
*Reads new tags to tag tree.*
- `RGTagModel (QAbstractItemModel *const externalTagModel, QObject *const parent=nullptr)`  
*Constructor.*
- `int rowCount (const QModelIndex &parent=QModelIndex()) const override`
- `bool setData (const QModelIndex &index, const QVariant &value, int role) override`
- `bool setHeaderData (int section, Qt::Orientation orientation, const QVariant &value, int role) override`
- `QModelIndex toSourceIndex (const QModelIndex &tagModelIndex) const`  
*Translates the model index from this model to host's tag model.*
- `~RGTagModel () override`  
*Destructor.*

### 6.1230.1 Detailed Description

The `RGTagModel` class is a wrapper above `QAbstractItemModel`. It holds data for the tag tree displayed in `ReverseGeocodingWidget`. The model gets the data from the tag model of host application and displays it in a `QTreeView`. It stores three type of tags: old tags (the tags that belong to the host's tag model), spacer tags (tags representing address elements or custom tags) and new tags (tags containing data retrieved from backend).

### 6.1230.2 Constructor & Destructor Documentation

#### 6.1230.2.1 RGTagModel()

```

Digikam::RGTagModel::RGTagModel (
    QAbstractItemModel *const externalTagModel,
    QObject *const parent = nullptr ) [explicit]

```

#### Parameters

<code>externalTagModel</code>	The tag model found in the host application.
<code>parent</code>	The parent object

### 6.1230.3 Member Function Documentation

#### 6.1230.3.1 addDataInTree()

```
void Digikam::RGTagModel::addDataInTree (
    TreeBranch * currentBranch,
    int currentRow,
    const QStringList & addressElements,
    const QStringList & elementsData )
```

##### Parameters

<i>currentBranch</i>	The branch from where the scan starts.
<i>currentRow</i>	The row of the current branch.
<i>addressElements</i>	A list containing address elements. Example: {Country}, {City}...
<i>elementsData</i>	A list containing the name of each address element found in elements. Example: France, Paris...

#### 6.1230.3.2 addExternalTags()

```
void Digikam::RGTagModel::addExternalTags (
    TreeBranch * parentBranch,
    int currentRow )
```

##### Parameters

<i>parentBranch</i>	The branch that will be parent for the old tag.
<i>currentRow</i>	The row where this external tag will be added.

#### 6.1230.3.3 addNewData()

```
QList< QList< TagData > > Digikam::RGTagModel::addNewData (
    const QStringList & elements,
    const QStringList & resultedData )
```

##### Parameters

<i>elements</i>	A list containing address elements. Example: {Country}, {City}...
<i>resultedData</i>	A list containing the name of each address element found in elements. Example: France, Paris...

##### Returns

A list containing new tags

#### 6.1230.3.4 addNewTag()

```
QPersistentModelIndex Digikam::RGTagModel::addNewTag (
    const QModelIndex & parent,
```

```
const QString & newTagName,
const QString & newElement )
```

**Parameters**

<i>parent</i>	The index of the parent.
<i>newTagName</i>	The name of the new tag.
<i>newElement</i>	The new element of the tag.

**6.1230.3.5 addSpacerTag()**

```
void Digikam::RGTagModel::addSpacerTag (
    const QModelIndex & parent,
    const QString & spacerName )
```

**Parameters**

<i>parent</i>	The index of the parent. If parent == QModelIndex(), then the spacer is added to top-level
<i>spacerName</i>	The name of the spacer. If it's an address element, the address element name will have the form {addressElement}. For example: {Country}, {City}...

**6.1230.3.6 branchFromIndex()**

```
TreeBranch * Digikam::RGTagModel::branchFromIndex (
    const QModelIndex & index ) const
```

**Parameters**

<i>index</i>	Current model index.
--------------	----------------------

**Returns**

The branch for the current index.

**6.1230.3.7 climbTreeAndGetSpacers()**

```
void Digikam::RGTagModel::climbTreeAndGetSpacers (
    const TreeBranch * currentBranch )
```

**Parameters**

<i>currentBranch</i>	The branch from where the search starts.
----------------------	--



**6.1230.3.8 deleteAllSpacersOrNewTags()**

```
void Digikam::RGTagModel::deleteAllSpacersOrNewTags (
    const QModelIndex & currentIndex,
    Type whatShouldRemove )
```

**Parameters**

<i>currentIndex</i>	If <i>whatShouldRemove</i> represents a spacer, the function will remove all spacers below <i>currentIndex</i> . If <i>whatShouldRemove</i> represents a new tag, the function will delete all new tags.
<i>whatShouldRemove</i>	The tag type that should be removed. The options are: spacers or new tags.

**6.1230.3.9 deleteTag()**

```
void Digikam::RGTagModel::deleteTag (
    const QModelIndex & currentIndex )
```

**Parameters**

<i>currentIndex</i>	The tag found at this index will be deleted.
---------------------	--

**6.1230.3.10 findAndDeleteSpacersOrNewTags()**

```
void Digikam::RGTagModel::findAndDeleteSpacersOrNewTags (
    TreeBranch * currentBranch,
    int currentRow,
    Type whatShouldRemove )
```

**Parameters**

<i>currentBranch</i>	The tree branch from where the scan starts.
<i>currentRow</i>	The row of current branch.
<i>whatShouldRemove</i>	The tag type that should to be removed. The options are: spacers or new tags.

**6.1230.3.11 fromSourceIndex()**

```
QModelIndex Digikam::RGTagModel::fromSourceIndex (
    const QModelIndex & externalTagModelIndex ) const
```

Translates the model index from host's tag model to this model.

**Returns**

The index of current old tag.

**6.1230.3.12 getSpacerAddress()**

```
QList< TagData > Digikam::RGTAGModel::getSpacerAddress (
    TreeBranch * currentBranch )
```

**Parameters**

<i>currentBranch</i>	The branch where the scan stops.
----------------------	----------------------------------

**Returns**

The tag address of *currentBranch*

**6.1230.3.13 getSpacers()**

```
QList< QList< TagData > > Digikam::RGTagModel::getSpacers ( )
```

**Returns**

The spacer list.

**6.1230.3.14 getTagType()**

```
Type Digikam::RGTagModel::getTagType (
    const QModelIndex & index ) const
```

**Parameters**

<i>index</i>	The index of the tag.
--------------	-----------------------

**Returns**

The type of the tag found at *index*.

**6.1230.3.15 readdNewTags()**

```
void Digikam::RGTagModel::readdNewTags (
    const QList< QList< TagData > > & tagAddressList )
```

**Parameters**

<i>tagAddressList</i>	A list containing new tags.
-----------------------	-----------------------------

**6.1230.3.16 readdTag()**

```
void Digikam::RGTagModel::readdTag (
    TreeBranch *& currentBranch,
    int currentRow,
    const QList< TagData > & tagAddressElements,
    int currentAddressElementIndex )
```

## Parameters

<i>currentBranch</i>	The branch from where the scan starts.
<i>currentRow</i>	The row of the currentBranch.
<i>tagAddressElements</i>	A list containing address elements. Example: {Country}, {City}...
<i>currentAddressElementIndex</i>	The current element in the tag address list.

## Note

tagAddressElements contains address tag: Places,Spain,Barcelona readTag climbs the tree and checks on each level if tagAddressElements[level] is found. if the tag is found, it climbs up the next level else, it recreates the new tag and climbs up that tree.

**6.1230.3.17 toSourceIndex()**

```
QModelIndex Digikam::RGTagModel::toSourceIndex (
    const QModelIndex & tagModelIndex ) const
```

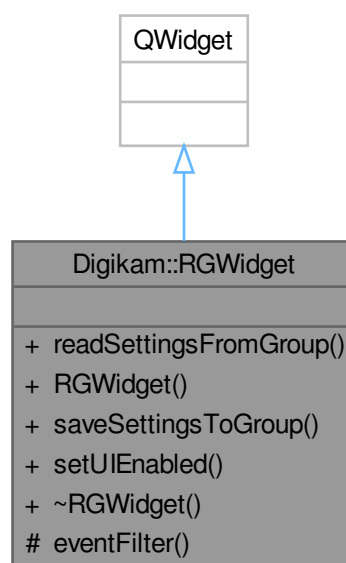
## Returns

The index of a tag in host's tag model.

**6.1231 Digikam::RGWidget Class Reference**

The [RGWidget](#) class represents the main widget for reverse geocoding.

Inheritance diagram for Digikam::RGWidget:



## Signals

- void [signalProgressChanged](#) (const int currentProgress)
- void [signalProgressSetup](#) (const int maxProgress, const QString &progressText)
- void [signalSetUIEnabled](#) (const bool enabledState)
- void **signalSetUIEnabled** (const bool enabledState, QObject \*const cancelObject, const QString &cancelSlot)
- void [signalUndoCommand](#) (GPSUndoCommand \*undoCommand)

## Public Member Functions

- void [readSettingsFromGroup](#) (const KConfigGroup \*const group)
- [RGWidget](#) (GPSItemModel \*const imageModel, QItemSelectionModel \*const selectionModel, QAbstractItemModel \*externTagModel, QWidget \*const parent=nullptr)  
*Constructor.*
- void [saveSettingsToGroup](#) (KConfigGroup \*const group)
- void [setUIEnabled](#) (const bool state)
- [~RGWidget](#) () override

## Protected Member Functions

- bool [eventFilter](#) (QObject \*watched, QEvent \*event) override

## 6.1231.1 Constructor & Destructor Documentation

### 6.1231.1.1 RGWidget()

```
Digikam::RGWidget::RGWidget (
    GPSItemModel *const imageModel,
    QItemSelectionModel *const selectionModel,
    QAbstractItemModel * externTagModel,
    QWidget *const parent = nullptr ) [explicit]
```

#### Parameters

<i>imageModel</i>	The image model
<i>selectionModel</i>	The image selection model
<i>externTagModel</i>	The tag model
<i>parent</i>	The parent object

### 6.1231.1.2 ~RGWidget()

```
Digikam::RGWidget::~~RGWidget ( ) [override]
```

#### Destructor

## 6.1231.2 Member Function Documentation

### 6.1231.2.1 eventFilter()

```
bool Digikam::RGWidget::eventFilter (
    QObject * watched,
    QEvent * event ) [override], [protected]
```

Here are filtered the events.

### 6.1231.2.2 readSettingsFromGroup()

```
void Digikam::RGWidget::readSettingsFromGroup (
    const KConfigGroup *const group )
```

Restores the settings of widgets contained in reverse geocoding widget.

#### Parameters

<i>group</i>	Here are stored the settings.
--------------	-------------------------------

### 6.1231.2.3 saveSettingsToGroup()

```
void Digikam::RGWidget::saveSettingsToGroup (
    KConfigGroup *const group )
```

Saves the settings of widgets contained in reverse geocoding widget.

#### Parameters

<i>group</i>	Here are stored the settings.
--------------	-------------------------------

### 6.1231.2.4 setUIEnabled()

```
void Digikam::RGWidget::setUIEnabled (
    const bool state )
```

Sets whether the containing widgets are enabled or disabled.

#### Parameters

<i>state</i>	If true, the controls are enabled.
--------------	------------------------------------

### 6.1231.2.5 signalProgressChanged

```
void Digikam::RGWidget::signalProgressChanged (
```

```
const int currentProgress ) [signal]
```

Counts how many images were processed.

#### Parameters

<i>currentProgress</i>	The number of processed images.
------------------------	---------------------------------

### 6.1231.2.6 signalProgressSetup

```
void Digikam::RGWidget::signalProgressSetup (
    const int maxProgress,
    const QString & progressText ) [signal]
```

Update the progress bar.

### 6.1231.2.7 signalSetUIEnabled

```
void Digikam::RGWidget::signalSetUIEnabled (
    const bool enabledState ) [signal]
```

This signal emits when containing widgets need to be enabled or disabled.

#### Parameters

<i>enabledState</i>	If true, the containing widgets will be enabled. Else, they will be disabled.
---------------------	---

### 6.1231.2.8 signalUndoCommand

```
void Digikam::RGWidget::signalUndoCommand (
    GPSUndoCommand * undoCommand ) [signal]
```

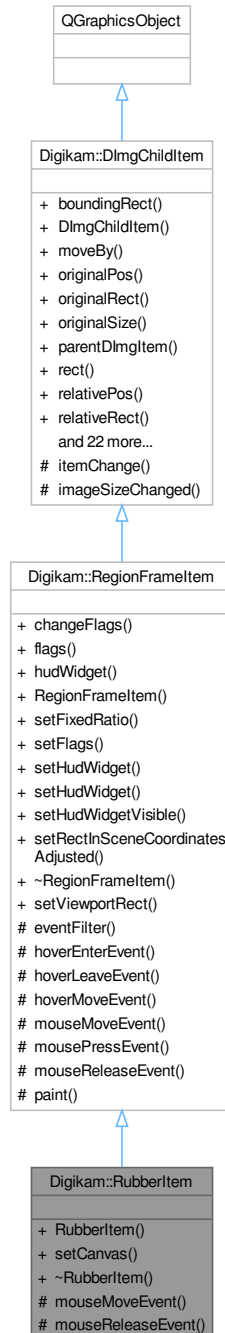
Sends the needed data to Undo/Redo Widget.

#### Parameters

<i>undoCommand</i>	Holds the data that will be used for undo or redo actions
--------------------	---

## 6.1232 Digikam::RubberItem Class Reference

Inheritance diagram for Digikam::RubberItem:



### Public Member Functions

- **RubberItem** ([ImagePreviewItem](#) \*const item)
- void **setCanvas** ([Canvas](#) \*const canvas)



## Public Member Functions inherited from [Digikam::RegionFrameItem](#)

- void **changeFlags** (Flags flags, bool addOrRemove)
- Flags **flags** () const
- QGraphicsWidget \* **hudWidget** () const
- **RegionFrameItem** (QGraphicsItem \*const parent)
- void **setFixedRatio** (double ratio)
- void **setFlags** (Flags flags)
- void **setHudWidget** (QGraphicsWidget \*const hudWidget)
- void **setHudWidget** (QWidget \*const widget, Qt::WindowFlags wFlags=Qt::WindowFlags())
- void **setHudWidgetVisible** (bool visible)
- void **setRectInSceneCoordinatesAdjusted** (const QRectF &rect)

## Public Member Functions inherited from [Digikam::DImgChildItem](#)

- QRectF **boundingRect** () const override
- **DImgChildItem** (QGraphicsItem \*const parent=nullptr)
- void **moveBy** (qreal dx, qreal dy)
- QPoint **originalPos** () const
- QRect **originalRect** () const
- QSize **originalSize** () const
- **GraphicsDImgItem** \* **parentDImgItem** () const
- QRectF **rect** () const
- QPointF **relativePos** () const
- QRectF **relativeRect** () const
- QSizeF **relativeSize** () const
- void **setOriginalPos** (const QPointF &posInOriginal)
- void **setOriginalPos** (qreal x, qreal y)
- void **setOriginalRect** (const QRectF &rect)
- void **setOriginalRect** (qreal x, qreal y, qreal width, qreal height)
- void **setOriginalSize** (const QSizeF &sizeInOriginal)
- void **setOriginalSize** (qreal width, qreal height)
- void **setPos** (const QPointF &zoomedPos)
- void **setPos** (qreal x, qreal y)
- void **setRect** (const QRectF &rect)
- void **setRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRectInSceneCoordinates** (const QRectF &rect)
- void **setRelativePos** (const QPointF &relativePosition)
- void **setRelativePos** (qreal x, qreal y)
- void **setRelativeRect** (const QRectF &rect)
- void **setRelativeRect** (qreal x, qreal y, qreal width, qreal height)
- void **setRelativeSize** (const QSizeF &relativeSize)
- void **setRelativeSize** (qreal width, qreal height)
- void **setSize** (const QSizeF &zoomedSize)
- void **setSize** (qreal width, qreal height)
- QSizeF **size** () const

## Protected Member Functions

- void **mouseMoveEvent** (QGraphicsSceneMouseEvent \*event) override
- void **mouseReleaseEvent** (QGraphicsSceneMouseEvent \*event) override

### Protected Member Functions inherited from [Digikam::RegionFrameItem](#)

- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- void **hoverEnterEvent** (QGraphicsSceneHoverEvent \*event) override
- void **hoverLeaveEvent** (QGraphicsSceneHoverEvent \*event) override
- void **hoverMoveEvent** (QGraphicsSceneHoverEvent \*event) override
- void **mouseMoveEvent** (QGraphicsSceneMouseEvent \*) override
- void **mousePressEvent** (QGraphicsSceneMouseEvent \*) override
- void **mouseReleaseEvent** (QGraphicsSceneMouseEvent \*) override
- void **paint** (QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget=nullptr) override

### Protected Member Functions inherited from [Digikam::DImgChildItem](#)

- QVariant **itemChange** (GraphicsItemChange change, const QVariant &value) override

### Additional Inherited Members

### Public Types inherited from [Digikam::RegionFrameItem](#)

- enum **Flag** { **NoFlags** = 0 , **ShowResizeHandles** = 1 << 0 , **MoveByDrag** = 1 << 1 , **GeometryEditable** = ShowResizeHandles | MoveByDrag }

### Public Slots inherited from [Digikam::RegionFrameItem](#)

- void **setViewportRect** (const QRectF &rect)

### Signals inherited from [Digikam::RegionFrameItem](#)

- void **geometryEdited** ()

### Signals inherited from [Digikam::DImgChildItem](#)

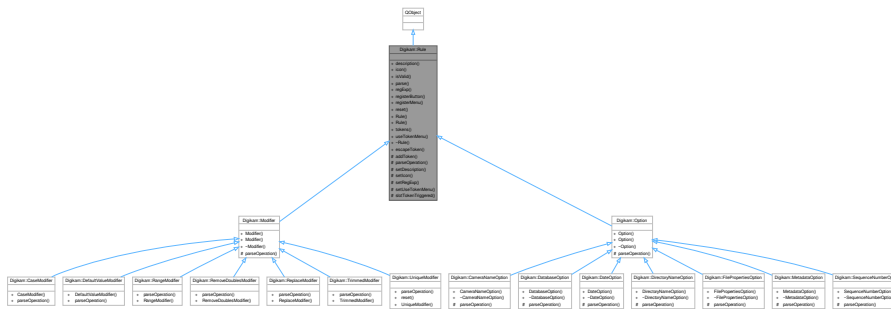
- void **geometryChanged** ()
- void **geometryOnImageChanged** ()
- void **positionChanged** ()
- void **positionOnImageChanged** ()
- void **sizeChanged** ()
- void **sizeOnImageChanged** ()

### Protected Slots inherited from [Digikam::DImgChildItem](#)

- void **imageSizeChanged** (const QSizeF &)

## 6.1233 Digikam::Rule Class Reference

Inheritance diagram for Digikam::Rule:



### Public Types

- enum **IconType** { **Action** = 0 , **Dialog** }

### Signals

- void **signalTokenTriggered** (const QString &)

### Public Member Functions

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- ParseResults **parse** (ParseSettings &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

### Static Public Member Functions

- static QString **escapeToken** (const QString &token)

### Protected Slots

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- virtual QString [parseOperation](#) ([ParseSettings](#) &settings, const QRegularExpressionMatch &match)=0
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## 6.1233.1 Member Function Documentation

### 6.1233.1.1 addToken()

```
bool Digikam::Rule::addToken (
    const QString & id,
    const QString & description,
    const QString & actionName = QString() ) [protected]
```

add a token to the parser, every parser should at least assign one token object

#### Parameters

<i>id</i>	the token id string (used for parsing)
<i>description</i>	the description of the token (used for example in the tooltip)
<i>actionName</i>	[optional] the name of the token action (only used when the token menu is displayed)

#### Returns

### 6.1233.1.2 escapeToken()

```
QString Digikam::Rule::escapeToken (
    const QString & token ) [static]
```

Escape the token characters to make them work in regular expressions

#### Parameters

<i>token</i>	the token to be escaped
--------------	-------------------------

#### Returns

A token with escaped characters. This token can then be used in a regular expression

### 6.1233.1.3 isValid()

```
bool Digikam::Rule::isValid ( ) const
```

Checks the validity of the parse object

**Returns**

true if valid

**6.1233.1.4 parseOperation()**

```
virtual QString Digikam::Rule::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [protected], [pure virtual]
```

TODO: describe me

**Parameters**

<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

**Returns**

Implemented in [Digikam::CaseModifier](#), [Digikam::DefaultValueModifier](#), [Digikam::RangeModifier](#), [Digikam::RemoveDoublesModifier](#), [Digikam::ReplaceModifier](#), [Digikam::TrimmedModifier](#), [Digikam::UniqueModifier](#), [Digikam::CameraNameOption](#), [Digikam::DatabaseOption](#), [Digikam::DateOption](#), [Digikam::DirectoryNameOption](#), [Digikam::FilePropertiesOption](#), [Digikam::MetadataOption](#), [Digikam::SequenceNumberOption](#), [Digikam::Modifier](#), and [Digikam::Option](#).

**6.1233.1.5 regExp()**

```
QRegularExpression & Digikam::Rule::regExp ( ) const
```

TODO: This is probably not needed anymore. Find out. returns the currently assigned regExp object. Note that it is returned as a const ref, meaning that if you use it in your custom parse operation, the main parse method has already searched for the pattern and filled in the results of this search, so that you can use `QRegularExpressionMatch::captured()` immediately, you don't have to search on your own.

For example when implementing the [Option::parseOperation\(\)](#) method, get the regExp object with

```
const QRegularExpression& reg = regExp();
```

and immediately fetch possible matches with

```
const QString& param1 = reg.captured(1);
```

**See also**

[Option](#)  
[Modifier](#)

**Returns**

a const ref to the assigned regexp object

### 6.1233.1.6 registerButton()

```
QPushButton * Digikam::Rule::registerButton (  
    QWidget * parent )
```

Register a button in the parent object. By calling this method, a new button for the parser object will be created and all necessary connections will be setup.

**Parameters**

<i>parent</i>	the parent object the button will be registered for
---------------	---

**Returns**

a pointer to the newly created button

**6.1233.1.7 registerMenu()**

```
QAction * Digikam::Rule::registerMenu (
    QMenu * parent )
```

Register a menu action in the parent object. By calling this method, a new action for the parser object will be created and all necessary connections will be setup.

**Parameters**

<i>parent</i>	the parent object the action will be registered for
---------------	---

**Returns**

a pointer to the newly created action

**6.1233.1.8 reset()**

```
void Digikam::Rule::reset ( ) [virtual]
```

Resets the parser to its initial state

Reimplemented in [Digikam::UniqueModifier](#).

**6.1233.1.9 setUseTokenMenu()**

```
void Digikam::Rule::setUseTokenMenu (
    bool value ) [protected]
```

If multiple tokens have been assigned to a rule, a menu will be created. If you want to display a menu for every defined token, set this method to 'true' and re-implement the

**See also**

slotTokenTriggered method.

**Parameters**

<i>value</i>	boolean parameter to set token menu usage
--------------	---

### 6.1233.1.10 tokens()

```
TokenList & Digikam::Rule::tokens ( ) const
```

#### Returns

a list of all registered tokens

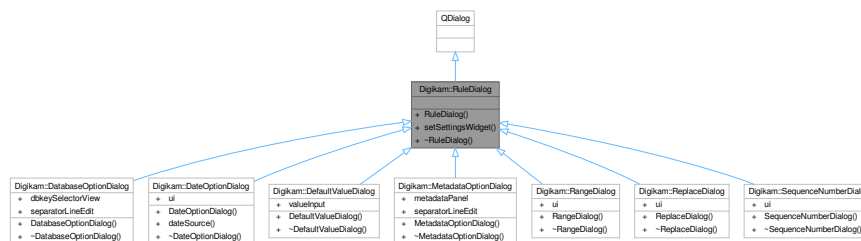
### 6.1233.1.11 useTokenMenu()

```
bool Digikam::Rule::useTokenMenu ( ) const
```

Returns true if a token menu is used.

## 6.1234 Digikam::RuleDialog Class Reference

Inheritance diagram for Digikam::RuleDialog:



### Public Member Functions

- **RuleDialog** (**Rule** \*const parent)
- void **setSettingsWidget** (QWidget \*const settingsWidget)

## 6.1235 Digikam::RuleType Class Reference

### Public Attributes

- SKey **key**
- SOperator **op** = EQ
- QString **val**

## 6.1236 Digikam::RuleTypeForConversion Class Reference

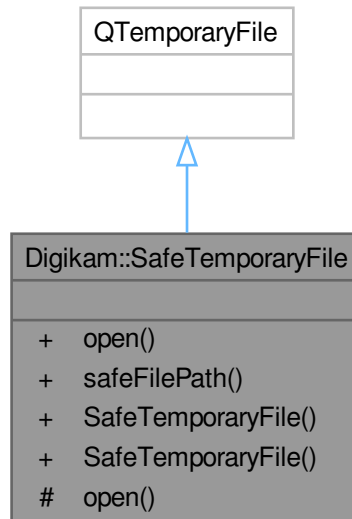
### Public Attributes

- QString **key**
- SearchXml::Relation **op** = SearchXml::Equal
- QString **val**



## 6.1237 Digikam::SafeTemporaryFile Class Reference

Inheritance diagram for Digikam::SafeTemporaryFile:



### Public Member Functions

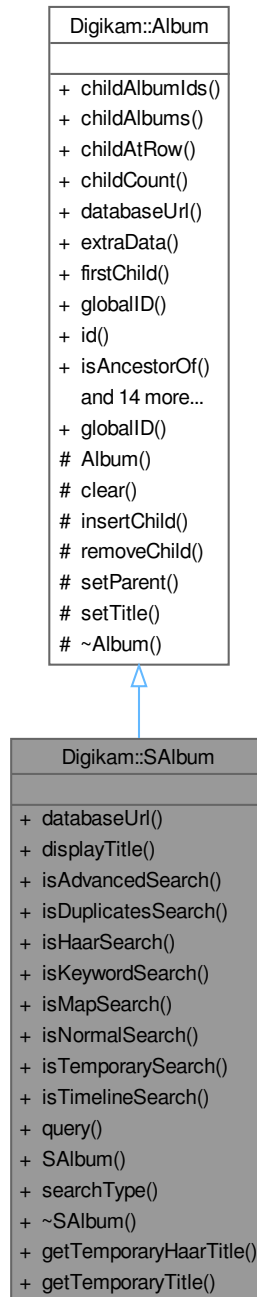
- `bool open ()`
- `QString safeFilePath () const`
- `SafeTemporaryFile (const QString &templ)`

### Protected Member Functions

- `bool open (QIODevice::OpenMode) override`

## 6.1238 Digikam::SAlbum Class Reference

Inheritance diagram for Digikam::SAlbum:



### Public Member Functions

- [CoreDbUrl databaseUrl](#) () const override
- [QString displayTitle](#) () const

- bool **isAdvancedSearch** () const
- bool **isDuplicatesSearch** () const
- bool **isHaarSearch** () const
- bool **isKeywordSearch** () const
- bool **isMapSearch** () const
- bool **isNormalSearch** () const
- bool **isTemporarySearch** () const
- bool **isTimelineSearch** () const
- QString **query** () const
- **SAlbum** (const QString &title, int id, bool root=false)
- DatabaseSearch::Type **searchType** () const

## Public Member Functions inherited from Digikam::Album

- QList< int > **childAlbumIds** (bool recursive=false)
- AlbumList **childAlbums** (bool recursive=false)
- Album \* **childAtRow** (int row) const
- int **childCount** () const
- void \* **extraData** (const void \*const key) const
- Album \* **firstChild** () const
- int **globalID** () const
- int **id** () const
- bool **isAncestorOf** (Album \*const album) const
- bool **isRoot** () const
- bool **isTrashAlbum** () const
- bool **isUsedByLabelsTree** () const
- Album \* **lastChild** () const
- Album \* **next** () const
- Album \* **parent** () const
- void **prepareForDeletion** ()
- Album \* **prev** () const
- void **removeExtraData** (const void \*const key)
- int **rowFromAlbum** () const
- void **setExtraData** (const void \*const key, void \*const value)
- void **setUsedByLabelsTree** (bool isUsed)
- QString **title** () const
- Type **type** () const

## Static Public Member Functions

- static QString **getTemporaryHaarTitle** (DatabaseSearch::HaarSearchType haarType)
- static QString **getTemporaryTitle** (DatabaseSearch::Type type, DatabaseSearch::HaarSearchType haarType=DatabaseSearch::HaarImageSearch)

## Static Public Member Functions inherited from Digikam::Album

- static int **globalID** (Type type, int id)  
*Produces the global id.*

## Friends

- class **AlbumManager**

## Additional Inherited Members

## Public Types inherited from [Digikam::Album](#)

- enum [Type](#) {  
    [PHYSICAL](#) = 0 , [TAG](#) , [DATE](#) , [SEARCH](#) ,  
    [FACE](#) }

## Protected Member Functions inherited from [Digikam::Album](#)

- [Album](#) ([Album::Type](#) type, int id, bool root)
- void [clear](#) ()
- void [insertChild](#) ([Album](#) \*const child)
- void [removeChild](#) ([Album](#) \*const child)
- void [setParent](#) ([Album](#) \*const parent)
- void [setTitle](#) (const QString &title)
- virtual [~Album](#) ()

## 6.1238.1 Detailed Description

A Search [Album](#) representation

## 6.1238.2 Member Function Documentation

### 6.1238.2.1 [databaseUrl\(\)](#)

```
CoreDbUrl Digikam::SAlbum::databaseUrl ( ) const [override], [virtual]
```

#### Returns

the kde url of the album

Implements [Digikam::Album](#).

### 6.1238.2.2 [getTemporaryHaarTitle\(\)](#)

```
QString Digikam::SAlbum::getTemporaryHaarTitle (
    DatabaseSearch::HaarSearchType haarType ) [static]
```

Returns the title for a temporary haar search depending on the sub-type used for this search

#### Parameters

<i>haarType</i>	type of the haar search to get the name for
-----------------	---

**Returns**

string that identifies this album uniquely as an unsaved search

**6.1238.2.3 getTemporaryTitle()**

```
QString Digikam::SAlbum::getTemporaryTitle (
    DatabaseSearch::Type type,
    DatabaseSearch::HaarSearchType haarType = DatabaseSearch::HaarImageSearch ) [static]
```

Returns the title of search albums that is used to mark them as a temporary search that isn't saved officially yet and is only used for viewing purposes.

**Parameters**

<i>type</i>	the type of the search to get the temporary title for
<i>haarType</i>	there are several haar searches, so that this search type needs a special handling

**Returns**

string that identifies this album uniquely as an unsaved search

**6.1238.2.4 isTemporarySearch()**

```
bool Digikam::SAlbum::isTemporarySearch ( ) const
```

Indicates whether this album is a temporary search or not.

**Returns**

true if this is a temporary search album, else false

**6.1239 Digikam::SaveProperties Class Reference****Public Attributes**

- qreal **altitude** = 0.0
- qreal **latitude** = 0.0
- qreal **longitude** = 0.0
- bool **shouldRemoveAltitude** = false
- bool **shouldRemoveCoordinates** = false
- bool **shouldWriteAltitude** = false
- bool **shouldWriteCoordinates** = false

**6.1240 Digikam::SavingContext Class Reference****Public Types**

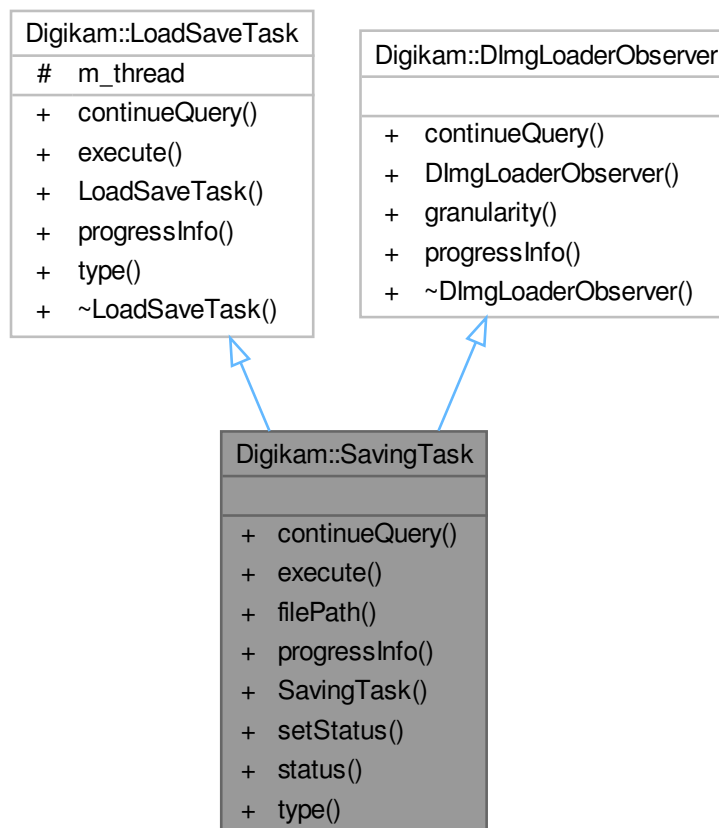
- enum **SavingState** { **SavingStateNone** , **SavingStateSave** , **SavingStateSaveAs** , **SavingStateVersion** }
- enum **SynchronizingState** { **NormalSaving** , **SynchronousSaving** }

## Public Attributes

- bool **abortingSaving** = false
- bool **destinationExisted** = false
- QUrl **destinationURL**
- SavingState **executedOperation** = SavingStateNone
- QString **format**
- QUrl **moveSrcURL**
- QString **originalFormat**
- [SafeTemporaryFile](#) \* **saveTempFile** = nullptr
- QString **saveTempFileName**
- SavingState **savingState** = SavingStateNone
- QUrl **srcURL**
- SynchronizingState **synchronizingState** = NormalSaving
- bool **synchronousSavingResult** = false
- [VersionFileOperation](#) **versionFileOperation**

## 6.1241 Digikam::SavingTask Class Reference

Inheritance diagram for Digikam::SavingTask:



## Public Types

- enum **SavingTaskStatus** { **SavingTaskStatusSaving** , **SavingTaskStatusStopping** }

## Public Types inherited from [Digikam::LoadSaveTask](#)

- enum **TaskType** { **TaskTypeLoading** , **TaskTypeSaving** }

## Public Member Functions

- bool **continueQuery** () override
- void **execute** () override
- QString **filePath** () const
- void **progressInfo** (float progress) override
- **SavingTask** (**LoadSaveThread** \*const thread, const **DImg** &img, const QString &filePath, const QString &format)
- void **setStatus** (SavingTaskStatus status)
- SavingTaskStatus **status** () const
- TaskType **type** () override

## Public Member Functions inherited from [Digikam::LoadSaveTask](#)

- **LoadSaveTask** (**LoadSaveThread** \*const thread)

## Public Member Functions inherited from [Digikam::DImgLoaderObserver](#)

- virtual float **granularity** ()

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::LoadSaveTask](#)

- **LoadSaveThread** \* **m\_thread** = nullptr

## 6.1241.1 Member Function Documentation

### 6.1241.1.1 **continueQuery()**

```
bool Digikam::SavingTask::continueQuery ( ) [override], [virtual]
```

Implements [Digikam::LoadSaveTask](#).

### 6.1241.1.2 **execute()**

```
void Digikam::SavingTask::execute ( ) [override], [virtual]
```

Implements [Digikam::LoadSaveTask](#).

**6.1241.1.3 progressInfo()**

```
void Digikam::SavingTask::progressInfo (
    float progress ) [override], [virtual]
```

Implements [Digikam::LoadSaveTask](#).

**6.1241.1.4 type()**

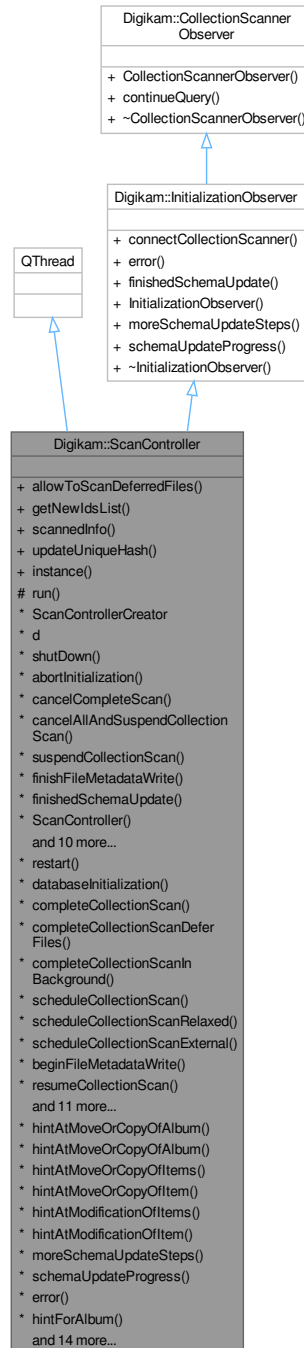
```
LoadingTask::TaskType Digikam::SavingTask::type ( ) [override], [virtual]
```

Implements [Digikam::LoadSaveTask](#).



## 6.1242 Digikam::ScanController Class Reference

Inheritance diagram for Digikam::ScanController:



### Classes

- class [FileMetadataWrite](#)
- class [Private](#)

## Public Types

- enum **Advice** { **Success** , **ContinueWithoutDatabase** , **AbortImmediately** }

## Public Types inherited from [Digikam::InitializationObserver](#)

- enum **UpdateResult** { **UpdateSuccess** , **UpdateError** , **UpdateErrorMustAbort** }

## Public Member Functions

- void **allowToScanDeferredFiles** ()
- QList< qlonglong > **getNewIdsList** () const
- **ItemInfo scannedInfo** (const QString &filePath, [CollectionScanner::FileScanMode](#) mode=[CollectionScanner::NormalScan](#))
- void **updateUniqueHash** ()

## Static Public Member Functions

- static [ScanController](#) \* **instance** ()

## Protected Member Functions

- void **run** () override

## Stop Operations

- class **ScanControllerCreator**
- void **shutDown** ()
- void **abortInitialization** ()
- void **cancelCompleteScan** ()
- void **cancelAllAndSuspendCollectionScan** ()
- void **suspendCollectionScan** ()
- void **finishFileMetadataWrite** (const [ItemInfo](#) &info, bool changed)
- void **collectionScanFinished** ()
- void **newImages** (const [ItemInfoList](#) &)
- void **partialScanDone** (const QString &path)
- void **completeScanDone** ()
- void **completeScanCanceled** ()
- void **errorFromInitialization** (const QString &)

## Start Operations

- void **restart** ()
- Advice **databaseInitialization** ()
- void **completeCollectionScan** (bool defer=false)
- void **completeCollectionScanDeferFiles** ()
- void **completeCollectionScanInBackground** (bool defer, bool fastScan=true)
- void **scheduleCollectionScan** (const QString &path)
- void **scheduleCollectionScanRelaxed** (const QString &path)
- void **scheduleCollectionScanExternal** (const QString &path)
- void **beginFileMetadataWrite** (const [ItemInfo](#) &info)
- void **resumeCollectionScan** ()
- void **restartCollectionScan** ()
- void **databaseInitialized** (bool success)
- void **collectionScanStarted** (const QString &message)

## Progress Operations

- void [hintAtMoveOrCopyOfAlbum](#) (const [PAlbum](#) \*const album, const [PAlbum](#) \*const dstAlbum, const QString &newAlbumName=QString())
- void **hintAtMoveOrCopyOfAlbum** (const [PAlbum](#) \*const album, const QString &dstPath, const QString &newAlbumName=QString())
- void [hintAtMoveOrCopyOfItems](#) (const QList< qulonglong > &ids, const [PAlbum](#) \*const dstAlbum, const QStringList &itemNames)
- void **hintAtMoveOrCopyOfItem** (qulonglong id, const [PAlbum](#) \*const dstAlbum, const QString &itemName)
- void [hintAtModificationOfItems](#) (const QList< qulonglong > &ids)
- void **hintAtModificationOfItem** (qulonglong id)
- void **totalFilesToScan** (int)
- void **startScanningAlbum** (const QString &albumRoot, const QString &album)
- void **filesScanned** (int)
- void **scanningProgress** (float progress)
- void **triggerShowProgressDialog** ()
- void **incrementProgressDialog** (int)
- void **progressFromInitialization** (const QString &, int)

## 6.1242.1 Member Function Documentation

### 6.1242.1.1 abortInitialization()

```
void Digikam::ScanController::abortInitialization ( )
```

If the controller is currently processing a database update (typically after first run), cancel this hard and as soon as possible. Any progress may be lost.

### 6.1242.1.2 beginFileMetadataWrite()

```
void Digikam::ScanController::beginFileMetadataWrite (
    const ItemInfo & info )
```

Implementation of [FileMetadataWrite](#), see there. Calling these methods is equivalent.

### 6.1242.1.3 cancelAllAndSuspendCollectionScan()

```
void Digikam::ScanController::cancelAllAndSuspendCollectionScan ( )
```

Cancels all running or scheduled operations and suspends scanning. This method returns when all scanning has stopped. This includes a call to [suspendCollectionScan\(\)](#). Restart with [resumeCollectionScan](#).

### 6.1242.1.4 cancelCompleteScan()

```
void Digikam::ScanController::cancelCompleteScan ( )
```

If the controller is currently doing a complete scan (typically at startup), stop this operation. It can be resumed later.

### 6.1242.1.5 completeCollectionScan()

```
void Digikam::ScanController::completeCollectionScan (
    bool defer = false )
```

Carries out a complete collection scan, providing progress feedback. Synchronous, returns when ready. The database will be locked while the scan is running. With the DeferFiles variant, deep files scanning (new files), the part which can take long, will be done during the time after the method returns, shortening the synchronous wait. After `completeCollectionScanDeferFiles`, you need to call `allowToScanDeferredFiles()` once to enable scanning the deferred files.

### 6.1242.1.6 completeCollectionScanInBackground()

```
void Digikam::ScanController::completeCollectionScanInBackground (
    bool defer,
    bool fastScan = true )
```

Scan Whole collection without to display a progress dialog or to manage splashscreen, as for [NewItemsFinder](#) tool.

### 6.1242.1.7 databaseInitialization()

```
ScanController::Advice Digikam::ScanController::databaseInitialization ( )
```

Calls [CoreDbAccess::checkReadyForUse\(\)](#), providing progress feedback if schema updating occurs. Synchronous, returns when ready.

### 6.1242.1.8 finishFileMetadataWrite()

```
void Digikam::ScanController::finishFileMetadataWrite (
    const ItemInfo & info,
    bool changed )
```

Implementation of [FileMetadataWrite](#), see there. Calling these methods is equivalent.

### 6.1242.1.9 getNewIdsList()

```
QList< qlonglong > Digikam::ScanController::getNewIdsList ( ) const
```

Returns item ids from new detected items

### 6.1242.1.10 hintAtModificationOfItems()

```
void Digikam::ScanController::hintAtModificationOfItems (
    const QList< qlonglong > & ids )
```

Hint at the fact that an item may have changed, although its modification date may not have changed. Note that a scan of the containing directory will need to be triggered nonetheless for the hints to take effect.

### 6.1242.1.11 hintAtMoveOrCopyOfAlbum()

```
void Digikam::ScanController::hintAtMoveOrCopyOfAlbum (
    const PAlbum *const album,
    const PAlbum *const dstAlbum,
    const QString & newAlbumName = QString() )
```

Hint at the imminent copy, move or rename of an album, so that the collection scanner is informed about this. If the album is renamed, give the new name in newAlbumName. DstAlbum is the new parent album / dstPath is the new parent directory of the album, so do not include the album name to dstPath.

### 6.1242.1.12 hintAtMoveOrCopyOfItems()

```
void Digikam::ScanController::hintAtMoveOrCopyOfItems (
    const QList< qlonglong > & ids,
    const PAlbum *const dstAlbum,
    const QStringList & itemNames )
```

Hint at the imminent copy, move or rename of items, so that the collection scanner is informed about this. Give the list of existing items, specify the destination with dstAlbum, and give the names at destination in itemNames (Unless for rename, names wont usually change. Give them nevertheless.)

### 6.1242.1.13 restart()

```
void Digikam::ScanController::restart ( )
```

Restart thread after shutdown.

### 6.1242.1.14 restartCollectionScan()

```
void Digikam::ScanController::restartCollectionScan ( )
```

Restart a suspended collection scanning. All scheduled scanning tasks are queued and will be done when [restartCollectionScan\(\)](#) has been called.

### 6.1242.1.15 resumeCollectionScan()

```
void Digikam::ScanController::resumeCollectionScan ( )
```

Resume a suspended collection scanning. All scheduled scanning tasks are queued and will be done when [resumeCollectionScan\(\)](#) has been called. Calling these methods is recursive, you must resume as often as you called suspend.

### 6.1242.1.16 scannedInfo()

```
ItemInfo Digikam::ScanController::scannedInfo (
    const QString & filePath,
    CollectionScanner::FileScanMode mode = CollectionScanner::NormalScan )
```

If necessary (modified or newly created, scans the file directly Returns the up-to-date [ItemInfo](#).

#### 6.1242.1.17 `scheduleCollectionScan()`

```
void Digikam::ScanController::scheduleCollectionScan (
    const QString & path )
```

Schedules a scan of the specified part of the collection. Asynchronous, returns immediately.

#### 6.1242.1.18 `scheduleCollectionScanExternal()`

```
void Digikam::ScanController::scheduleCollectionScanExternal (
    const QString & path )
```

Schedules a scan of the specified part of the collection. Asynchronous, returns immediately. A very long delay with timer restart may be introduced before the actual scanning starts, so that you can call this often without checking for duplicates. This method is only for the `QFileSystemWatcher`.

#### 6.1242.1.19 `scheduleCollectionScanRelaxed()`

```
void Digikam::ScanController::scheduleCollectionScanRelaxed (
    const QString & path )
```

Schedules a scan of the specified part of the collection. Asynchronous, returns immediately. A small delay may be introduced before the actual scanning starts, so that you can call this often without checking for duplicates. This method must only be used from the main thread.

#### 6.1242.1.20 `shutDown()`

```
void Digikam::ScanController::shutDown ( )
```

Wait for the thread to finish. Returns after all tasks are done.

#### 6.1242.1.21 `suspendCollectionScan()`

```
void Digikam::ScanController::suspendCollectionScan ( )
```

Temporarily suspend collection scanning. All scheduled scanning tasks are queued and will be done when [resumeCollectionScan\(\)](#) has been called. Calling these methods is recursive, you must resume as often as you called suspend.

#### 6.1242.1.22 `updateUniqueHash()`

```
void Digikam::ScanController::updateUniqueHash ( )
```

Carries out a complete collection scan, at the same time updating the unique hash in the database and thumbnail database. Synchronous, returns when ready. The database will be locked while the scan is running.

## 6.1243 Digikam::ScanController::FileMetadataWrite Class Reference

### Public Member Functions

- void **changed** (bool wasChanged)
- **FileMetadataWrite** (const [ItemInfo](#) &info)

### Protected Attributes

- bool **m\_changed** = false
- [ItemInfo](#) **m\_info**

### 6.1243.1 Detailed Description

When writing metadata to the file, the file content on disk changes, but the information is taken from the database; therefore, the resulting scanning process can be optimized.

Thus, if you write metadata of an [ItemInfo](#) from the database to disk, do this in the scope of a [FileMetadataWrite](#) object.

## 6.1244 Digikam::ScanController::Private Class Reference

### Public Member Functions

- QPixmap **actionPixmap** ()
- QPixmap **albumPixmap** ()
- QPixmap **errorPixmap** ()
- void **garbageCollectHints** (bool setAccessTime)
- QPixmap **restartPixmap** ()
- QPixmap **rootPixmap** ()

### Public Attributes

- QPixmap **actionPix**
- ScanController::Advice **advice** = ScanController::Success
- QPixmap **albumPix**
- QStringList **completeScanDeferredAlbums**
- QWaitCondition **condVar**
- bool **continueInitialization** = false
- bool **continuePartialScan** = false
- bool **continueScan** = false
- bool **deferFileScanning** = false
- QPixmap **errorPix**
- QEventLoop \* **eventLoop** = nullptr
- QTimer \* **externalTimer** = nullptr
- bool **fileWatchInstalled** = false
- bool **finishScanAllowed** = true
- [CollectionScannerHintContainer](#) \* **hints** = [CollectionScanner::createHintContainer\(\)](#)
- bool **idle** = false

- QDateTime **lastHintAdded**
- QMutex **mutex**
- bool **needsCompleteScan** = false
- bool **needsInitialization** = false
- bool **needsUpdateUniqueHash** = false
- bool **needTotalFiles** = false
- QList< qulonglong > **newIdsList**
- bool **performFastScan** = true
- [DProgressDlg](#) \* **progressDialog** = nullptr
- QTimer \* **relaxedTimer** = nullptr
- QPixmap **rootPix**
- bool **running** = false
- int **scanSuspended** = 0
- QStringList **scanTasks**
- QTimer \* **showTimer** = nullptr
- int **totalFilesToScan** = 0

## 6.1245 Digikam::ScanControllerCreator Class Reference

### Public Attributes

- [ScanController](#) object



## 6.1246 Digikam::ScanControllerLoadingCacheFileWatch Class Reference

Inheritance diagram for Digikam::ScanControllerLoadingCacheFileWatch:



### Additional Inherited Members

### Public Member Functions inherited from [Digikam::LoadingCacheFileWatch](#)

- void **addedImage** (const QString &filePath)
- void **checkFileWatch** (const QString &filePath)
- void **removeImage** (const QString &filePath)

### Protected Member Functions inherited from [Digikam::LoadingCacheFileWatch](#)

- void [notifyFileChanged](#) (const QString &filePath)

## Protected Attributes inherited from [Digikam::LoadingCacheFileWatch](#)

- class [LoadingCache](#) \* `m_cache` = nullptr
- QHash< QString, QPair< qint64, QDateTime > > `m_watchHash`

## 6.1247 Digikam::ScanStateFilter Class Reference

Inheritance diagram for Digikam::ScanStateFilter:



## Signals

- void **signalInfosToDispatch** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Public Member Functions

- FacePipelineExtendedPackage::Ptr **filter** (const [ItemInfo](#) &info)
- void **process** (const [ItemInfo](#) &info)
- void **process** (const QList< [ItemInfo](#) > &infos)
- **ScanStateFilter** ([FacePipeline::FilterMode](#) fmode, [FacePipeline::Private](#) \*const dd)

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Public Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- [FacePipeline::FilterMode](#) **mode** = [FacePipeline::SkipAlreadyScanned](#)
- [FacePipelineFaceTagsIface::Roles](#) **tasks**

## Protected Slots

- void **dispatch** ()

## Protected Member Functions

- void **run** () override

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

### Protected Attributes

- `QList< ItemInfo > toBeSkipped`
- `QList< ItemInfo > toFilter`
- `QList< FacePipelineExtendedPackage::Ptr > toSend`

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum `State` { `Inactive` , `Scheduled` , `Running` , `Deactivating` }

### Public Slots inherited from [Digikam::DynamicThread](#)

- void `start` ()
- void `stop` ()
- void `wait` ()

## 6.1247.1 Member Function Documentation

### 6.1247.1.1 `run()`

```
void Digikam::ScanStateFilter::run ( ) [override], [protected], [virtual]
```

Implement this pure virtual function in your subclass.

Implements [Digikam::DynamicThread](#).

## 6.1248 [Digikam::SchemeManager](#) Class Reference

### Public Types

- enum `BackgroundRole` { `NormalBackground` = 0 , `AlternateBackground` = 1 , `ActiveBackground` = 2 , `LinkBackground` = 3 , `VisitedBackground` = 4 , `NegativeBackground` = 5 , `NeutralBackground` = 6 , `PositiveBackground` = 7 }
- enum `ColorSet` { `View` , `Window` , `Button` , `Selection` , `Tooltip` , `Complementary` }
- enum `DecorationRole` { `FocusColor` , `HoverColor` }
- enum `ForegroundRole` { `NormalText` = 0 , `InactiveText` = 1 , `ActiveText` = 2 , `LinkText` = 3 , `VisitedText` = 4 , `NegativeText` = 5 , `NeutralText` = 6 , `PositiveText` = 7 }
- enum `ShadeRole` { `LightShade` , `MidlightShade` , `MidShade` , `DarkShade` , `ShadowShade` }

## Public Member Functions

- QBrush [background](#) ([BackgroundRole=NormalBackground](#)) const
- QBrush [decoration](#) ([DecorationRole](#)) const
- QBrush [foreground](#) ([ForegroundRole=NormalText](#)) const
- [SchemeManager](#) & [operator=](#) (const [SchemeManager](#) &)
- [SchemeManager](#) (const [SchemeManager](#) &)
- [SchemeManager](#) (QPalette::ColorGroup state, [ColorSet](#) set=[View](#), KSharedConfigPtr config=KSharedConfigPtr())
- QColor [shade](#) ([ShadeRole](#)) const
- [~SchemeManager](#) ()=default

## Static Public Member Functions

- static void [adjustBackground](#) (QPalette &, [BackgroundRole](#) newRole=[NormalBackground](#), QPalette::ColorRole color=QPalette::Base, [ColorSet](#) set=[View](#), const KSharedConfigPtr &config=KSharedConfigPtr())
- static void [adjustForeground](#) (QPalette &, [ForegroundRole](#) newRole=[NormalText](#), QPalette::ColorRole color=QPalette::Text, [ColorSet](#) set=[View](#), const KSharedConfigPtr &config=KSharedConfigPtr())
- static int [contrast](#) ()
- static qreal [contrastF](#) (const KSharedConfigPtr &config=KSharedConfigPtr())
- static QPalette [createApplicationPalette](#) (const KSharedConfigPtr &config)
- static QColor [shade](#) (const QColor &, [ShadeRole](#))
- static QColor [shade](#) (const QColor &color, [ShadeRole](#) role, qreal [contrast](#), qreal chromaAdjust=0.0)

*Retrieve the requested shade color, using the specified color as the base color and the specified contrast.*

### 6.1248.1 Detailed Description

A set of methods used to work with colors.

[SchemeManager](#) currently provides access to the system color palette that the user has selected (in the future, it is expected to do more). It greatly expands on QPalette by providing five distinct "sets" with several color choices each, covering background, foreground, and decoration colors.

A [SchemeManager](#) instance represents colors corresponding to a "set", where a set consists of those colors used to draw a particular type of element, such as a menu, button, view, selected text, or tooltip. Each set has a distinct set of colors, so you should always use the correct set for drawing and never assume that a particular foreground for one set is the same as the foreground for any other set. Individual colors may be quickly referenced by creating an anonymous instance and invoking a lookup member.

#### Note

The color palettes for the various states of a widget (active, inactive, disabled) may be wildly different. Therefore, it is important to take the state into account. This is why the [SchemeManager](#) constructor requires a QPalette::ColorGroup as an argument.

To facilitate working with potentially-varying states, two convenience API's are provided. These are [SchemeManager::adjustBackground](#) and its sister [SchemeManager::adjustForeground](#), and the helper class KStatefulBrush.

#### See also

[SchemeManager::ColorSet](#), [SchemeManager::ForegroundRole](#), [SchemeManager::BackgroundRole](#), [SchemeManager::DecorationRole](#), [SchemeManager::ShadeRole](#)

## 6.1248.2 Member Enumeration Documentation

### 6.1248.2.1 BackgroundRole

enum [Digikam::SchemeManager::BackgroundRole](#)

This enumeration describes the background color being selected from the given set.

Background colors are suitable for drawing under text, and should never be used to draw text. In combination with one of the overloads of [SchemeManager::shade](#), they may be used to generate colors for drawing frames, bevels, and similar decorations.

#### Enumerator

NormalBackground	Normal background.
AlternateBackground	Alternate background; for example, for use in lists. This color may be the same as BackgroundNormal, especially in sets other than View and Window.
ActiveBackground	Third color; for example, items which are new, active, requesting attention, etc. Alerting the user that a certain field must be filled out would be a good usage (although NegativeBackground could be used to the same effect, depending on what you are trying to achieve). Unlike ActiveText, this should not be used for mouseover effects.
LinkBackground	Fourth color; corresponds to (unvisited) links. Exactly what this might be used for is somewhat harder to qualify; it might be used for bookmarks, as a 'you can click here' indicator, or to highlight recent content (i.e. in a most-recently-accessed list).
VisitedBackground	Fifth color; corresponds to visited links. This can also be used to indicate "not recent" content, especially when a color is needed to denote content which is "old" or "archival".
NegativeBackground	Sixth color; for example, errors, untrusted content, etc.
NeutralBackground	Seventh color; for example, warnings, secure/encrypted content.
PositiveBackground	Eighth color; for example, success messages, trusted content.

### 6.1248.2.2 ColorSet

enum [Digikam::SchemeManager::ColorSet](#)

This enumeration describes the color set for which a color is being selected.

Color sets define a color "environment", suitable for drawing all parts of a given region. Colors from different sets should not be combined.

#### Enumerator

View	Views; for example, frames, input fields, etc. If it contains things that can be selected, it is probably a View.
Window	Non-editable window elements; for example, menus. If it isn't a Button, View, or Tooltip, it is probably a Window.
Button	Buttons and button-like controls. In addition to buttons, "button-like" controls such as non-editable dropdowns, scrollbar sliders, slider handles, etc. should also use this role.

## Enumerator

Selection	Selected items in views. Note that unfocused or disabled selections should use the Window role. This makes it more obvious to the user that the view containing the selection does not have input focus.
Tooltip	Tooltips. The tooltip set can often be substituted for the view set when editing is not possible, but the Window set is deemed inappropriate. "What's This" help is an excellent example, another might be pop-up notifications (depending on taste).
Complementary	Complementary areas. Some applications want some areas to have a different color scheme. Usually dark areas over a light theme. For instance the fullscreen UI of a picture viewer, or the logout/lock screen of the plasma workspace ask for a dark color scheme even on light themes.

**6.1248.2.3 DecorationRole**

```
enum Digikam::SchemeManager::DecorationRole
```

This enumeration describes the decoration color being selected from the given set.

Decoration colors are used to draw decorations (such as frames) for special purposes. Like color shades, they are neither foreground nor background colors. Text should not be painted over a decoration color, and decoration colors should not be used to draw text.

## Enumerator

FocusColor	Color used to draw decorations for items which have input focus.
HoverColor	Color used to draw decorations for items which will be activated by clicking.

**6.1248.2.4 ForegroundRole**

```
enum Digikam::SchemeManager::ForegroundColor
```

This enumeration describes the foreground color being selected from the given set.

Foreground colors are suitable for drawing text or glyphs (such as the symbols on window decoration buttons, assuming a suitable background brush is used), and should never be used to draw backgrounds.

For window decorations, the following is suggested, but not set in stone:

- Maximize - PositiveText
- Minimize - NeutralText
- Close - NegativeText
- WhatsThis - LinkText
- Sticky - ActiveText

## Enumerator

NormalText	Normal foreground.
InactiveText	Second color; for example, comments, items which are old, inactive or disabled. Generally used for things that are meant to be "less important". InactiveText is not the same role as NormalText in the inactive state.
ActiveText	Third color; for example items which are new, active, requesting attention, etc. May be used as a hover color for clickable items.
LinkText	Fourth color; use for (unvisited) links. May also be used for other clickable items or content that indicates relationships, items that indicate somewhere the user can visit, etc.
VisitedText	Fifth color; used for (visited) links. As with LinkText, may be used for items that have already been "visited" or accessed. May also be used to indicate "historical" (i.e. "old") items or information, especially if InactiveText is being used in the same context to express something different.
NegativeText	Sixth color; for example, errors, untrusted content, deletions, etc.
NeutralText	Seventh color; for example, warnings, secure/encrypted content.
PositiveText	Eighth color; for example, additions, success messages, trusted content.

**6.1248.2.5 ShadeRole**

```
enum Digikam::SchemeManager::ShadeRole
```

This enumeration describes the color shade being selected from the given set.

Color shades are used to draw "3d" elements, such as frames and bevels. They are neither foreground nor background colors. Text should not be painted over a shade, and shades should not be used to draw text.

## Enumerator

LightShade	The light color is lighter than dark() or shadow() and contrasts with the base color.
MidlightShade	The midlight color is in between base() and light().
MidShade	The mid color is in between base() and dark().
DarkShade	The dark color is in between mid() and shadow().
ShadowShade	The shadow color is darker than light() or midlight() and contrasts the base color.

**6.1248.3 Constructor & Destructor Documentation****6.1248.3.1 SchemeManager() [1/2]**

```
Digikam::SchemeManager::SchemeManager (
    const SchemeManager & other )
```

Construct a copy of another [SchemeManager](#).

**6.1248.3.2 ~SchemeManager()**

```
Digikam::SchemeManager::~SchemeManager ( ) [default]
```

Destructor



### 6.1248.3.3 SchemeManager() [2/2]

```
Digikam::SchemeManager::SchemeManager (
    QPalette::ColorGroup state,
    ColorSet set = View,
    KSharedConfigPtr config = KSharedConfigPtr() ) [explicit]
```

Construct a palette from given color set and state, using the colors from the given KConfig (if null, the system colors are used).

## 6.1248.4 Member Function Documentation

### 6.1248.4.1 adjustBackground()

```
void Digikam::SchemeManager::adjustBackground (
    QPalette & palette,
    BackgroundRole newRole = NormalBackground,
    QPalette::ColorRole color = QPalette::Base,
    ColorSet set = View,
    const KSharedConfigPtr & config = KSharedConfigPtr() ) [static]
```

Adjust a QPalette by replacing the specified QPalette::ColorRole with the requested background color for all states. Using this method is safer than replacing individual states, as it insulates you against changes in QPalette::Color↔Group.

#### Note

Although it is possible to replace a foreground color using this method, it's bad usability to do so. Just say "no".

### 6.1248.4.2 adjustForeground()

```
void Digikam::SchemeManager::adjustForeground (
    QPalette & palette,
    ForegroundRole newRole = NormalText,
    QPalette::ColorRole color = QPalette::Text,
    ColorSet set = View,
    const KSharedConfigPtr & config = KSharedConfigPtr() ) [static]
```

Adjust a QPalette by replacing the specified QPalette::ColorRole with the requested foreground color for all states. Using this method is safer than replacing individual states, as it insulates you against changes in QPalette::Color↔Group.

#### Note

Although it is possible to replace a background color using this method, it's bad usability to do so. Just say "no".

**6.1248.4.3 background()**

```
QBrush Digikam::SchemeManager::background (
    BackgroundRole role = NormalBackground ) const
```

Retrieve the requested background brush.

**6.1248.4.4 contrast()**

```
int Digikam::SchemeManager::contrast ( ) [static]
```

Returns the contrast for borders.

**Returns**

the contrast (between 0 for minimum and 10 for maximum contrast)

**6.1248.4.5 contrastF()**

```
qreal Digikam::SchemeManager::contrastF (
    const KSharedConfigPtr & config = KSharedConfigPtr() ) [static]
```

Returns the contrast for borders as a floating point value.

**Parameters**

<i>config</i>	pointer to the config from which to read the contrast setting (the default is to use KSharedConfig::openConfig())
---------------	---

**Returns**

the contrast (between 0.0 for minimum and 1.0 for maximum contrast)

**6.1248.4.6 createApplicationPalette()**

```
QPalette Digikam::SchemeManager::createApplicationPalette (
    const KSharedConfigPtr & config ) [static]
```

Used to obtain the QPalette that will be used to set the application palette from desktop platform theme.

**Parameters**

<i>config</i>	KConfig from which to load the colors
---------------	---------------------------------------

**Returns**

the QPalette

**6.1248.4.7 decoration()**

```
QBrush Digikam::SchemeManager::decoration (
    DecorationRole role ) const
```

Retrieve the requested decoration brush.

**6.1248.4.8 foreground()**

```
QBrush Digikam::SchemeManager::foreground (
    ForegroundRole role = NormalText ) const
```

Retrieve the requested foreground brush.

**6.1248.4.9 operator=()**

```
SchemeManager & Digikam::SchemeManager::operator= (
    const SchemeManager & other )
```

Standard assignment operator

**6.1248.4.10 shade() [1/3]**

```
QColor Digikam::SchemeManager::shade (
    const QColor & color,
    ShadeRole role ) [static]
```

Retrieve the requested shade color, using the specified color as the base color and the system contrast setting.

**Note**

Shades are chosen such that all shades would contrast with the base color. This means that if base is very dark, the 'dark' shades will be lighter than the base color, with `midlight() == shadow()`. Conversely, if the base color is very light, the 'light' shades will be darker than the base color, with `light() == mid()`.

**6.1248.4.11 shade() [2/3]**

```
QColor Digikam::SchemeManager::shade (
    const QColor & color,
    ShadeRole role,
    qreal contrast,
    qreal chromaAdjust = 0.0 ) [static]
```

**Parameters**

<i>color</i>	The color base color
<i>role</i>	The color role
<i>contrast</i>	Amount roughly specifying the contrast by which to adjust the base color, between -1.0 and 1.0 (values between 0.0 and 1.0 correspond to the value from <a href="#">SchemeManager::contrastF</a> )
<i>chromaAdjust</i>	(optional) Amount by which to adjust the chroma of the shade (1.0 means no adjustment)

**Note**

Shades are chosen such that all shades would contrast with the base color. This means that if base is very dark, the 'dark' shades will be lighter than the base color, with `midlight() == shadow()`. Conversely, if the base color is very light, the 'light' shades will be darker than the base color, with `light() == mid()`.

**6.1248.4.12 shade() [3/3]**

```
QColor Digikam::SchemeManager::shade (
    ShadeRole role ) const
```

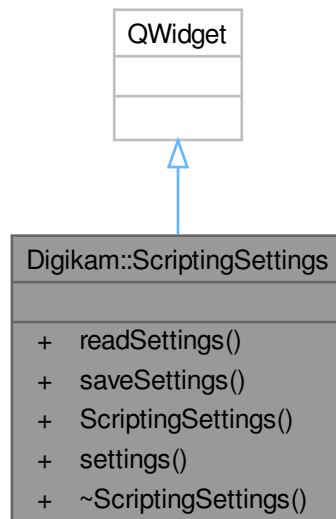
Retrieve the requested shade color, using `SchemeManager::background(SchemeManager::NormalBackground)` as the base color and the contrast setting from the `KConfig` used to create this `SchemeManager` instance (the system contrast setting, if no `KConfig` was specified).

**Note**

Shades are chosen such that all shades would contrast with the base color. This means that if base is very dark, the 'dark' shades will be lighter than the base color, with `midlight() == shadow()`. Conversely, if the base color is very light, the 'light' shades will be darker than the base color, with `light() == mid()`.

**6.1249 Digikam::ScriptingSettings Class Reference**

Inheritance diagram for `Digikam::ScriptingSettings`:

**Public Member Functions**

- void **readSettings** (const `KConfigGroup` &group)
- void **saveSettings** (`KConfigGroup` &group)
- **ScriptingSettings** (`QWidget` \*const parent=nullptr)
- void **settings** (`DownloadSettings` \*const settings) const

## 6.1250 Digikam::SearchChangeset Class Reference

### Public Types

- enum **Operation** { **Unknown** , **Added** , **Deleted** , **Changed** }

### Public Member Functions

- Operation **operation** () const
- **SearchChangeset** (int searchId, Operation operation)
- int **searchId** () const

## 6.1251 Digikam::SearchesDBJobInfo Class Reference

Inheritance diagram for Digikam::SearchesDBJobInfo:



### Public Member Functions

- `const QSet< qlonglong > & imagelds () const`
- `bool isAlbumUpdate () const`

- bool **isDuplicatesJob** () const
- double **maxThreshold** () const
- double **minThreshold** () const
- const QSet< qlonglong > & **refImagelds** () const
- [Haarface::ReflmageSelMethod](#) **refImageSelectionMethod** () const
- **SearchesDBJobInfo** (QList< int > &&searchIds)
- **SearchesDBJobInfo** (QSet< qlonglong > &&imagelds, bool isAlbumUpdate, [Haarface::ReflmageSelMethod](#) referenceSelectionMethod, QSet< qlonglong > &&refImagelds)
- const QList< int > & **searchIds** () const
- int **searchResultRestriction** () const
- void **setMaxThreshold** (double t)
- void **setMinThreshold** (double t)
- void **setSearchResultRestriction** (int type)

### Public Member Functions inherited from [Digikam::DBJobInfo](#)

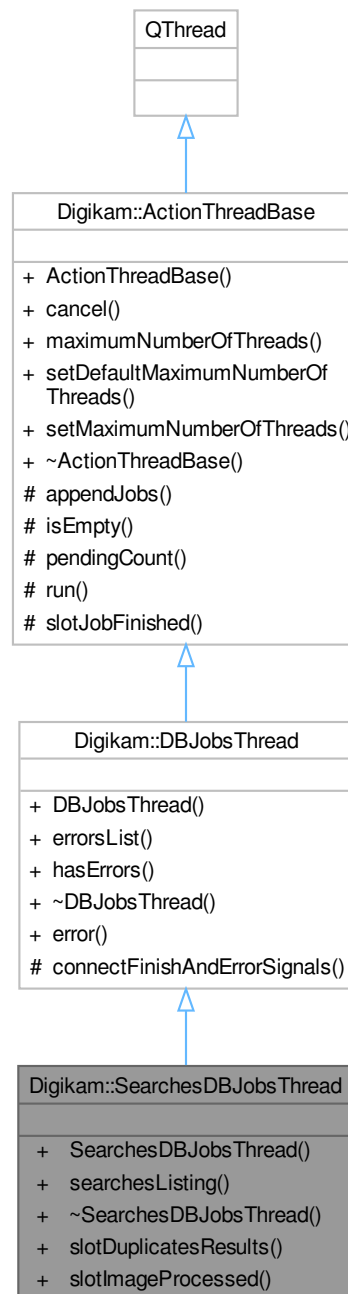
- bool **isFoldersJob** () const
- bool **isListAvailableImagesOnly** () const
- bool **isRecursive** () const
- void **setFoldersJob** ()
- void **setListAvailableImagesOnly** ()
- void **setRecursive** ()

### Public Attributes

- bool **m\_albumUpdate** = false
- bool **m\_duplicates** = false
- QSet< qlonglong > **m\_imagelds**
- double **m\_maxThreshold** = 1.0
- double **m\_minThreshold** = 0.4
- QSet< qlonglong > **m\_refImagelds**  
*Image ids of the reference images if duplicates are available.*
- [Haarface::ReflmageSelMethod](#) **m\_refImageSelectionMethod** = [Haarface::ReflmageSelMethod::OlderOrLarger](#)
- QList< int > **m\_searchIds**
- int **m\_searchResultRestriction** = 0

## 6.1252 Digikam::SearchesDBJobsThread Class Reference

Inheritance diagram for Digikam::SearchesDBJobsThread:



### Public Slots

- void **slotDuplicatesResults** (const `Haarface::DuplicatesResultsMap` &)
- void **slotImageProcessed** (const `ItemInfo` &, const `QImage` &, int dup)



## Public Slots inherited from [Digikam::DBJobsThread](#)

- void [error](#) (const QString &errString)  
*Appends the error string to m\_errorsList.*

## Signals

- void [signalProgress](#) (int percentage, const [ItemInfo](#) &inf, const QImage &img, int dup)

## Signals inherited from [Digikam::DBJobsThread](#)

- void [data](#) (const QList< [ItemListerRecord](#) > &records)
- void [finished](#) ()

## Public Member Functions

- [SearchesDBJobsThread](#) (QObject \*const parent)
- void [searchesListing](#) (const [SearchesDBJobInfo](#) &info)  
*Starts searches listing and scanning.*

## Public Member Functions inherited from [Digikam::DBJobsThread](#)

- [DBJobsThread](#) (QObject \*const parent)
- QList< QString > & [errorsList](#) ()  
*A method to get all errors reported from jobs.*
- bool [hasErrors](#) ()  
*hasErrors: a method to check for jobs errors*

## Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- [ActionThreadBase](#) (QObject \*const parent=nullptr)
- void [cancel](#) (bool isCancel=true)
- int [maximumNumberOfThreads](#) () const
- void [setDefaultMaximumNumberOfThreads](#) ()
- void [setMaximumNumberOfThreads](#) (int n)

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void [slotJobFinished](#) ()

## Protected Member Functions inherited from [Digikam::DBJobsThread](#)

- void [connectFinishAndErrorSignals](#) (DBJob \*const j)  
*Connects the signals of job to the signals of the thread.*

## Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void [appendJobs](#) (const [ActionJobCollection](#) &jobs)
- bool [isEmpty](#) () const
- int [pendingCount](#) () const
- void [run](#) () override

### 6.1252.1 Member Function Documentation

#### 6.1252.1.1 [searchesListing\(\)](#)

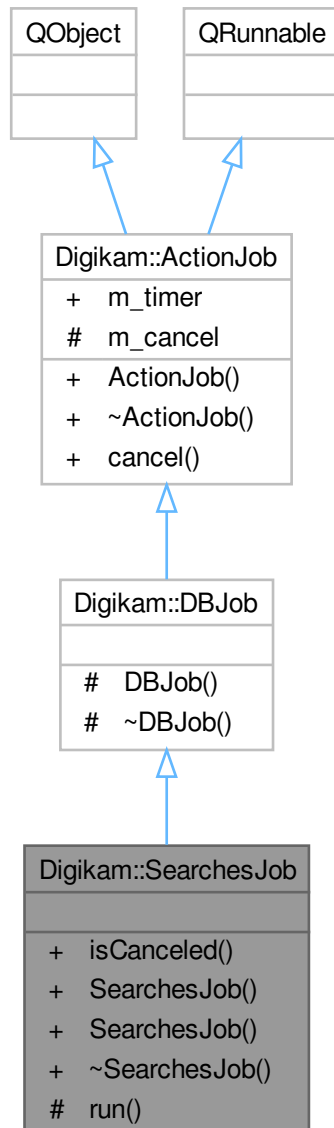
```
void Digikam::SearchesDBJobsThread::searchesListing (  
    const SearchesDBJobInfo & info )
```

##### Parameters

<i>info</i>	represents the searches job info
-------------	----------------------------------

## 6.1253 Digikam::SearchesJob Class Reference

Inheritance diagram for Digikam::SearchesJob:



### Signals

- void **signalDuplicatesResults** (const Haarlfacce::DuplicatesResultsMap &)
- void **signalImageProcessed** (const [ItemInfo](#) &, const QImage &, int dup)

### Signals inherited from [Digikam::DBJob](#)

- void **data** (const QList< [ItemListerRecord](#) > &records)
- void **error** (const QString &err)



## Public Types

- enum **WidgetRectType** { **LabelAndValueWidgetRects** , **ValueWidgetRectsOnly** }

## Signals

- void **signalVisibilityChanged** ()

## Public Member Functions

- bool **isVisible** () override
- virtual void **read** ([SearchXmlCachingReader](#) &reader)=0
- virtual void **reset** ()=0
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** ([SearchField](#) \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const
- virtual void **write** ([SearchXmlWriter](#) &writer)=0

## Static Public Member Functions

- static [SearchField](#) \* **createField** (const QString &fieldName, [SearchFieldGroup](#) \*const parent)

## Protected Slots

- void **clearButtonClicked** ()

## Protected Member Functions

- virtual void **setupLabels** (QGridLayout \*layout, int line)
- virtual void **setupValueWidgets** (QGridLayout \*layout, int row, int column)=0
- void **setValidValueState** (bool valuesValid)
- virtual void **setValueWidgetsVisible** (bool visible)=0
- virtual QList< QRect > **valueWidgetRects** () const =0

## Protected Attributes

- bool **m\_categoryLabelVisible** = true
- [AnimatedClearButton](#) \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

## 6.1254.1 Member Function Documentation

### 6.1254.1.1 createField()

```
SearchField * Digikam::SearchField::createField (
    const QString & fieldName,
    SearchFieldGroup *const parent ) [static]
```

### 6.1254.1.2 isVisible()

```
bool Digikam::SearchField::isVisible ( ) [override], [virtual]
```

Implements [Digikam::VisibilityObject](#).

### 6.1254.1.3 setVisible()

```
void Digikam::SearchField::setVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::VisibilityObject](#).

### 6.1254.1.4 write()

```
virtual void Digikam::SearchField::write (
    SearchXmlWriter & writer ) [pure virtual]
```

Implemented in [Digikam::SearchFieldRangeInt](#).

## 6.1255 Digikam::SearchFieldAlbum Class Reference

Inheritance diagram for Digikam::SearchFieldAlbum:



### Public Types

- enum **Operation** { **All** , **OneOf** , **InTree** }
- enum **Type** { **TypeAlbum** , **TypeTag** }

## Public Types inherited from [Digikam::SearchField](#)

- enum **WidgetRectType** { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Public Member Functions

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- void [reset](#) () override
- **SearchFieldAlbum** (QObject \*const parent, Type type)
- void [setupValueWidgets](#) (QGridLayout \*layout, int row, int column) override
- void [setValueWidgetsVisible](#) (bool visible) override
- QList< QRect > [valueWidgetRects](#) () const override
- void [write](#) ([SearchXmlWriter](#) &writer) override

## Public Member Functions inherited from [Digikam::SearchField](#)

- bool [isVisible](#) () override
- **SearchField** (QObject \*const parent)
- void [setCategoryLabelVisible](#) (bool visible)
- void [setCategoryLabelVisibleFromPreviousField](#) ([SearchField](#) \*const previousField)
- void [setFieldName](#) (const QString &fieldName)
- virtual void [setText](#) (const QString &label, const QString &detailLabel)
- void [setup](#) (QGridLayout \*const layout, int row=-1)
- void [setVisible](#) (bool visible) override
- virtual bool [supportsField](#) (const QString &fieldName)
- QList< QRect > [widgetRects](#) (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void [updateState](#) ()

## Protected Slots inherited from [Digikam::SearchField](#)

- void [clearButtonClicked](#) ()

## Protected Attributes

- [AlbumTreeViewSelectComboBox](#) \* [m\\_albumComboBox](#) = nullptr
- [AbstractCheckableAlbumModel](#) \* [m\\_model](#) = nullptr
- [SqueezedComboBox](#) \* [m\\_operation](#) = nullptr
- [TagTreeViewSelectComboBox](#) \* [m\\_tagComboBox](#) = nullptr
- Type [m\\_type](#) = TypeAlbum
- QWidget \* [m\\_wrapperBox](#) = nullptr

## Protected Attributes inherited from [Digikam::SearchField](#)

- bool [m\\_categoryLabelVisible](#) = true
- [AnimatedClearButton](#) \* [m\\_clearButton](#) = nullptr
- QLabel \* [m\\_detailLabel](#) = nullptr
- QLabel \* [m\\_label](#) = nullptr
- QString [m\\_name](#)
- bool [m\\_valuesValid](#) = false



## Additional Inherited Members

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const` parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

## 6.1255.1 Member Function Documentation

### 6.1255.1.1 `read()`

```
void Digikam::SearchFieldAlbum::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1255.1.2 `reset()`

```
void Digikam::SearchFieldAlbum::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1255.1.3 `setupValueWidgets()`

```
void Digikam::SearchFieldAlbum::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1255.1.4 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldAlbum::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

**6.1255.1.5 valueWidgetRects()**

```
QList< QRect > Digikam::SearchFieldAlbum::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

**6.1255.1.6 write()**

```
void Digikam::SearchFieldAlbum::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1256 Digikam::SearchFieldCheckBox Class Reference

Inheritance diagram for Digikam::SearchFieldCheckBox:



### Public Member Functions

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- void [reset](#) () override

- **SearchFieldCheckBox** (QObject \*const parent)
- void **setLabel** (const QString &text)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

### Public Member Functions inherited from [Digikam::SearchField](#)

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** ([SearchField](#) \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

### Protected Slots

- void **slotToggled** (bool checked)

### Protected Slots inherited from [Digikam::SearchField](#)

- void **clearButtonClicked** ()

### Protected Attributes

- QCheckBox \* **m\_checkBox** = nullptr
- QString **m\_text**

### Protected Attributes inherited from [Digikam::SearchField](#)

- bool **m\_categoryLabelVisible** = true
- [AnimatedClearButton](#) \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

### Additional Inherited Members

### Public Types inherited from [Digikam::SearchField](#)

- enum **WidgetRectType** { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const` parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

## 6.1256.1 Member Function Documentation

### 6.1256.1.1 `read()`

```
void Digikam::SearchFieldCheckBox::read (  
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1256.1.2 `reset()`

```
void Digikam::SearchFieldCheckBox::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1256.1.3 `setValueWidgets()`

```
void Digikam::SearchFieldCheckBox::setValueWidgets (  
    QGridLayout * layout,  
    int row,  
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1256.1.4 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldCheckBox::setValueWidgetsVisible (  
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

**6.1256.1.5 valueWidgetRects()**

```
QList< QRect > Digikam::SearchFieldCheckBox::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

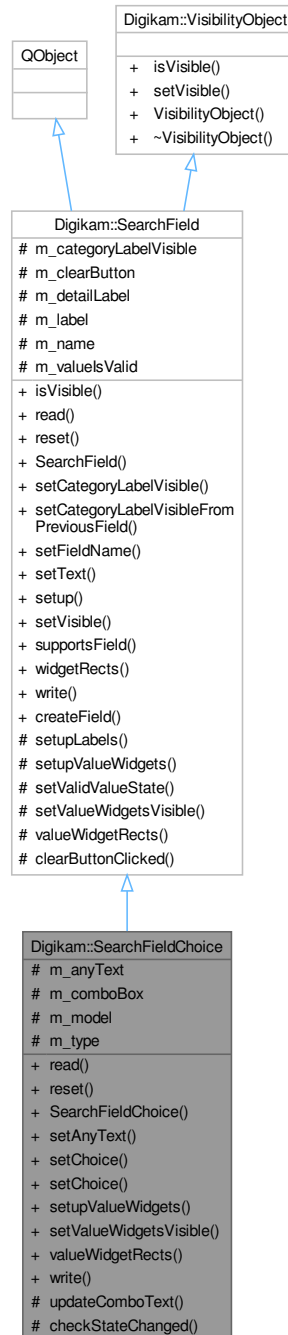
**6.1256.1.6 write()**

```
void Digikam::SearchFieldCheckBox::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1257 Digikam::SearchFieldChoice Class Reference

Inheritance diagram for Digikam::SearchFieldChoice:



### Public Member Functions

- void `read` (`SearchXmlCachingReader &reader`) override
- void `reset` () override

- **SearchFieldChoice** (QObject \*const parent)
- void **setAnyText** (const QString &anyText)
- void **setChoice** (const QMap< int, QString > &map)
- void **setChoice** (const QStringList &choice)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

### Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** (SearchField \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

### Protected Slots

- void **checkStateChanged** ()

### Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

### Protected Member Functions

- void **updateComboText** ()

### Protected Member Functions inherited from Digikam::SearchField

- virtual void **setupLabels** (QGridLayout \*layout, int line)
- void **setValidValueState** (bool valuesValid)

### Protected Attributes

- QString **m\_anyText**
- ChoiceSearchComboBox \* **m\_comboBox** = nullptr
- ChoiceSearchModel \* **m\_model** = nullptr
- QMetaType::Type **m\_type** = QMetaType::UnknownType



## Protected Attributes inherited from [Digikam::SearchField](#)

- bool `m_categoryLabelVisible` = true
- [AnimatedClearButton](#) \* `m_clearButton` = nullptr
- [QLabel](#) \* `m_detailLabel` = nullptr
- [QLabel](#) \* `m_label` = nullptr
- [QString](#) `m_name`
- bool `m_valuelsValid` = false

## Additional Inherited Members

## Public Types inherited from [Digikam::SearchField](#)

- enum `WidgetRectType` { `LabelAndValueWidgetRects` , `ValueWidgetRectsOnly` }

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static [SearchField](#) \* `createField` (const [QString](#) &fieldName, [SearchFieldGroup](#) \*const parent)

## 6.1257.1 Member Function Documentation

### 6.1257.1.1 `read()`

```
void Digikam::SearchFieldChoice::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1257.1.2 `reset()`

```
void Digikam::SearchFieldChoice::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1257.1.3 `setupValueWidgets()`

```
void Digikam::SearchFieldChoice::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1257.1.4 setValueWidgetsVisible()

```
void Digikam::SearchFieldChoice::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1257.1.5 valueWidgetRects()

```
QList< QRect > Digikam::SearchFieldChoice::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

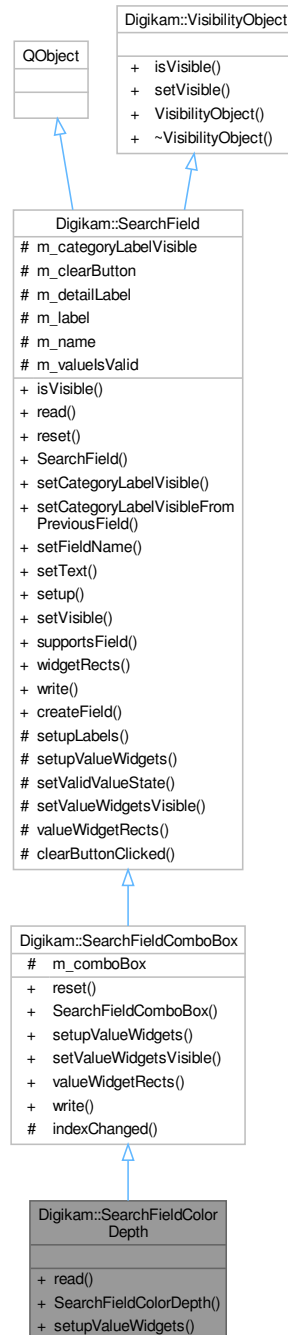
#### 6.1257.1.6 write()

```
void Digikam::SearchFieldChoice::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1258 Digikam::SearchFieldColorDepth Class Reference

Inheritance diagram for Digikam::SearchFieldColorDepth:



### Public Member Functions

- void `read` ([SearchXmlCachingReader](#) &reader) override
- `SearchFieldColorDepth` (`QObject` \*const parent)
- void `setValueWidgets` (`QGridLayout` \*layout, int row, int column) override

## Public Member Functions inherited from [Digikam::SearchFieldComboBox](#)

- void [reset](#) () override
- **SearchFieldComboBox** (QObject \*const parent)
- void [setupValueWidgets](#) (QGridLayout \*layout, int row, int column) override
- void [setValueWidgetsVisible](#) (bool visible) override
- QList< QRect > [valueWidgetRects](#) () const override
- void [write](#) ([SearchXmlWriter](#) &writer) override

## Public Member Functions inherited from [Digikam::SearchField](#)

- bool [isVisible](#) () override
- **SearchField** (QObject \*const parent)
- void [setCategoryLabelVisible](#) (bool visible)
- void [setCategoryLabelVisibleFromPreviousField](#) ([SearchField](#) \*const previousField)
- void [setFieldName](#) (const QString &fieldName)
- virtual void [setText](#) (const QString &label, const QString &detailLabel)
- void [setup](#) (QGridLayout \*const layout, int row=-1)
- void [setVisible](#) (bool visible) override
- virtual bool [supportsField](#) (const QString &fieldName)
- QList< QRect > [widgetRects](#) (WidgetRectType=ValueWidgetRectsOnly) const

## Additional Inherited Members

## Public Types inherited from [Digikam::SearchField](#)

- enum **WidgetRectType** { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Signals inherited from [Digikam::SearchField](#)

- void [signalVisibilityChanged](#) ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static [SearchField](#) \* [createField](#) (const QString &fieldName, [SearchFieldGroup](#) \*const parent)

## Protected Slots inherited from [Digikam::SearchFieldComboBox](#)

- void [indexChanged](#) (int)

## Protected Slots inherited from [Digikam::SearchField](#)

- void [clearButtonClicked](#) ()

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void [setupLabels](#) (QGridLayout \*layout, int line)
- void [setValidValueState](#) (bool valueIsValid)

## Protected Attributes inherited from [Digikam::SearchFieldComboBox](#)

- `QComboBox * m_comboBox = nullptr`

## Protected Attributes inherited from [Digikam::SearchField](#)

- `bool m_categoryLabelVisible = true`
- [AnimatedClearButton](#) \* `m_clearButton = nullptr`
- `QLabel * m_detailLabel = nullptr`
- `QLabel * m_label = nullptr`
- `QString m_name`
- `bool m_valuesValid = false`

## 6.1258.1 Member Function Documentation

### 6.1258.1.1 `read()`

```
void Digikam::SearchFieldColorDepth::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1258.1.2 `setupValueWidgets()`

```
void Digikam::SearchFieldColorDepth::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1259 Digikam::SearchFieldComboBox Class Reference

Inheritance diagram for Digikam::SearchFieldComboBox:



### Public Member Functions

- void `reset()` override
- `SearchFieldComboBox` (`QObject *const parent`)

- void [setupValueWidgets](#) (QGridLayout \*layout, int row, int column) override
- void [setValueWidgetsVisible](#) (bool visible) override
- QList< QRect > [valueWidgetRects](#) () const override
- void [write](#) (SearchXmlWriter &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool [isVisible](#) () override
- virtual void [read](#) (SearchXmlCachingReader &reader)=0
- **SearchField** (QObject \*const parent)
- void [setCategoryLabelVisible](#) (bool visible)
- void [setCategoryLabelVisibleFromPreviousField](#) (SearchField \*const previousField)
- void [setFieldName](#) (const QString &fieldName)
- virtual void [setText](#) (const QString &label, const QString &detailLabel)
- void [setup](#) (QGridLayout \*const layout, int row=-1)
- void [setVisible](#) (bool visible) override
- virtual bool [supportsField](#) (const QString &fieldName)
- QList< QRect > [widgetRects](#) (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void [indexChanged](#) (int)

## Protected Slots inherited from Digikam::SearchField

- void [clearButtonClicked](#) ()

## Protected Attributes

- QComboBox \* [m\\_comboBox](#) = nullptr

## Protected Attributes inherited from Digikam::SearchField

- bool [m\\_categoryLabelVisible](#) = true
- [AnimatedClearButton](#) \* [m\\_clearButton](#) = nullptr
- QLabel \* [m\\_detailLabel](#) = nullptr
- QLabel \* [m\\_label](#) = nullptr
- QString [m\\_name](#)
- bool [m\\_valuesValid](#) = false

## Additional Inherited Members

## Public Types inherited from Digikam::SearchField

- enum [WidgetRectType](#) { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const parent`)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

## 6.1259.1 Member Function Documentation

### 6.1259.1.1 `reset()`

```
void Digikam::SearchFieldComboBox::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1259.1.2 `setupValueWidgets()`

```
void Digikam::SearchFieldComboBox::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1259.1.3 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldComboBox::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1259.1.4 `valueWidgetRects()`

```
QList< QRect > Digikam::SearchFieldComboBox::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).



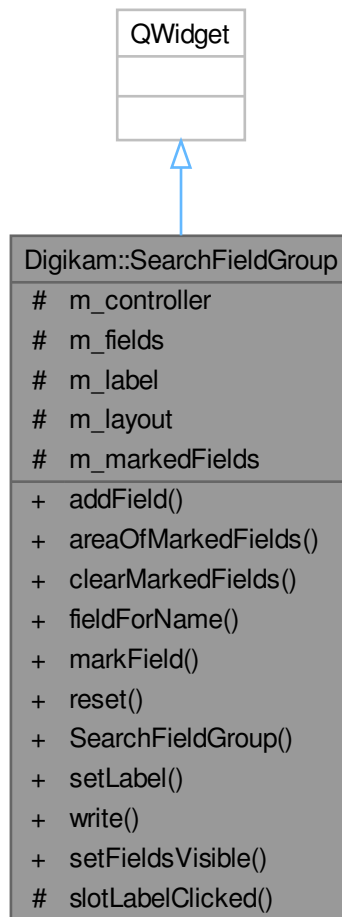
### 6.1259.1.5 write()

```
void Digikam::SearchFieldComboBox::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1260 Digikam::SearchFieldGroup Class Reference

Inheritance diagram for Digikam::SearchFieldGroup:



### Public Slots

- void **setFieldsVisible** (bool visible)

**Public Member Functions**

- void **addField** ([SearchField](#) \*const field)
- QList< QRect > **areaOfMarkedFields** () const
- void **clearMarkedFields** ()
- [SearchField](#) \* **fieldForName** (const QString &fieldName) const
- void **markField** ([SearchField](#) \*const field)
- void **reset** ()
- **SearchFieldGroup** ([SearchGroup](#) \*const parent)
- void **setLabel** ([SearchFieldGroupLabel](#) \*const label)
- void **write** ([SearchXmlWriter](#) &writer)

**Protected Slots**

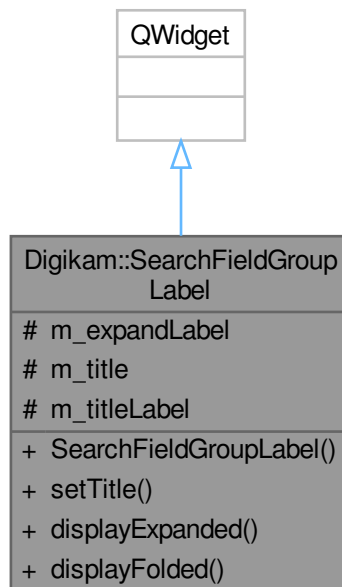
- void **slotLabelClicked** ()

**Protected Attributes**

- [VisibilityController](#) \* **m\_controller** = nullptr
- QList< [SearchField](#) \* > **m\_fields**
- [SearchFieldGroupLabel](#) \* **m\_label** = nullptr
- QGridLayout \* **m\_layout** = nullptr
- QSet< [SearchField](#) \* > **m\_markedFields**

## 6.1261 Digikam::SearchFieldGroupLabel Class Reference

Inheritance diagram for Digikam::SearchFieldGroupLabel:



### Public Slots

- void **displayExpanded** ()
- void **displayFolded** ()

### Signals

- void **clicked** ()

### Public Member Functions

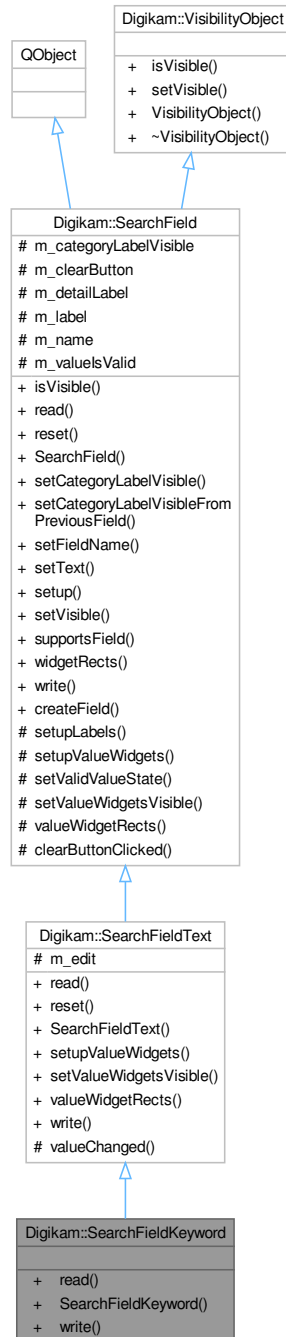
- **SearchFieldGroupLabel** (QWidget \*const parent)
- void **setTitle** (const QString &title)

### Protected Attributes

- QLabel \* **m\_expandLabel** = nullptr
- QString **m\_title**
- [DClickLabel](#) \* **m\_titleLabel** = nullptr

## 6.1262 Digikam::SearchFieldKeyword Class Reference

Inheritance diagram for Digikam::SearchFieldKeyword:



### Public Member Functions

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- [SearchFieldKeyword](#) (QObject \*const parent)
- void [write](#) ([SearchXmlWriter](#) &writer) override

## Public Member Functions inherited from Digikam::SearchFieldText

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- void [reset](#) () override
- **SearchFieldText** (QObject \*const parent)
- void [setupValueWidgets](#) (QGridLayout \*layout, int row, int column) override
- void [setValueWidgetsVisible](#) (bool visible) override
- QList< QRect > [valueWidgetRects](#) () const override
- void [write](#) ([SearchXmlWriter](#) &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool [isVisible](#) () override
- **SearchField** (QObject \*const parent)
- void [setCategoryLabelVisible](#) (bool visible)
- void [setCategoryLabelVisibleFromPreviousField](#) ([SearchField](#) \*const previousField)
- void [setFieldName](#) (const QString &fieldName)
- virtual void [setText](#) (const QString &label, const QString &detailLabel)
- void [setup](#) (QGridLayout \*const layout, int row=-1)
- void [setVisible](#) (bool visible) override
- virtual bool [supportsField](#) (const QString &fieldName)
- QList< QRect > [widgetRects](#) (WidgetRectType=ValueWidgetRectsOnly) const

## Additional Inherited Members

## Public Types inherited from Digikam::SearchField

- enum **WidgetRectType** { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Signals inherited from Digikam::SearchField

- void [signalVisibilityChanged](#) ()

## Static Public Member Functions inherited from Digikam::SearchField

- static [SearchField](#) \* [createField](#) (const QString &fieldName, [SearchFieldGroup](#) \*const parent)

## Protected Slots inherited from Digikam::SearchFieldText

- void [valueChanged](#) (const QString &text)

## Protected Slots inherited from Digikam::SearchField

- void [clearButtonClicked](#) ()

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void **setupLabels** (QGridLayout \*layout, int line)
- void **setValidValueState** (bool valueIsValid)

## Protected Attributes inherited from [Digikam::SearchFieldText](#)

- QLineEdit \* **m\_edit** = nullptr

## Protected Attributes inherited from [Digikam::SearchField](#)

- bool **m\_categoryLabelVisible** = true
- [AnimatedClearButton](#) \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valueIsValid** = false

## 6.1262.1 Member Function Documentation

### 6.1262.1.1 read()

```
void Digikam::SearchFieldKeyword::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1262.1.2 write()

```
void Digikam::SearchFieldKeyword::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1263 Digikam::SearchFieldLabels Class Reference

Inheritance diagram for Digikam::SearchFieldLabels:



### Public Member Functions

- void `read` (`SearchXmlCachingReader &reader`) override
- void `reset` () override

- **SearchFieldLabels** (QObject \*const parent)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** (SearchField \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void **updateState** ()

## Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

## Protected Attributes

- ColorLabelFilter \* **m\_colorLabelFilter** = nullptr
- PickLabelFilter \* **m\_pickLabelFilter** = nullptr

## Protected Attributes inherited from Digikam::SearchField

- bool **m\_categoryLabelVisible** = true
- AnimatedClearButton \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

## Additional Inherited Members

## Public Types inherited from Digikam::SearchField

- enum **WidgetRectType** { LabelAndValueWidgetRects , ValueWidgetRectsOnly }



## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const` parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

### 6.1263.1 Member Function Documentation

#### 6.1263.1.1 `read()`

```
void Digikam::SearchFieldLabels::read (  
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1263.1.2 `reset()`

```
void Digikam::SearchFieldLabels::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1263.1.3 `setValueWidgets()`

```
void Digikam::SearchFieldLabels::setValueWidgets (  
    QGridLayout * layout,  
    int row,  
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1263.1.4 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldLabels::setValueWidgetsVisible (  
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

**6.1263.1.5 valueWidgetRects()**

```
QList< QRect > Digikam::SearchFieldLabels::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

**6.1263.1.6 write()**

```
void Digikam::SearchFieldLabels::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1264 Digikam::SearchFieldMonthDay Class Reference

Inheritance diagram for Digikam::SearchFieldMonthDay:



### Public Member Functions

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- void [reset](#) () override

- **SearchFieldMonthDay** (QObject \*const parent)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** (SearchField \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void **slotIndexChanged** ()

## Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

## Protected Attributes

- QComboBox \* **m\_dayBox** = nullptr
- QLabel \* **m\_dayLabel** = nullptr
- QComboBox \* **m\_monthBox** = nullptr

## Protected Attributes inherited from Digikam::SearchField

- bool **m\_categoryLabelVisible** = true
- **AnimatedClearButton** \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

## Additional Inherited Members

## Public Types inherited from Digikam::SearchField

- enum **WidgetRectType** { **LabelAndValueWidgetRects** , **ValueWidgetRectsOnly** }

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const` parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

## 6.1264.1 Member Function Documentation

### 6.1264.1.1 `read()`

```
void Digikam::SearchFieldMonthDay::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1264.1.2 `reset()`

```
void Digikam::SearchFieldMonthDay::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1264.1.3 `setValueWidgets()`

```
void Digikam::SearchFieldMonthDay::setValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1264.1.4 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldMonthDay::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

**6.1264.1.5 valueWidgetRects()**

```
QList< QRect > Digikam::SearchFieldMonthDay::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

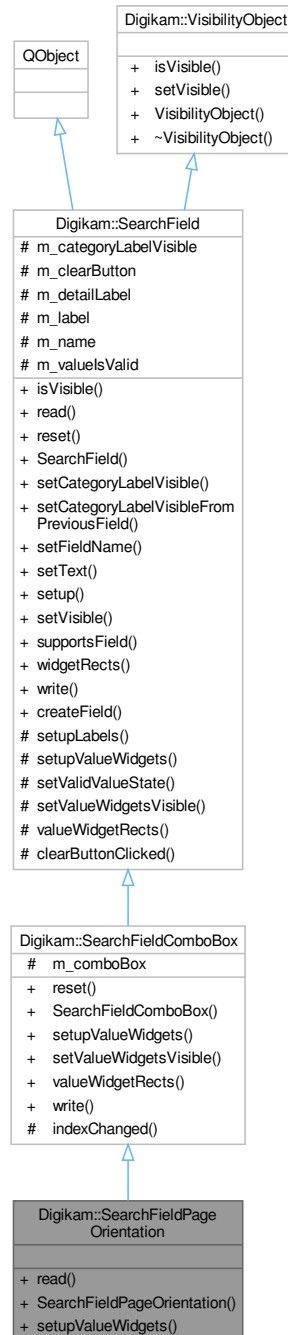
**6.1264.1.6 write()**

```
void Digikam::SearchFieldMonthDay::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1265 Digikam::SearchFieldPageOrientation Class Reference

Inheritance diagram for Digikam::SearchFieldPageOrientation:



### Public Member Functions

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- [SearchFieldPageOrientation](#) ([QObject](#) \*const parent)
- void [setValueWidgets](#) ([QGridLayout](#) \*layout, int row, int column) override

## Public Member Functions inherited from [Digikam::SearchFieldComboBox](#)

- void [reset](#) () override
- [SearchFieldComboBox](#) (QObject \*const parent)
- void [setValueWidgetsVisible](#) (bool visible) override
- QList< QRect > [valueWidgetRects](#) () const override
- void [write](#) ([SearchXmlWriter](#) &writer) override

## Public Member Functions inherited from [Digikam::SearchField](#)

- bool [isVisible](#) () override
- [SearchField](#) (QObject \*const parent)
- void [setCategoryLabelVisible](#) (bool visible)
- void [setCategoryLabelVisibleFromPreviousField](#) ([SearchField](#) \*const previousField)
- void [setFieldName](#) (const QString &fieldName)
- virtual void [setText](#) (const QString &label, const QString &detailLabel)
- void [setup](#) (QGridLayout \*const layout, int row=-1)
- void [setVisible](#) (bool visible) override
- virtual bool [supportsField](#) (const QString &fieldName)
- QList< QRect > [widgetRects](#) (WidgetRectType=ValueWidgetRectsOnly) const

## Additional Inherited Members

## Public Types inherited from [Digikam::SearchField](#)

- enum [WidgetRectType](#) { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Signals inherited from [Digikam::SearchField](#)

- void [signalVisibilityChanged](#) ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static [SearchField](#) \* [createField](#) (const QString &fieldName, [SearchFieldGroup](#) \*const parent)

## Protected Slots inherited from [Digikam::SearchFieldComboBox](#)

- void [indexChanged](#) (int)

## Protected Slots inherited from [Digikam::SearchField](#)

- void [clearButtonClicked](#) ()

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void [setupLabels](#) (QGridLayout \*layout, int line)
- void [setValidValueState](#) (bool valueIsValid)



## Protected Attributes inherited from [Digikam::SearchFieldComboBox](#)

- `QComboBox * m_comboBox = nullptr`

## Protected Attributes inherited from [Digikam::SearchField](#)

- `bool m_categoryLabelVisible = true`
- [AnimatedClearButton](#) \* `m_clearButton = nullptr`
- `QLabel * m_detailLabel = nullptr`
- `QLabel * m_label = nullptr`
- `QString m_name`
- `bool m_valuesValid = false`

## 6.1265.1 Member Function Documentation

### 6.1265.1.1 `read()`

```
void Digikam::SearchFieldPageOrientation::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

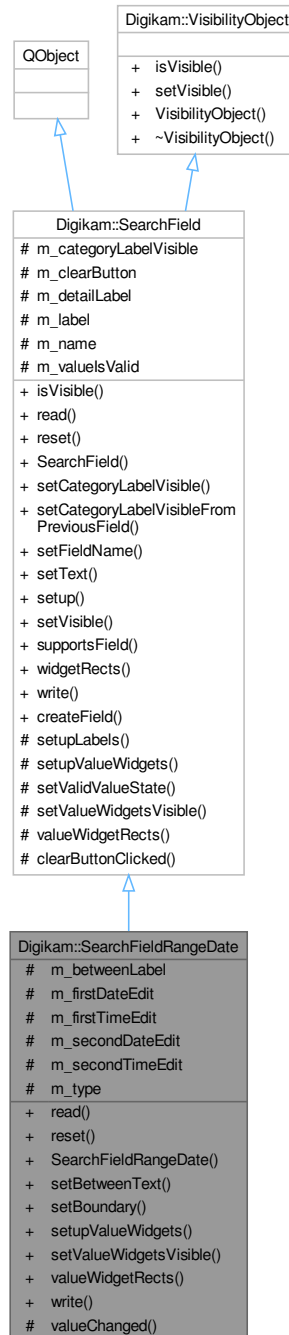
### 6.1265.1.2 `setupValueWidgets()`

```
void Digikam::SearchFieldPageOrientation::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Reimplemented from [Digikam::SearchFieldComboBox](#).

## 6.1266 Digikam::SearchFieldRangeDate Class Reference

Inheritance diagram for Digikam::SearchFieldRangeDate:



### Public Types

- enum **Type** { **DateOnly** , **DateTime** }

## Public Types inherited from Digikam::SearchField

- enum **WidgetRectType** { **LabelAndValueWidgetRects** , **ValueWidgetRectsOnly** }

## Public Member Functions

- void **read** ([SearchXmlCachingReader](#) &reader) override
- void **reset** () override
- **SearchFieldRangeDate** (QObject \*const parent, Type type)
- void **setBetweenText** (const QString &between)
- void **setBoundary** (const QDateTime &min, const QDateTime &max)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** ([SearchXmlWriter](#) &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** ([SearchField](#) \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void **valueChanged** ()

## Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

## Protected Attributes

- QLabel \* **m\_betweenLabel** = nullptr
- [DDateEdit](#) \* **m\_firstDateEdit** = nullptr
- [QTimeEdit](#) \* **m\_firstTimeEdit** = nullptr
- [DDateEdit](#) \* **m\_secondDateEdit** = nullptr
- [QTimeEdit](#) \* **m\_secondTimeEdit** = nullptr
- Type **m\_type** = DateOnly

## Protected Attributes inherited from [Digikam::SearchField](#)

- bool `m_categoryLabelVisible` = true
- [AnimatedClearButton](#) \* `m_clearButton` = nullptr
- QLabel \* `m_detailLabel` = nullptr
- QLabel \* `m_label` = nullptr
- QString `m_name`
- bool `m_valuelsValid` = false

## Additional Inherited Members

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static [SearchField](#) \* `createField` (const QString &fieldName, [SearchFieldGroup](#) \*const parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valuelsValid)

## 6.1266.1 Member Function Documentation

### 6.1266.1.1 read()

```
void Digikam::SearchFieldRangeDate::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1266.1.2 reset()

```
void Digikam::SearchFieldRangeDate::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1266.1.3 setupValueWidgets()

```
void Digikam::SearchFieldRangeDate::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1266.1.4 setValueWidgetsVisible()

```
void Digikam::SearchFieldRangeDate::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1266.1.5 valueWidgetRects()

```
QList< QRect > Digikam::SearchFieldRangeDate::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1266.1.6 write()

```
void Digikam::SearchFieldRangeDate::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1267 Digikam::SearchFieldRangeDouble Class Reference

Inheritance diagram for Digikam::SearchFieldRangeDouble:



### Public Member Functions

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- void [reset](#) () override

- **SearchFieldRangeDouble** (QObject \*const parent)
- void **setBetweenText** (const QString &text)
- void **setBoundary** (double min, double max, int decimals, double step)
- void **setFactor** (double factor)
- void **setInvertStepping** (bool invert)
- void **setNoValueText** (const QString &text)
- void **setNumberPrefixAndSuffix** (const QString &prefix, const QString &suffix)
- void **setSingleSteps** (double smaller, double larger)
- void **setSuggestedInitialValue** (double initialValue)
- void **setSuggestedValues** (const QList< double > &values)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** (SearchField \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgerRects** (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void **valueChanged** ()

## Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

## Protected Attributes

- QLabel \* **m\_betweenLabel** = nullptr
- double **m\_factor** = 1.0
- CustomStepsDoubleSpinBox \* **m\_firstBox** = nullptr
- double **m\_max** = 100.0
- double **m\_min** = 0.0
- CustomStepsDoubleSpinBox \* **m\_secondBox** = nullptr

## Protected Attributes inherited from Digikam::SearchField

- bool **m\_categoryLabelVisible** = true
- AnimatedClearButton \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

## Additional Inherited Members

## Public Types inherited from [Digikam::SearchField](#)

- enum **WidgetRectType** { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Signals inherited from [Digikam::SearchField](#)

- void **signalVisibilityChanged** ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static [SearchField](#) \* **createField** (const QString &fieldName, [SearchFieldGroup](#) \*const parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void **setupLabels** (QGridLayout \*layout, int line)
- void **setValidValueState** (bool valueIsValid)

## 6.1267.1 Member Function Documentation

### 6.1267.1.1 read()

```
void Digikam::SearchFieldRangeDouble::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1267.1.2 reset()

```
void Digikam::SearchFieldRangeDouble::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1267.1.3 setupValueWidgets()

```
void Digikam::SearchFieldRangeDouble::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).



#### 6.1267.1.4 setValueWidgetsVisible()

```
void Digikam::SearchFieldRangeDouble::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1267.1.5 valueWidgetRects()

```
QList< QRect > Digikam::SearchFieldRangeDouble::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1267.1.6 write()

```
void Digikam::SearchFieldRangeDouble::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1268 Digikam::SearchFieldRangeInt Class Reference

Inheritance diagram for Digikam::SearchFieldRangeInt:



### Public Member Functions

- void **enableFractionMagic** (const QString &prefix)
- void **read** ([SearchXmlCachingReader](#) &reader) override

- void [reset](#) () override
- **SearchFieldRangeInt** (QObject \*const parent)
- void **setBetweenText** (const QString &text)
- void **setBoundary** (int min, int max, int step=1)
- void **setInvertStepping** (bool invert)
- void **setNoValueText** (const QString &text)
- void **setNumberPrefixAndSuffix** (const QString &prefix, const QString &suffix)
- void **setSingleSteps** (int smaller, int larger)
- void **setSuggestedInitialValue** (int initialValue)
- void **setSuggestedValues** (const QList< int > &values)
- void [setupValueWidgets](#) (QGridLayout \*layout, int row, int column) override
- void [setValueWidgetsVisible](#) (bool visible) override
- QList< QRect > [valueWidgetRects](#) () const override
- void [write](#) ([SearchXmlWriter](#) &writer) override

### Public Member Functions inherited from [Digikam::SearchField](#)

- bool [isVisible](#) () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** ([SearchField](#) \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void [setVisible](#) (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

### Protected Slots

- void [valueChanged](#) ()

### Protected Slots inherited from [Digikam::SearchField](#)

- void [clearButtonClicked](#) ()

### Protected Attributes

- QLabel \* **m\_betweenLabel** = nullptr
- [CustomStepsIntSpinBox](#) \* **m\_firstBox** = nullptr
- int **m\_max** = 100
- int **m\_min** = 0
- bool **m\_reciprocal** = false
- [CustomStepsIntSpinBox](#) \* **m\_secondBox** = nullptr

### Protected Attributes inherited from [Digikam::SearchField](#)

- bool **m\_categoryLabelVisible** = true
- [AnimatedClearButton](#) \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

## Additional Inherited Members

## Public Types inherited from [Digikam::SearchField](#)

- enum **WidgetRectType** { [LabelAndValueWidgetRects](#) , [ValueWidgetRectsOnly](#) }

## Signals inherited from [Digikam::SearchField](#)

- void **signalVisibilityChanged** ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static [SearchField](#) \* **createField** (const QString &fieldName, [SearchFieldGroup](#) \*const parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void **setupLabels** (QGridLayout \*layout, int line)
- void **setValidValueState** (bool valueIsValid)

## 6.1268.1 Member Function Documentation

### 6.1268.1.1 read()

```
void Digikam::SearchFieldRangeInt::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1268.1.2 reset()

```
void Digikam::SearchFieldRangeInt::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1268.1.3 setupValueWidgets()

```
void Digikam::SearchFieldRangeInt::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1268.1.4 setValueWidgetsVisible()

```
void Digikam::SearchFieldRangeInt::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1268.1.5 valueWidgetRects()

```
QList< QRect > Digikam::SearchFieldRangeInt::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1268.1.6 write()

```
void Digikam::SearchFieldRangeInt::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1269 Digikam::SearchFieldRangeTime Class Reference

Inheritance diagram for Digikam::SearchFieldRangeTime:



### Public Member Functions

- void `read` (`SearchXmlCachingReader &reader`) override
- void `reset` () override

- **SearchFieldRangeTime** (QObject \*const parent)
- void **setBetweenText** (const QString &between)
- void **setBoundary** (const QTime &min, const QTime &max)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** (SearchField \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void **valueChanged** ()

## Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

## Protected Attributes

- QLabel \* **m\_betweenLabel** = nullptr
- QTimeEdit \* **m\_firstTimeEdit** = nullptr
- QTimeEdit \* **m\_secondTimeEdit** = nullptr

## Protected Attributes inherited from Digikam::SearchField

- bool **m\_categoryLabelVisible** = true
- [AnimatedClearButton](#) \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

## Additional Inherited Members

## Public Types inherited from Digikam::SearchField

- enum **WidgetRectType** { **LabelAndValueWidgetRects** , **ValueWidgetRectsOnly** }

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const parent`)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

## 6.1269.1 Member Function Documentation

### 6.1269.1.1 `read()`

```
void Digikam::SearchFieldRangeTime::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1269.1.2 `reset()`

```
void Digikam::SearchFieldRangeTime::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1269.1.3 `setupValueWidgets()`

```
void Digikam::SearchFieldRangeTime::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1269.1.4 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldRangeTime::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).



### 6.1269.1.5 valueWidgetRects()

```
QList< QRect > Digikam::SearchFieldRangeTime::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

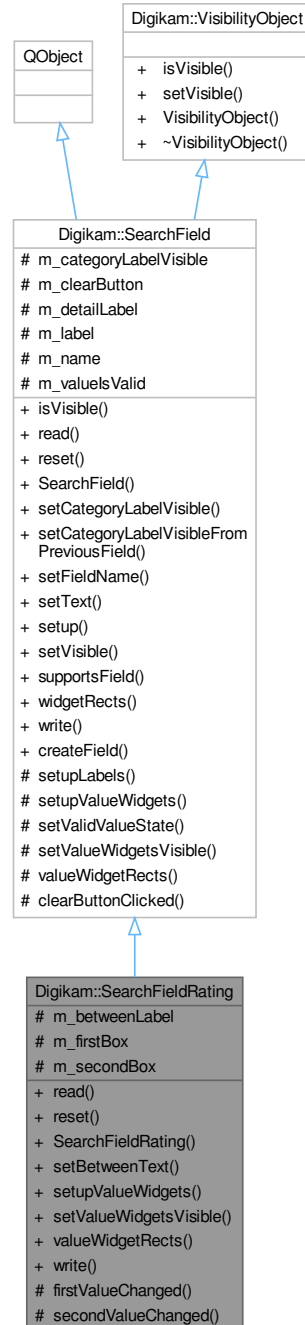
### 6.1269.1.6 write()

```
void Digikam::SearchFieldRangeTime::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1270 Digikam::SearchFieldRating Class Reference

Inheritance diagram for Digikam::SearchFieldRating:



### Public Member Functions

- void `read` (`SearchXmlCachingReader &reader`) override
- void `reset` () override

- **SearchFieldRating** (QObject \*const parent)
- void **setBetweenText** (const QString &text)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

### Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** (SearchField \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgetRects** (WidgetRectType=ValueWidgetRectsOnly) const

### Protected Slots

- void **firstValueChanged** ()
- void **secondValueChanged** ()

### Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

### Protected Attributes

- QLabel \* **m\_betweenLabel** = nullptr
- RatingComboBox \* **m\_firstBox** = nullptr
- RatingComboBox \* **m\_secondBox** = nullptr

### Protected Attributes inherited from Digikam::SearchField

- bool **m\_categoryLabelVisible** = true
- AnimatedClearButton \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuesValid** = false

### Additional Inherited Members

### Public Types inherited from Digikam::SearchField

- enum **WidgetRectType** { LabelAndValueWidgetRects , ValueWidgetRectsOnly }

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const parent`)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

## 6.1270.1 Member Function Documentation

### 6.1270.1.1 `read()`

```
void Digikam::SearchFieldRating::read (
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1270.1.2 `reset()`

```
void Digikam::SearchFieldRating::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1270.1.3 `setupValueWidgets()`

```
void Digikam::SearchFieldRating::setupValueWidgets (
    QGridLayout * layout,
    int row,
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1270.1.4 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldRating::setValueWidgetsVisible (
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1270.1.5 valueWidgetRects()

```
QList< QRect > Digikam::SearchFieldRating::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

### 6.1270.1.6 write()

```
void Digikam::SearchFieldRating::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1271 Digikam::SearchFieldText Class Reference

Inheritance diagram for Digikam::SearchFieldText:



### Public Member Functions

- void [read](#) ([SearchXmlCachingReader](#) &reader) override
- void [reset](#) () override

- **SearchFieldText** (QObject \*const parent)
- void **setupValueWidgets** (QGridLayout \*layout, int row, int column) override
- void **setValueWidgetsVisible** (bool visible) override
- QList< QRect > **valueWidgetRects** () const override
- void **write** (SearchXmlWriter &writer) override

## Public Member Functions inherited from Digikam::SearchField

- bool **isVisible** () override
- **SearchField** (QObject \*const parent)
- void **setCategoryLabelVisible** (bool visible)
- void **setCategoryLabelVisibleFromPreviousField** (SearchField \*const previousField)
- void **setFieldName** (const QString &fieldName)
- virtual void **setText** (const QString &label, const QString &detailLabel)
- void **setup** (QGridLayout \*const layout, int row=-1)
- void **setVisible** (bool visible) override
- virtual bool **supportsField** (const QString &fieldName)
- QList< QRect > **widgerRects** (WidgetRectType=ValueWidgetRectsOnly) const

## Protected Slots

- void **valueChanged** (const QString &text)

## Protected Slots inherited from Digikam::SearchField

- void **clearButtonClicked** ()

## Protected Attributes

- QLineEdit \* **m\_edit** = nullptr

## Protected Attributes inherited from Digikam::SearchField

- bool **m\_categoryLabelVisible** = true
- AnimatedClearButton \* **m\_clearButton** = nullptr
- QLabel \* **m\_detailLabel** = nullptr
- QLabel \* **m\_label** = nullptr
- QString **m\_name**
- bool **m\_valuelsValid** = false

## Additional Inherited Members

## Public Types inherited from Digikam::SearchField

- enum **WidgetRectType** { LabelAndValueWidgetRects , ValueWidgetRectsOnly }

## Signals inherited from [Digikam::SearchField](#)

- void `signalVisibilityChanged` ()

## Static Public Member Functions inherited from [Digikam::SearchField](#)

- static `SearchField * createField` (const QString &fieldName, `SearchFieldGroup *const` parent)

## Protected Member Functions inherited from [Digikam::SearchField](#)

- virtual void `setupLabels` (QGridLayout \*layout, int line)
- void `setValidValueState` (bool valueIsValid)

### 6.1271.1 Member Function Documentation

#### 6.1271.1.1 `read()`

```
void Digikam::SearchFieldText::read (  
    SearchXmlCachingReader & reader ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1271.1.2 `reset()`

```
void Digikam::SearchFieldText::reset ( ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1271.1.3 `setupValueWidgets()`

```
void Digikam::SearchFieldText::setupValueWidgets (  
    QGridLayout * layout,  
    int row,  
    int column ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

#### 6.1271.1.4 `setValueWidgetsVisible()`

```
void Digikam::SearchFieldText::setValueWidgetsVisible (  
    bool visible ) [override], [virtual]
```

Implements [Digikam::SearchField](#).



### 6.1271.1.5 valueWidgetRects()

```
QList< QRect > Digikam::SearchFieldText::valueWidgetRects ( ) const [override], [virtual]
```

Implements [Digikam::SearchField](#).

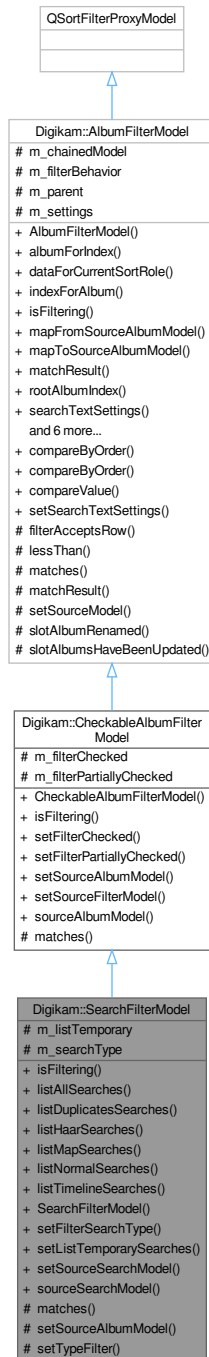
### 6.1271.1.6 write()

```
void Digikam::SearchFieldText::write (
    SearchXmlWriter & writer ) [override], [virtual]
```

Implements [Digikam::SearchField](#).

## 6.1272 Digikam::SearchFilterModel Class Reference

Inheritance diagram for Digikam::SearchFilterModel:



### Public Member Functions

- bool `isFiltering` () const override
- void `listAllSearches` ()

- void **listDuplicatesSearches** ()
- void **listHaarSearches** ()
- void **listMapSearches** ()
- void **listNormalSearches** ()
- void **listTimelineSearches** ()
- **SearchFilterModel** (QObject \*const parent=nullptr)
- void **setFilterSearchType** (DatabaseSearch::Type)
- void **setListTemporarySearches** (bool list)
- void **setSourceSearchModel** ([SearchModel](#) \*const source)
- [SearchModel](#) \* **sourceSearchModel** () const

### Public Member Functions inherited from [Digikam::CheckableAlbumFilterModel](#)

- **CheckableAlbumFilterModel** (QObject \*const parent=nullptr)
- void **setFilterChecked** (bool filter)
- void **setFilterPartiallyChecked** (bool filter)
- void **setSourceAlbumModel** ([AbstractCheckableAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([CheckableAlbumFilterModel](#) \*const source)
- [AbstractCheckableAlbumModel](#) \* **sourceAlbumModel** () const

### Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- **AlbumFilterModel** (QObject \*const parent=nullptr)
- [Album](#) \* **albumForIndex** (const QModelIndex &index) const  
*Convenience methods.*
- QVariant **dataForCurrentSortRole** ([Album](#) \*album) const
- QModelIndex **indexForAlbum** ([Album](#) \*album) const
- QModelIndex **mapFromSourceAlbumModel** (const QModelIndex &index) const
- QModelIndex **mapToSourceAlbumModel** (const QModelIndex &index) const
- [MatchResult](#) **matchResult** (const QModelIndex &index) const
- QModelIndex **rootAlbumIndex** () const
- [SearchTextSettings](#) **searchTextSettings** () const
- void **setFilterBehavior** ([FilterBehavior](#) behavior)
- void **setSourceAlbumModel** ([AbstractAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([AlbumFilterModel](#) \*const source)
- [AbstractAlbumModel](#) \* **sourceAlbumModel** () const
- [AlbumFilterModel](#) \* **sourceFilterModel** () const
- void **updateFilter** ()

### Protected Member Functions

- bool **matches** ([Album](#) \*album) const override
- void **setSourceAlbumModel** ([AbstractAlbumModel](#) \*const source)
- void **setTypeFilter** (int type)

### Protected Member Functions inherited from [Digikam::AlbumFilterModel](#)

- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- [MatchResult](#) **matchResult** ([Album](#) \*album) const
- void **setSourceModel** (QAbstractItemModel \*const model) override

### Protected Attributes

- bool `m_listTemporary` = false
- int `m_searchType` = -1

### Protected Attributes inherited from [Digikam::CheckableAlbumFilterModel](#)

- bool `m_filterChecked` = false
- bool `m_filterPartiallyChecked` = false

### Protected Attributes inherited from [Digikam::AlbumFilterModel](#)

- `QPointer< AlbumFilterModel >` `m_chainedModel` = nullptr
- `FilterBehavior` `m_filterBehavior` = `FullFiltering`
- `QObject *` `m_parent` = nullptr
- `SearchTextSettings` `m_settings`

### Additional Inherited Members

### Public Types inherited from [Digikam::AlbumFilterModel](#)

- enum `FilterBehavior` { `SimpleFiltering` , `FullFiltering` , `StrictFiltering` }
- enum `MatchResult` {  
    `NoMatch` = 0 , `DirectMatch` , `ParentMatch` , `ChildMatch` ,  
    `SpecialMatch` }

### Public Slots inherited from [Digikam::AlbumFilterModel](#)

- void `setSearchTextSettings` (const `SearchTextSettings` &settings)

### Signals inherited from [Digikam::AlbumFilterModel](#)

- void `hasSearchResult` (bool hasResult)
- void `searchTextSettingsAboutToChange` (bool searched, bool willSearch)
- void `searchTextSettingsChanged` (bool wasSearching, bool searched)
- void `signalFilterChanged` ()

### Static Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- template<typename T >  
    static int `compareByOrder` (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int `compareByOrder` (int compareResult, Qt::SortOrder sortOrder)
- template<typename T >  
    static int `compareValue` (const T &a, const T &b)

### Protected Slots inherited from [Digikam::AlbumFilterModel](#)

- void `slotAlbumRenamed` (`Album *`album)
- void `slotAlbumsHaveBeenUpdated` (int type)

## 6.1272.1 Detailed Description

[Filter](#) model for searches that can filter by search type

## 6.1272.2 Member Function Documentation

### 6.1272.2.1 isFiltering()

```
bool Digikam::SearchFilterModel::isFiltering ( ) const [override], [virtual]
```

Returns if the currently applied filters will result in any filtering.

#### Returns

`true` if the current selected filter could result in any filtering without checking if this really happens.

Reimplemented from [Digikam::CheckableAlbumFilterModel](#).

### 6.1272.2.2 matches()

```
bool Digikam::SearchFilterModel::matches (
    Album * album ) const [override], [protected], [virtual]
```

This method provides the basic match checking algorithm. Return true if this single album matches the current criteria. This method can be overridden to provide custom filtering.

#### Parameters

<i>album</i>	the album to tell if it matches the filter criteria or not.
--------------	---

Reimplemented from [Digikam::CheckableAlbumFilterModel](#).

### 6.1272.2.3 setFilterSearchType()

```
void Digikam::SearchFilterModel::setFilterSearchType (
    DatabaseSearch::Type type )
```

Set the DatabaseSearch::Type.

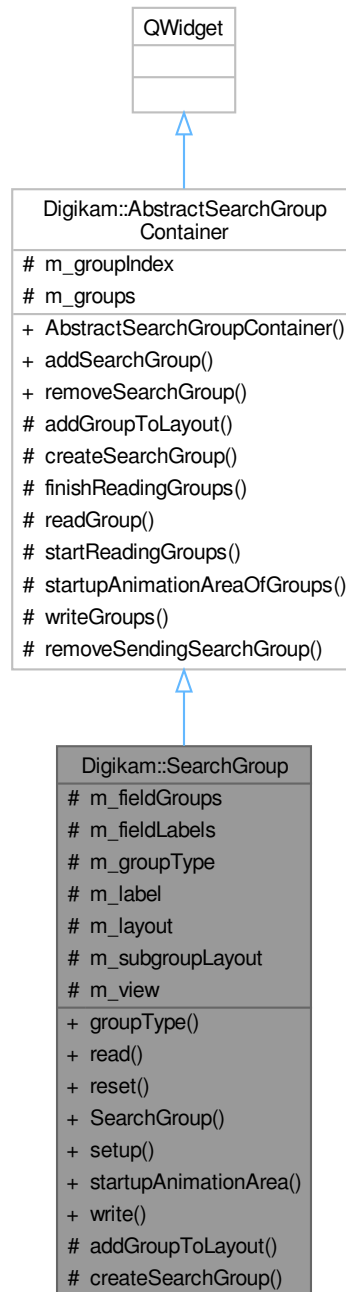
### 6.1272.2.4 setListTemporarySearches()

```
void Digikam::SearchFilterModel::setListTemporarySearches (
    bool list )
```

Sets if temporary search albums shall be listed

## 6.1273 Digikam::SearchGroup Class Reference

Inheritance diagram for Digikam::SearchGroup:



### Public Types

- enum **Type** { **FirstGroup** , **ChainGroup** }

## Signals

- void **removeRequested** ()

## Public Member Functions

- Type **groupType** () const
- void **read** ([SearchXmlCachingReader](#) &reader)
- void **reset** ()
- **SearchGroup** ([SearchView](#) \*const parent)
- void **setup** (Type type=FirstGroup)
- [QList](#)< [QRect](#) > **startupAnimationArea** () const
- void **write** ([SearchXmlWriter](#) &writer)

## Public Member Functions inherited from [Digikam::AbstractSearchGroupContainer](#)

- [AbstractSearchGroupContainer](#) ([QWidget](#) \*const parent=nullptr)

## Protected Member Functions

- void **addGroupToLayout** ([SearchGroup](#) \*group) override  
*Re-implement: Adds a newly created group to the layout structures.*
- [SearchGroup](#) \* **createSearchGroup** () override  
*Re-implement: create and setup a search group.*

## Protected Member Functions inherited from [Digikam::AbstractSearchGroupContainer](#)

- void **finishReadingGroups** ()  
*Call when the XML part is finished.*
- void **readGroup** ([SearchXmlCachingReader](#) &reader)  
*Call when a group element is the current element.*
- void **startReadingGroups** ([SearchXmlCachingReader](#) &reader)  
*Call before reading the XML part that could contain group elements.*
- [QList](#)< [QRect](#) > **startupAnimationAreaOfGroups** () const  
*Collects the data from the same method of all contained groups (position relative to this widget)*
- void **writeGroups** ([SearchXmlWriter](#) &writer) const  
*Write contained groups to writer.*

## Protected Attributes

- [QList](#)< [SearchFieldGroup](#) \* > **m\_fieldGroups**
- [QList](#)< [SearchFieldGroupLabel](#) \* > **m\_fieldLabels**
- Type **m\_groupType** = FirstGroup
- [SearchGroupLabel](#) \* **m\_label** = nullptr
- [QVBoxLayout](#) \* **m\_layout** = nullptr
- [QVBoxLayout](#) \* **m\_subgroupLayout** = nullptr
- [SearchView](#) \* **m\_view** = nullptr

## Protected Attributes inherited from [Digikam::AbstractSearchGroupContainer](#)

- int `m_groupIndex` = 0
- `QList< SearchGroup * >` `m_groups`

## Additional Inherited Members

## Public Slots inherited from [Digikam::AbstractSearchGroupContainer](#)

- `SearchGroup * addSearchGroup ()`
- void `removeSearchGroup (SearchGroup *group)`

## Protected Slots inherited from [Digikam::AbstractSearchGroupContainer](#)

- void `removeSendingSearchGroup ()`

## 6.1273.1 Member Function Documentation

### 6.1273.1.1 addGroupToLayout()

```
void Digikam::SearchGroup::addGroupToLayout (  
    SearchGroup * group ) [override], [protected], [virtual]
```

Implements [Digikam::AbstractSearchGroupContainer](#).

### 6.1273.1.2 createSearchGroup()

```
SearchGroup * Digikam::SearchGroup::createSearchGroup ( ) [override], [protected], [virtual]
```

Implements [Digikam::AbstractSearchGroupContainer](#).



## 6.1274 Digikam::SearchGroupLabel Class Reference

Inheritance diagram for Digikam::SearchGroupLabel:



### Signals

- void **removeClicked** ()

### Public Member Functions

- SearchXml::Operator **defaultFieldOperator** () const
- SearchXml::Operator **groupOperator** () const
- **SearchGroupLabel** ([SearchViewThemedPartsCache](#) \*const cache, SearchGroup::Type type, QWidget \*const parent=nullptr)
- void **setDefaultFieldOperator** (SearchXml::Operator op)
- void **setGroupOperator** (SearchXml::Operator op)

### Protected Slots

- void **boxesToggled** ()
- void **toggleGroupOperator** ()
- void **toggleShowOptions** ()

### Protected Member Functions

- void **adjustOperatorOptions** ()
- void **paintEvent** (QPaintEvent \*) override
- void **setExtended** (bool extended)
- void **updateGroupLabel** ()

## 6.1275 Digikam::SearchInfo Class Reference

### Public Types

- typedef QList< [SearchInfo](#) > **List**

### Public Member Functions

- bool **isNull** () const
- bool **operator<** (const [SearchInfo](#) &info) const

### Public Attributes

- int **id** = 0
- QString **name**
- QString **query**
- DatabaseSearch::Type **type** = DatabaseSearch::UndefinedType

### 6.1275.1 Detailed Description

A container class for transporting search information from the database to [AlbumManager](#)

### 6.1275.2 Member Function Documentation

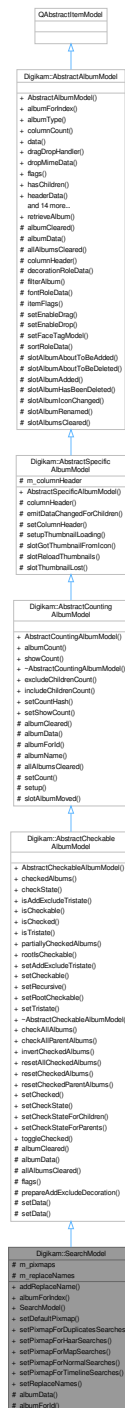
#### 6.1275.2.1 operator<()

```
bool Digikam::SearchInfo::operator< (  
    const SearchInfo & info ) const [inline]
```

needed for sorting

## 6.1276 Digikam::SearchModel Class Reference

Inheritance diagram for Digikam::SearchModel:



### Public Member Functions

- void **addReplaceName** (const QString &technicalName, const QString &userVisibleName)
- **SAlbum** \* **albumForIndex** (const QModelIndex &index) const

- [searchModel](#) (QObject \*const parent=nullptr)
- void **setDefaultPixmap** (const QPixmap &pix)
- void **setPixmapForDuplicatesSearches** (const QPixmap &pix)
- void **setPixmapForHaarSearches** (const QPixmap &pix)
- void **setPixmapForMapSearches** (const QPixmap &pix)
- void [setPixmapForNormalSearches](#) (const QPixmap &pix)
- void **setPixmapForTimelineSearches** (const QPixmap &pix)
- void [setReplaceNames](#) (const QHash< QString, QString > &replaceNames)

### Public Member Functions inherited from [Digikam::AbstractCheckableAlbumModel](#)

- [AbstractCheckableAlbumModel](#) ([Album::Type](#) albumType, Album \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), QObject \*const parent=nullptr)
- QList< [Album](#) \* > **checkedAlbums** () const  
*Returns a list of album with check state Checked.*
- Qt::CheckState **checkState** ([Album](#) \*album) const  
*Returns the check state of the album.*
- bool **isAddExcludeTristate** () const
- bool **isCheckable** () const
- bool **isChecked** ([Album](#) \*album) const  
*Returns if the given album has the check state Checked.*
- bool **isTristate** () const
- QList< [Album](#) \* > **partiallyCheckedAlbums** () const  
*Returns a list of album with partially check state Checked.*
- bool **rootIsCheckable** () const
- void [setAddExcludeTristate](#) (bool b)
- void **setCheckable** (bool isCheckable)  
*Triggers if the albums in this model are checkable.*
- void [setRecursive](#) (bool recursive)
- void [setRootCheckable](#) (bool rootIsCheckable)
- void [setTristate](#) (bool isTristate)

### Public Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- **AbstractCountingAlbumModel** ([Album::Type](#) albumType, Album \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), QObject \*const parent=nullptr)  
*Supports displaying a count alongside the album name in DisplayRole.*
- virtual int [albumCount](#) ([Album](#) \*album) const
- bool **showCount** () const

### Public Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- **AbstractSpecificAlbumModel** ([Album::Type](#) albumType, Album \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), QObject \*const parent=nullptr)  
*Abstract base class, do not instantiate.*

## Public Member Functions inherited from Digikam::AbstractAlbumModel

- [AbstractAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [Album](#) \* [albumForIndex](#) (const [QModelIndex](#) &index) const
- [Album::Type](#) [albumType](#) () const
- int **columnCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [AlbumModelDragDropHandler](#) \* [dragDropHandler](#) () const
- bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role=[Qt::DisplayRole](#)) const override
- [QModelIndex](#) **index** (int row, int column, const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QModelIndex](#) [indexForAlbum](#) ([Album](#) \*album) const
- bool [isFaceTagModel](#) () const
- [QMimeData](#) \* **mimeData** (const [QModelIndexList](#) &indexes) const override
- [QStringList](#) **mimeTypes** () const override
- [QModelIndex](#) **parent** (const [QModelIndex](#) &index) const override
- [Album](#) \* **rootAlbum** () const
- [RootAlbumBehavior](#) [rootAlbumBehavior](#) () const
- [QModelIndex](#) [rootAlbumIndex](#) () const
- int **rowCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- void [setDragDropHandler](#) ([AlbumModelDragDropHandler](#) \*handler)
- void [setDropIndex](#) (const [QModelIndex](#) &index)
- [Qt::DropActions](#) **supportedDropActions** () const override

## Protected Member Functions

- [QVariant](#) [albumData](#) ([Album](#) \*a, int role) const override  
*For subclassing convenience: A part of the implementation of data()*
- [Album](#) \* [albumForId](#) (int id) const override  
*need to implement in subclass*

## Protected Member Functions inherited from Digikam::AbstractCheckableAlbumModel

- void [albumCleared](#) ([Album](#) \*album) override  
*Notification when an entry is removed.*
- void [allAlbumsCleared](#) () override  
*Notification when all entries are removed.*
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- void [prepareAddExcludeDecoration](#) ([Album](#) \*a, [QPixmap](#) &icon) const
- bool **setData** (const [QModelIndex](#) &index, const [QVariant](#) &value, int role, bool recursive)
- bool [setData](#) (const [QModelIndex](#) &index, const [QVariant](#) &value, int role=[Qt::EditRole](#)) override

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [albumCleared](#) ([Album](#) \*album) override  
*Notification when an entry is removed.*
- virtual QString [albumName](#) ([Album](#) \*a) const  
*Can reimplement in subclass.*
- void [allAlbumsCleared](#) () override  
*Notification when all entries are removed.*
- void **setCount** ([Album](#) \*album, int count)  
*If you do not use setCountHash, excludeChildrenCount and includeChildrenCount, you can set a count here.*
- void [setup](#) ()

### Protected Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString [columnHeader](#) () const override  
*For subclassing convenience: A part of the implementation of headerData()*
- void **emitDataChangedForChildren** ([Album](#) \*album)
- void **setColumnHeader** (const QString &header)
- void **setupThumbnailLoading** ()  
*You need to call this from your constructor if you intend to load the thumbnail facilities of this class.*

### Protected Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- virtual QVariant [decorationRoleData](#) ([Album](#) \*a) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual bool [filterAlbum](#) ([Album](#) \*album) const
- virtual QVariant [fontRoleData](#) ([Album](#) \*a) const  
*For subclassing convenience: A part of the implementation of data()*
- virtual Qt::ItemFlags **itemFlags** ([Album](#) \*album) const  
*For subclassing convenience: A part of the implementation of itemFlags()*
- void [setEnableDrag](#) (bool enable)
- void **setEnableDrop** (bool enable)
- void **setFaceTagModel** (bool enable)
- virtual QVariant [sortRoleData](#) ([Album](#) \*a) const  
*For subclassing convenience: A part of the implementation of data()*

### Protected Attributes

- QHash< int, QPixmap > **m\_pixmap**s
- QHash< QString, QString > **m\_replaceNames**

### Protected Attributes inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString **m\_columnHeader**

## Additional Inherited Members

### Public Types inherited from [Digikam::AbstractAlbumModel](#)

- enum [AlbumDataRole](#) {  
[AlbumTitleRole](#) = Qt::UserRole , [AlbumTypeRole](#) = Qt::UserRole + 1 , [AlbumPointerRole](#) = Qt::UserRole + 2  
, [AlbumIdRole](#) = Qt::UserRole + 3 ,  
[AlbumGlobalIdRole](#) = Qt::UserRole + 4 , [AlbumSortRole](#) = Qt::UserRole + 5 }
- enum [RootAlbumBehavior](#) { [IncludeRootAlbum](#) , [IgnoreRootAlbum](#) }

### Public Slots inherited from [Digikam::AbstractCheckableAlbumModel](#)

- void **checkAllAlbums** (const QModelIndex &parent=QModelIndex())  
*Checks all albums beneath the given parent.*
- void **checkAllParentAlbums** (const QModelIndex &child)  
*Checks all parent albums starting at the child, including it.*
- void **invertCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Inverts the checked state of all albums under the given parent.*
- void **resetAllCheckedAlbums** ()  
*Resets the checked state of all albums to Qt::Unchecked.*
- void **resetCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Resets the checked state of all albums under the given parent.*
- void **resetCheckedParentAlbums** (const QModelIndex &child)  
*Resets the checked state of all parents of the child including it.*
- void **setChecked** ([Album](#) \*album, bool isChecked)  
*Sets the check state of album to Checked or Unchecked.*
- void **setCheckState** ([Album](#) \*album, Qt::CheckState state)  
*Sets the check state of the album.*
- void **setCheckStateForChildren** ([Album](#) \*album, Qt::CheckState state)  
*Sets the checked state recursively for all children of but not for the given album.*
- void **setCheckStateForParents** ([Album](#) \*album, Qt::CheckState state)  
*Sets the checked state recursively for all parents of but not for the given album.*
- void **toggleChecked** ([Album](#) \*album)  
*Toggles the check state of album between Checked or Unchecked.*

### Public Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [excludeChildrenCount](#) (const QModelIndex &index)
- void [includeChildrenCount](#) (const QModelIndex &index)
- void [setCountHash](#) (const QHash< int, int > &idCountHash)
- void **setShowCount** (bool show)  
*Call to enable or disable showing the count. Default is false.*

### Signals inherited from [Digikam::AbstractCheckableAlbumModel](#)

- void [checkStateChanged](#) ([Album](#) \*album, Qt::CheckState checkState)

### Signals inherited from [Digikam::AbstractCountingAlbumModel](#)

- void **signalUpdateAlbumCount** ([Album](#) \*album)

## Signals inherited from [Digikam::AbstractAlbumModel](#)

- void [rootAlbumAvailable](#) ()

## Static Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- static [Album](#) \* [retrieveAlbum](#) (const [QModelIndex](#) &index)

## Protected Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [slotAlbumMoved](#) ([Album](#) \*album)

## Protected Slots inherited from [Digikam::AbstractSpecificAlbumModel](#)

- void [slotGotThumbnailFromIcon](#) ([Album](#) \*album, const [QPixmap](#) &thumbnail)
- void [slotReloadThumbnails](#) ()
- void [slotThumbnailLost](#) ([Album](#) \*album)

## Protected Slots inherited from [Digikam::AbstractAlbumModel](#)

- void [slotAlbumAboutToBeAdded](#) ([Album](#) \*album, [Album](#) \*parent, [Album](#) \*prev)
- void [slotAlbumAboutToBeDeleted](#) ([Album](#) \*album)
- void [slotAlbumAdded](#) ([Album](#) \*)
- void [slotAlbumHasBeenDeleted](#) ([Album](#) \*album)
- void [slotAlbumIconChanged](#) ([Album](#) \*album)
- void [slotAlbumRenamed](#) ([Album](#) \*album)
- void [slotAlbumsCleared](#) ()

## 6.1276.1 Constructor & Destructor Documentation

### 6.1276.1.1 [searchModel\(\)](#)

```
Digikam::searchModel::searchModel (
    QObject *const parent = nullptr ) [explicit]
```

Create a model containing searches

## 6.1276.2 Member Function Documentation

### 6.1276.2.1 [albumData\(\)](#)

```
QVariant Digikam::searchModel::albumData (
    Album * a,
    int role ) const [override], [protected], [virtual]
```

#### Note

these can be reimplemented in a subclass

Reimplemented from [Digikam::AbstractCheckableAlbumModel](#).



### 6.1276.2.2 albumForId()

```
Album * Digikam::SearchModel::albumForId (
    int id ) const [override], [protected], [virtual]
```

Implements [Digikam::AbstractCountingAlbumModel](#).

### 6.1276.2.3 setPixmapForNormalSearches()

```
void Digikam::SearchModel::setPixmapForNormalSearches (
    const QPixmap & pix )
```

Set pixmaps for the DecorationRole

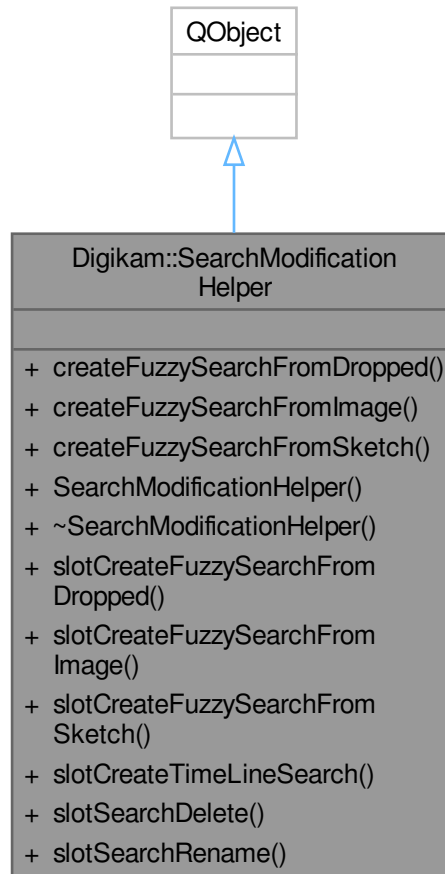
### 6.1276.2.4 setReplaceNames()

```
void Digikam::SearchModel::setReplaceNames (
    const QHash< QString, QString > & replaceNames )
```

Set a hash of internal names (key) that shall be replaced by a user-visible string (value). This affects Qt::DisplayRole and AlbumTitleRole.

## 6.1277 Digikam::SearchModificationHelper Class Reference

Inheritance diagram for Digikam::SearchModificationHelper:



### Public Slots

- void [slotCreateFuzzySearchFromDropped](#) (const QString &name, const QString &filePath, float threshold, float maxThreshold, const QList< int > &targetAlbums, bool overwriteIfExisting)
- void [slotCreateFuzzySearchFromImage](#) (const QString &name, const [ItemInfo](#) &image, float threshold, float maxThreshold, const QList< int > &targetAlbums, bool overwriteIfExisting=false)
- void [slotCreateFuzzySearchFromSketch](#) (const QString &name, [SketchWidget](#) \*sketchWidget, unsigned int numberOfResults, const QList< int > &targetAlbums, bool overwriteIfExisting=false)
- [SAlbum](#) \* [slotCreateTimeLineSearch](#) (const QString &desiredName, const [DateRangeList](#) &dateRanges, bool overwriteIfExisting=false)
- void [slotSearchDelete](#) ([SAlbum](#) \*searchAlbum)
- void [slotSearchRename](#) ([SAlbum](#) \*searchAlbum)

## Public Member Functions

- [SAlbum](#) \* [createFuzzySearchFromDropped](#) (const QString &name, const QString &filePath, float threshold, float maxThreshold, const QList< int > &targetAlbums, bool overwriteIfExists=false)
- [SAlbum](#) \* [createFuzzySearchFromImage](#) (const QString &name, const [ItemInfo](#) &image, float threshold, float maxThreshold, const QList< int > &targetAlbums, bool overwriteIfExists=false)
- [SAlbum](#) \* [createFuzzySearchFromSketch](#) (const QString &name, [SketchWidget](#) \*sketchWidget, unsigned int numberOfResults, const QList< int > &targetAlbums, bool overwriteIfExists=false)
- [SearchModificationHelper](#) (QObject \*const parent, QWidget \*const dialogParent)
- [~SearchModificationHelper](#) () override

### 6.1277.1 Detailed Description

Utility class providing methods to modify search albums ([SAlbum](#)) in a way useful to implement views.

#### Author

jwienke

### 6.1277.2 Constructor & Destructor Documentation

#### 6.1277.2.1 SearchModificationHelper()

```
Digikam::SearchModificationHelper::SearchModificationHelper (
    QObject *const parent,
    QWidget *const dialogParent )
```

Constructor.

#### Parameters

<i>parent</i>	the parent for qt parent child mechanism
<i>dialogParent</i>	parent widget for dialogs displayed by this object

#### 6.1277.2.2 ~SearchModificationHelper()

```
Digikam::SearchModificationHelper::~~SearchModificationHelper ( ) [override]
```

Destructor.

### 6.1277.3 Member Function Documentation

#### 6.1277.3.1 createFuzzySearchFromDropped()

```
SAlbum * Digikam::SearchModificationHelper::createFuzzySearchFromDropped (
    const QString & name,
    const QString & filePath,
    float threshold,
    float maxThreshold,
    const QList< int > & targetAlbums,
    bool overwriteIfExists = false )
```

See also

[slotCreateFuzzySearchFromDropped\(\)](#)

Returns

the newly created album

### 6.1277.3.2 createFuzzySearchFromImage()

```
SAlbum * Digikam::SearchModificationHelper::createFuzzySearchFromImage (
    const QString & name,
    const ItemInfo & image,
    float threshold,
    float maxThreshold,
    const QList< int > & targetAlbums,
    bool overwriteIfExists = false )
```

See also

[slotCreateFuzzySearchFromImage\(\)](#)

Returns

the newly created album

### 6.1277.3.3 createFuzzySearchFromSketch()

```
SAlbum * Digikam::SearchModificationHelper::createFuzzySearchFromSketch (
    const QString & name,
    SketchWidget * sketchWidget,
    unsigned int numberOfResults,
    const QList< int > & targetAlbums,
    bool overwriteIfExists = false )
```

See also

[slotCreateFuzzySearchFromSketch\(\)](#)

Returns

the newly created album

### 6.1277.3.4 slotCreateFuzzySearchFromDropped

```
void Digikam::SearchModificationHelper::slotCreateFuzzySearchFromDropped (
    const QString & name,
    const QString & filePath,
    float threshold,
    float maxThreshold,
    const QList< int > & targetAlbums,
    bool overwriteIfExists ) [slot]
```

Creates a new fuzzy search for finding similar photos based on the file path of a photo and selects it in the album manager after creation.

## Parameters

<i>name</i>	of the new search
<i>filePath</i>	path of the image to base this search on
<i>threshold</i>	minimum threshold for image search
<i>maxThreshold</i>	maximum threshold for image search
<i>targetAlbums</i>	The image must be in one of these albums
<i>overwriteIfExists</i>	if true, an existing search with the desired name will be overwritten without prompting the user for a new name

## 6.1277.3.5 slotCreateFuzzySearchFromImage

```
void Digikam::SearchModificationHelper::slotCreateFuzzySearchFromImage (
    const QString & name,
    const ItemInfo & image,
    float threshold,
    float maxThreshold,
    const QList< int > & targetAlbums,
    bool overwriteIfExists = false ) [slot]
```

Creates a new fuzzy search for finding similar photos based on one photo and selects it in the album manager after creation.

## Parameters

<i>name</i>	of the new search
<i>image</i>	the image to base this search on
<i>threshold</i>	the threshold for image search, $0 \leq \text{threshold} \leq 1$
<i>maxThreshold</i>	the maximum threshold of similarity.
<i>targetAlbums</i>	The image must be in one of these albums
<i>overwriteIfExists</i>	if true, an existing search with the desired name will be overwritten without prompting the user for a new name

## 6.1277.3.6 slotCreateFuzzySearchFromSketch

```
void Digikam::SearchModificationHelper::slotCreateFuzzySearchFromSketch (
    const QString & name,
    SketchWidget * sketchWidget,
    unsigned int numberOfResults,
    const QList< int > & targetAlbums,
    bool overwriteIfExists = false ) [slot]
```

Creates a new fuzzy search based on a sketch created by the user and selects it in the [AlbumManager](#) after creation.

## Parameters

<i>name</i>	the name of the new sketch search
<i>sketchWidget</i>	the widget containing the sketch of the user
<i>numberOfResults</i>	max number of results to display
<i>targetAlbums</i>	The image must be in one of these albums
<i>overwriteIfExists</i>	if true, an existing search with the desired name will be overwritten without prompting the

### 6.1277.3.7 slotCreateTimeLineSearch

```
SAlbum * Digikam::SearchModificationHelper::slotCreateTimeLineSearch (
    const QString & desiredName,
    const DateRangeList & dateRanges,
    bool overwriteIfExists = false ) [slot]
```

Creates a new timeline search.

#### Parameters

<i>desiredName</i>	desired name for the search. If this name already exists and <code>overwriteIfExists</code> is false, then the user will be prompted for a new name
<i>dateRanges</i>	date ranges to contain in this timeline search. If this is empty, no search will be created.
<i>overwriteIfExists</i>	if true, an existing search with the desired name will be overwritten without prompting the user for a new name

### 6.1277.3.8 slotSearchDelete

```
void Digikam::SearchModificationHelper::slotSearchDelete (
    SAlbum * searchAlbum ) [slot]
```

Deletes the given search after prompting the user.

#### Parameters

<i>searchAlbum</i>	search to delete
--------------------	------------------

### 6.1277.3.9 slotSearchRename

```
void Digikam::SearchModificationHelper::slotSearchRename (
    SAlbum * searchAlbum ) [slot]
```

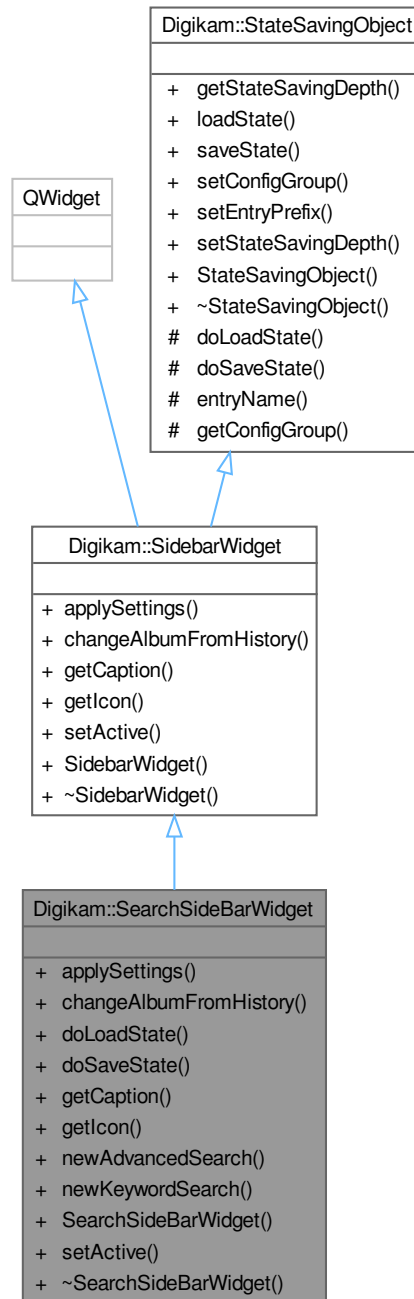
Renames the given search via a dialog.

#### Parameters

<i>searchAlbum</i>	search to rename
--------------------	------------------

## 6.1278 Digikam::SearchSideBarWidget Class Reference

Inheritance diagram for Digikam::SearchSideBarWidget:



### Public Member Functions

- void `applySettings()` override
- void `changeAlbumFromHistory` (const QList< Album \* > &album) override

- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- void [newAdvancedSearch](#) ()
- void [newKeywordSearch](#) ()
- [SearchSideBarWidget](#) (QWidget \*const parent, [searchModel](#) \*const searchModel, [SearchModificationHelper](#) \*const searchModificationHelper)
- void [setActive](#) (bool active) override

### Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

### Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.1278.1 Member Function Documentation

### 6.1278.1.1 [applySettings\(\)](#)

```
void Digikam::SearchSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).



### 6.1278.1.2 changeAlbumFromHistory()

```
void Digikam::SearchSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.1278.1.3 doLoadState()

```
void Digikam::SearchSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1278.1.4 doSaveState()

```
void Digikam::SearchSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1278.1.5 getCaption()

```
const QString Digikam::SearchSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.1278.1.6 getIcon()

```
const QIcon Digikam::SearchSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.1278.1.7 setActive()

```
void Digikam::SearchSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

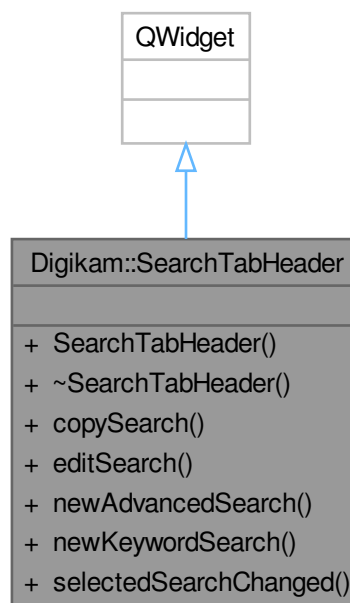
## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.1279 Digikam::SearchTabHeader Class Reference

Inheritance diagram for Digikam::SearchTabHeader:



### Public Slots

- void `copySearch` ([SAlbum](#) \*album)
- void `editSearch` ([SAlbum](#) \*album)
- void `newAdvancedSearch` ()
- void `newKeywordSearch` ()
- void `selectedSearchChanged` ([Album](#) \*album)

### Signals

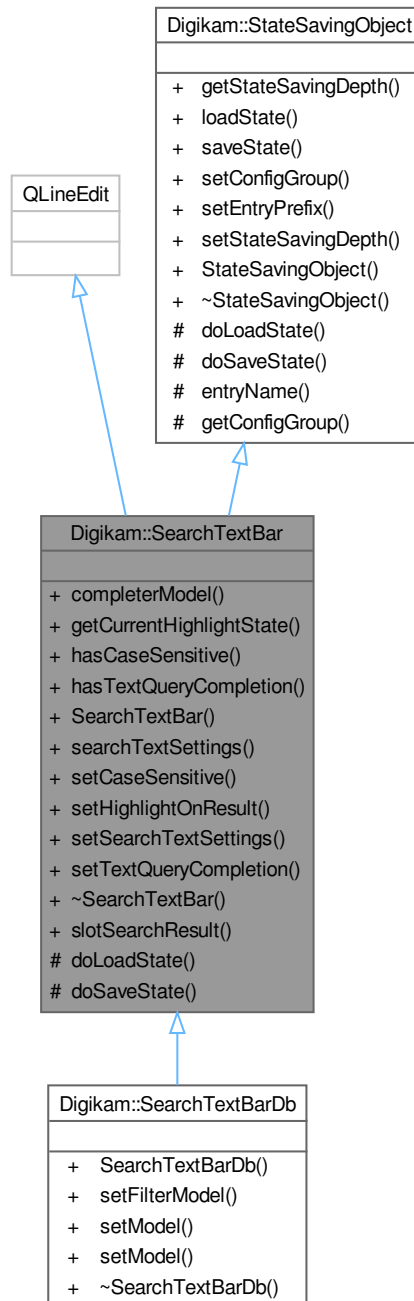
- void `searchShallBeSelected` (const QList< [Album](#) \* > &albums)

### Public Member Functions

- `SearchTabHeader` ([QWidget](#) \*const parent)

## 6.1280 Digikam::SearchTextBar Class Reference

Inheritance diagram for Digikam::SearchTextBar:



### Public Types

- enum `HighlightState` { `NEUTRAL` , `HAS_RESULT` , `NO_RESULT` }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots

- void [slotSearchResult](#) (bool match)

## Signals

- void [completerActivated](#) ()
- void [completerHighlighted](#) (int albumId)
- void [signalSearchTextSettings](#) (const [SearchTextSettings](#) &settings)

## Public Member Functions

- [ModelCompleter](#) \* [completerModel](#) () const
- [HighlightState](#) [getCurrentHighlightState](#) () const
- bool [hasCaseSensitive](#) () const
- bool [hasTextQueryCompletion](#) () const
- [SearchTextBar](#) (QWidget \*const parent, const QString &name, const QString &msg=QString())
- [SearchTextSettings](#) [searchTextSettings](#) () const
- void [setCaseSensitive](#) (bool b)
- void [setHighlightOnResult](#) (bool highlight)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [setTextQueryCompletion](#) (bool b)

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Protected Member Functions

- void [doLoadState](#) () override
- void [doSaveState](#) () override

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.1280.1 Detailed Description

A text input for searching entries with visual feedback. Can be used on QAbstractItemModels.

## 6.1280.2 Member Enumeration Documentation

### 6.1280.2.1 HighlightState

```
enum Digikam::SearchTextBar::HighlightState
```

Possible highlighting states a [SearchTextBar](#) can have.

Enumerator

NEUTRAL	No highlighting at all. Background is colored in a neutral way according to the theme.
HAS_RESULT	The background color of the text input indicates that a result was found.
NO_RESULT	The background color indicates that no result was found.

## 6.1280.3 Member Function Documentation

### 6.1280.3.1 doLoadState()

```
void Digikam::SearchTextBar::doLoadState ( ) [override], [protected], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1280.3.2 doSaveState()

```
void Digikam::SearchTextBar::doSaveState ( ) [override], [protected], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1280.3.3 getCurrentHighlightState()

```
SearchTextBar::HighlightState Digikam::SearchTextBar::getCurrentHighlightState ( ) const
```

Tells the current highlighting state of the text input indicated via the background color.

Returns

current highlight state

### 6.1280.3.4 setCaseSensitive()

```
void Digikam::SearchTextBar::setCaseSensitive (
    bool b )
```

Indicate whether this search text bar can be toggled to between case- sensitive and -insensitive or if always case-insensitive shall be used.

## Parameters

<i>b</i>	if <code>true</code> the user can decide the toggle between case sensitivity, on <code>false</code> every search is case- insensitive
----------	---

**6.1280.3.5 setHighlightOnResult()**

```
void Digikam::SearchTextBar::setHighlightOnResult (
    bool highlight )
```

Tells whether highlighting for found search results shall be used or not (green and red).

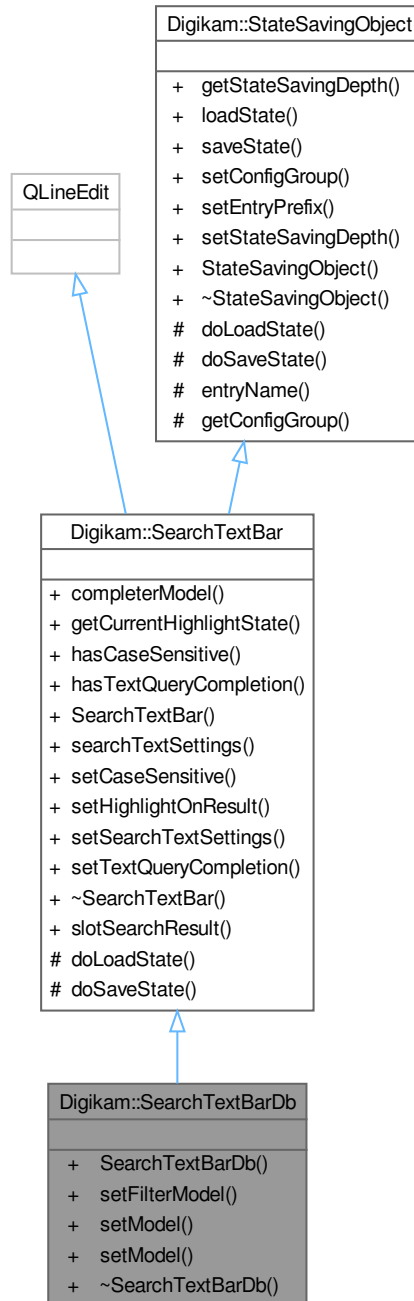
Default behavior has highlighting enabled.

## Parameters

<i>highlight</i>	<code>true</code> activates green and red highlighting, with <code>false</code> the normal widget background color will be displayed always
------------------	---

## 6.1281 Digikam::SearchTextBarDb Class Reference

Inheritance diagram for Digikam::SearchTextBarDb:



### Public Member Functions

- **SearchTextBarDb** (QWidget \*const parent, const QString &name, const QString &msg=QString())
- void **setFilterModel** (AlbumFilterModel \*const filterModel)
- void **setModel** (AbstractAlbumModel \*const model)
- void **setModel** (QAbstractItemModel \*model, int uniqueIdRole, int displayRole=Qt::DisplayRole)

## Public Member Functions inherited from [Digikam::SearchTextBar](#)

- [ModelCompleter](#) \* **completerModel** () const
- [HighlightState](#) **getCurrentHighlightState** () const
- bool **hasCaseSensitive** () const
- bool **hasTextQueryCompletion** () const
- [SearchTextBar](#) (QWidget \*const parent, const QString &name, const QString &msg=QString())
- [SearchTextSettings](#) **searchTextSettings** () const
- void **setCaseSensitive** (bool b)
- void **setHighlightOnResult** (bool highlight)
- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)
- void **setTextQueryCompletion** (bool b)

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::SearchTextBar](#)

- enum [HighlightState](#) { [NEUTRAL](#) , [HAS\\_RESULT](#) , [NO\\_RESULT](#) }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots inherited from [Digikam::SearchTextBar](#)

- void **slotSearchResult** (bool match)

## Signals inherited from [Digikam::SearchTextBar](#)

- void **completerActivated** ()
- void **completerHighlighted** (int albumId)
- void **signalSearchTextSettings** (const [SearchTextSettings](#) &settings)

## Protected Member Functions inherited from [Digikam::SearchTextBar](#)

- void **doLoadState** () override
- void **doSaveState** () override



## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

### 6.1281.1 Detailed Description

A text input for searching entries with visual feedback. Can be used on Database Models.

#### Author

Gilles Caulier

### 6.1281.2 Member Function Documentation

#### 6.1281.2.1 [setFilterModel\(\)](#)

```
void Digikam::SearchTextBarDb::setFilterModel (
    AlbumFilterModel *const filterModel )
```

Sets the filter model this text bar shall use to invoke filtering on and reading the result for highlighting from.

#### Parameters

<i>filterModel</i>	filter model to use for filtering. <code>null</code> means there is no model to use and external connections need to be created with <code>signalSearchTextSettings</code> and <code>slotSearchResult</code>
--------------------	--

#### 6.1281.2.2 [setModel\(\)](#) [1/2]

```
void Digikam::SearchTextBarDb::setModel (
    AbstractAlbumModel *const model )
```

Sets the album model this text bar shall use to invoke filtering on and reading the result for highlighting from.

#### Parameters

<i>model</i>	album model to use for filtering. <code>null</code> means there is no model to use and external connections need to be created with <code>signalSearchTextSettings</code> and <code>slotSearchResult</code>
--------------	---

#### 6.1281.2.3 [setModel\(\)](#) [2/2]

```
void Digikam::SearchTextBarDb::setModel (
    QAbstractItemModel * model,
    int uniqueIdRole,
    int displayRole = Qt::DisplayRole )
```

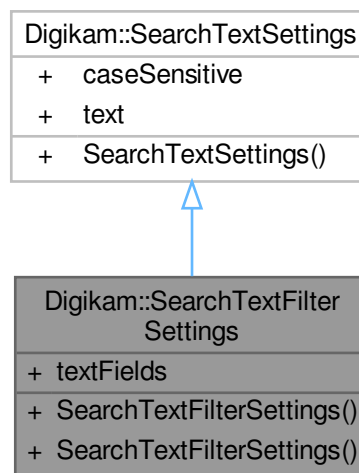
If the given model is `!= null`, the model is used to populate the completion for this text field.

## Parameters

<i>model</i>	to fill from or null for manual mode
<i>uniqueIdRole</i>	a role for which the model will return a unique integer for each entry
<i>displayRole</i>	the role to retrieve the text for completion, default is Qt::DisplayRole.

## 6.1282 Digikam::SearchTextFilterSettings Class Reference

Inheritance diagram for Digikam::SearchTextFilterSettings:



### Public Types

- enum `TextFilterFields` {
  - `None` = 0x00 , `ImageName` = 0x01 , `ImageTitle` = 0x02 , `ImageComment` = 0x04 ,
  - `TagName` = 0x08 , `AlbumName` = 0x10 , `ImageAspectRatio` = 0x20 , `ImagePixelSize` = 0x40 ,
  - `All` = ImageName | ImageTitle | ImageComment | TagName | AlbumName | ImageAspectRatio | ImagePixelSize }

### Public Member Functions

- `SearchTextFilterSettings` (const [SearchTextSettings](#) &settings)

### Public Attributes

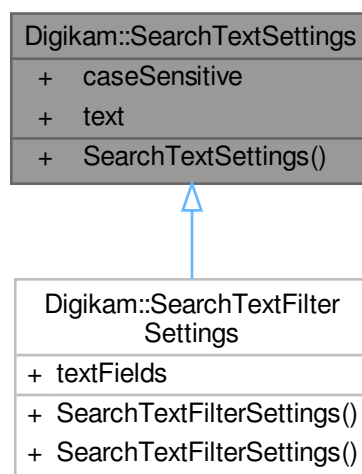
- TextFilterFields `textFields` = None

## Public Attributes inherited from [Digikam::SearchTextSettings](#)

- Qt::CaseSensitivity **caseSensitive** = Qt::CaseInsensitive
- QString **text**

## 6.1283 Digikam::SearchTextSettings Class Reference

Inheritance diagram for Digikam::SearchTextSettings:

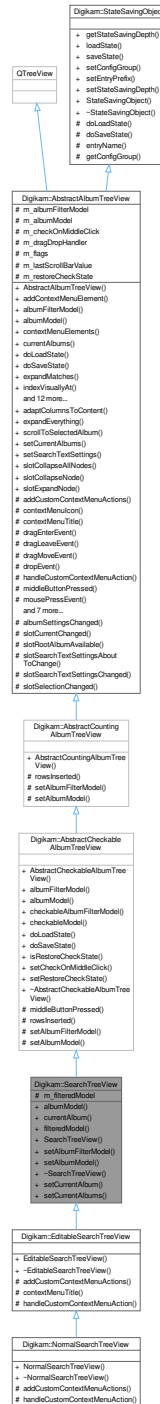


### Public Attributes

- Qt::CaseSensitivity **caseSensitive** = Qt::CaseInsensitive
- QString **text**

## 6.1284 Digikam::SearchTreeView Class Reference

Inheritance diagram for Digikam::SearchTreeView:



### Public Slots

- void **setCurrentAlbum** (int searchId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< Album \* > &albums, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()
  - slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()
  - slotExpandNode - expands recursively selected nodes*

## Public Member Functions

- [searchModel](#) \* [albumModel](#) () const
  - Note: not filtered by search type.*
- [SAlbum](#) \* [currentAlbum](#) () const
- [SearchFilterModel](#) \* [filteredModel](#) () const
  - Contains only the searches with appropriate type - prefer to [albumModel\(\)](#)*
- [SearchTreeView](#) (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)
- void [setAlbumFilterModel](#) ([SearchFilterModel](#) \*const [filteredModel](#), [CheckableAlbumFilterModel](#) \*const model)
- void [setAlbumModel](#) ([searchModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- [CheckableAlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [albumModel](#) () const
- [CheckableAlbumFilterModel](#) \* [checkableAlbumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [checkableModel](#) () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Protected Attributes

- [SearchFilterModel](#) \* **m\_filteredModel** = nullptr

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* **m\_albumFilterModel** = nullptr
- [AbstractSpecificAlbumModel](#) \* **m\_albumModel** = nullptr
- bool **m\_checkOnMiddleClick** = false
- [AlbumModelDragDropHandler](#) \* **m\_dragDropHandler** = nullptr
- Flags **m\_flags** = DefaultFlags
- int **m\_lastScrollBarValue** = 0
- bool **m\_restoreCheckState** = false

## Additional Inherited Members

### Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) { [CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) , [AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const [QList](#)< [Album](#) \* > &selectedAlbums)

### Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

### Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

## Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album)
- virtual QPixmap [contextMenuIcon](#) () const
- virtual QString [contextMenuTitle](#) () const
- void **dragEnterEvent** ([QDragEnterEvent](#) \*e) override
- void **dragLeaveEvent** ([QDragLeaveEvent](#) \*e) override
- void **dragMoveEvent** ([QDragMoveEvent](#) \*e) override
- void **dropEvent** ([QDropEvent](#) \*e) override
- virtual void [handleCustomContextMenuAction](#) ([QAction](#) \*action, const [AlbumPointer](#)< [Album](#) > &album)
- void **mousePressEvent** ([QMouseEvent](#) \*e) override

*Other helper methods.*

- virtual QPixmap [pixmapForDrag](#) (const [QStyleOptionViewItem](#) &option, [QList](#)< [QModelIndex](#) > indexes)
- void **rowsAboutToBeRemoved** (const [QModelIndex](#) &parent, int start, int end) override
- void **rowsInserted** (const [QModelIndex](#) &index, int start, int end) override
- void **setAlbumFilterModel** ([AlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([AbstractSpecificAlbumModel](#) \*const model)
- virtual bool [showContextMenuAt](#) ([QContextMenuEvent](#) \*event, [Album](#) \*albumForEvent)
- void **startDrag** ([Qt::DropActions](#) supportedActions) override

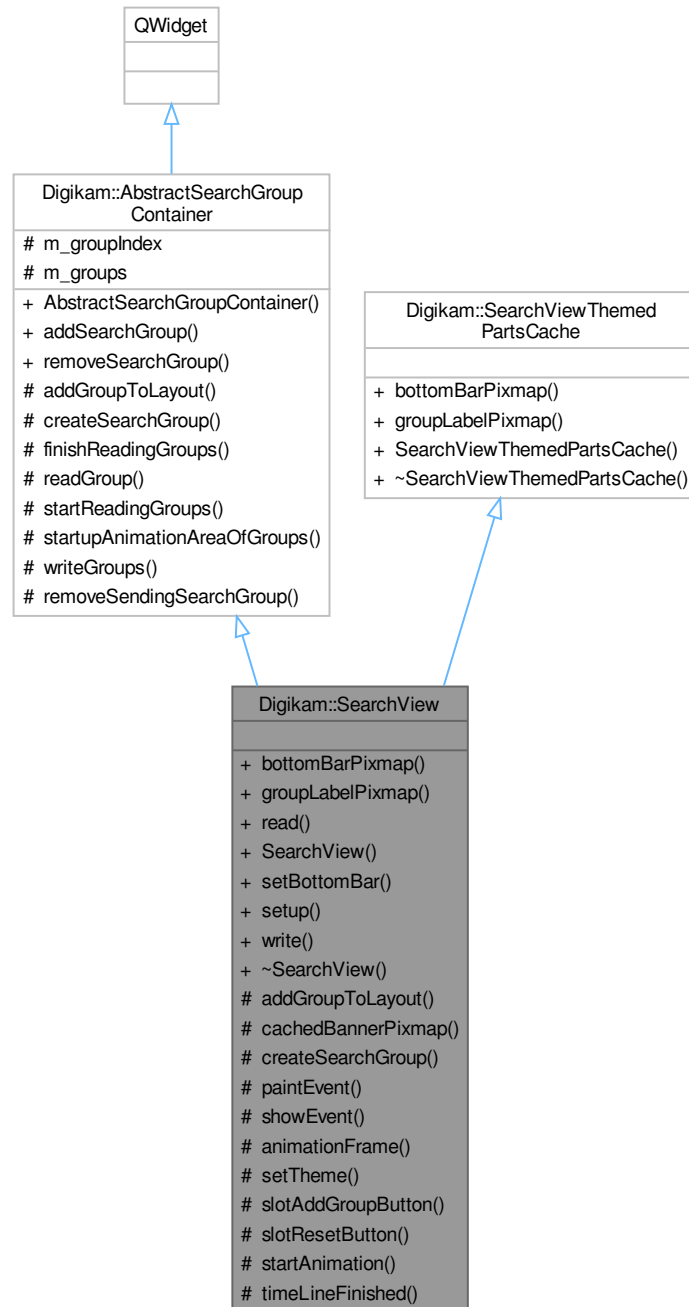
## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- [KConfigGroup](#) [getConfigGroup](#) () const



## 6.1285 Digikam::SearchView Class Reference

Inheritance diagram for Digikam::SearchView:



### Signals

- void **searchCancel** ()
- void **searchOk** ()
- void **searchTryout** ()

### Public Member Functions

- QPixmap [bottomBarPixmap](#) (int w, int h) override
- QPixmap [groupLabelPixmap](#) (int w, int h) override
- void **read** (const QString &search)
- void **setBottomBar** ([SearchViewBottomBar](#) \*const bar)
- void **setup** ()
- QString **write** () const

### Public Member Functions inherited from [Digikam::AbstractSearchGroupContainer](#)

- [AbstractSearchGroupContainer](#) (QWidget \*const parent=nullptr)

### Protected Slots

- void **animationFrame** (int)
- void **setTheme** ()
- void **slotAddGroupButton** ()
- void **slotResetButton** ()
- void **startAnimation** ()
- void **timeLineFinished** ()

### Protected Slots inherited from [Digikam::AbstractSearchGroupContainer](#)

- void **removeSendingSearchGroup** ()

### Protected Member Functions

- void [addGroupToLayout](#) ([SearchGroup](#) \*group) override  
*Re-implement: Adds a newly created group to the layout structures.*
- QPixmap **cachedBannerPixmap** (int w, int h) const
- [SearchGroup](#) \* [createSearchGroup](#) () override  
*Re-implement: create and setup a search group.*
- void **paintEvent** (QPaintEvent \*e) override
- void **showEvent** (QShowEvent \*event) override

### Protected Member Functions inherited from [Digikam::AbstractSearchGroupContainer](#)

- void **finishReadingGroups** ()  
*Call when the XML part is finished.*
- void **readGroup** ([SearchXmlCachingReader](#) &reader)  
*Call when a group element is the current element.*
- void **startReadingGroups** ([SearchXmlCachingReader](#) &reader)  
*Call before reading the XML part that could contain group elements.*
- QList< QRect > **startupAnimationAreaOfGroups** () const  
*Collects the data from the same method of all contained groups (position relative to this widget)*
- void **writeGroups** ([SearchXmlWriter](#) &writer) const  
*Write contained groups to writer.*

## Additional Inherited Members

### Public Slots inherited from [Digikam::AbstractSearchGroupContainer](#)

- [SearchGroup](#) \* **addSearchGroup** ()
- void **removeSearchGroup** ([SearchGroup](#) \*group)

### Protected Attributes inherited from [Digikam::AbstractSearchGroupContainer](#)

- int **m\_groupIndex** = 0
- QList< [SearchGroup](#) \* > **m\_groups**

## 6.1285.1 Member Function Documentation

### 6.1285.1.1 addGroupToLayout()

```
void Digikam::SearchView::addGroupToLayout (
    SearchGroup * group ) [override], [protected], [virtual]
```

Implements [Digikam::AbstractSearchGroupContainer](#).

### 6.1285.1.2 bottomBarPixmap()

```
QPixmap Digikam::SearchView::bottomBarPixmap (
    int w,
    int h ) [override], [virtual]
```

Implements [Digikam::SearchViewThemedPartsCache](#).

### 6.1285.1.3 createSearchGroup()

```
SearchGroup * Digikam::SearchView::createSearchGroup ( ) [override], [protected], [virtual]
```

Implements [Digikam::AbstractSearchGroupContainer](#).

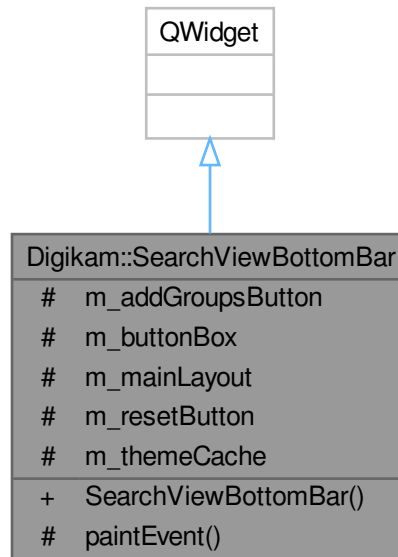
### 6.1285.1.4 groupLabelPixmap()

```
QPixmap Digikam::SearchView::groupLabelPixmap (
    int w,
    int h ) [override], [virtual]
```

Implements [Digikam::SearchViewThemedPartsCache](#).

## 6.1286 Digikam::SearchViewBottomBar Class Reference

Inheritance diagram for Digikam::SearchViewBottomBar:



### Signals

- void **addGroupPressed** ()
- void **cancelPressed** ()
- void **okPressed** ()
- void **resetPressed** ()
- void **tryoutPressed** ()

### Public Member Functions

- **SearchViewBottomBar** ([SearchViewThemedPartsCache](#) \*const cache, QWidget \*const parent=nullptr)

### Protected Member Functions

- void **paintEvent** (QPaintEvent \*) override

### Protected Attributes

- QPushButton \* **m\_addGroupsButton** = nullptr
- QDialogButtonBox \* **m\_buttonBox** = nullptr
- QHBoxLayout \* **m\_mainLayout** = nullptr
- QPushButton \* **m\_resetButton** = nullptr
- [SearchViewThemedPartsCache](#) \* **m\_themeCache** = nullptr

## 6.1287 Digikam::SearchViewThemedPartsCache Class Reference

Inheritance diagram for Digikam::SearchViewThemedPartsCache:

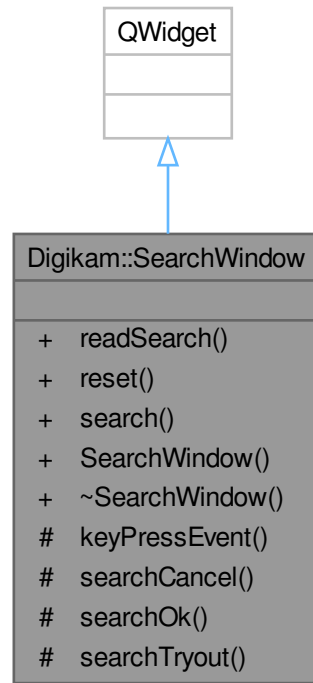


### Public Member Functions

- virtual QPixmap **bottomBarPixmap** (int w, int h)=0
- virtual QPixmap **groupLabelPixmap** (int w, int h)=0

## 6.1288 Digikam::SearchWindow Class Reference

Inheritance diagram for Digikam::SearchWindow:



### Signals

- void [searchEdited](#) (int id, const QString &query)

### Public Member Functions

- void [readSearch](#) (int id, const QString &query)
- void [reset](#) ()
- QString [search](#) () const
- [SearchWindow](#) ()

### Protected Slots

- void [searchCancel](#) ()
- void [searchOk](#) ()
- void [searchTryout](#) ()

### Protected Member Functions

- void [keyPressEvent](#) (QKeyEvent \*) override

## 6.1288.1 Constructor & Destructor Documentation

### 6.1288.1.1 SearchWindow()

```
Digikam::SearchWindow::SearchWindow ( )
```

Create a new [SearchWindow](#) with an empty advanced search

## 6.1288.2 Member Function Documentation

### 6.1288.2.1 readSearch()

```
void Digikam::SearchWindow::readSearch (
    int id,
    const QString & query )
```

Read the given search into the search widgets. The id will be emitted with the searchEdited signal.

### 6.1288.2.2 reset()

```
void Digikam::SearchWindow::reset ( )
```

Reset the search widget to an empty search. Current id is -1.

### 6.1288.2.3 search()

```
QString Digikam::SearchWindow::search ( ) const
```

Returns the currently produced search string

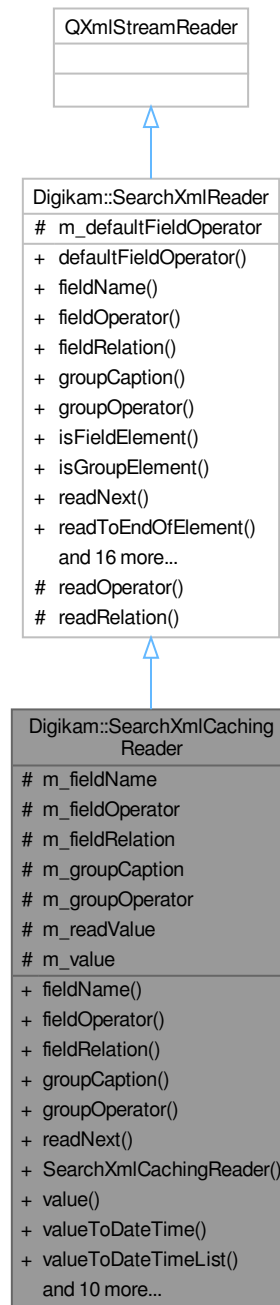
### 6.1288.2.4 searchEdited

```
void Digikam::SearchWindow::searchEdited (
    int id,
    const QString & query ) [signal]
```

Signals that the user has finished editing the search. The given query is the same as [search\(\)](#).

## 6.1289 Digikam::SearchXmlCachingReader Class Reference

Inheritance diagram for Digikam::SearchXmlCachingReader:



### Public Member Functions

- QString **fieldName** () const
- SearchXml::Operator **fieldOperator** () const



- SearchXml::Relation **fieldRelation** () const
- QString **groupCaption** () const
- SearchXml::Operator **groupOperator** () const
- SearchXml::Element **readNext** ()
- [SearchXmlCachingReader](#) (const QString &xml)
- QString **value** ()
- QDateTime **valueToDateTime** ()
- QList< QDateTime > **valueToDateTimeList** ()
- double **valueToDouble** ()
- QList< double > **valueToDoubleList** ()
- QList< double > **valueToDoubleOrDoubleList** ()
- int **valueToInt** ()
- QList< int > **valueToIntList** ()
- QList< int > **valueToIntOrIntList** ()
- qlonglong **valueToLongLong** ()
- QList< qlonglong > **valueToLongLongList** ()
- QStringList **valueToStringList** ()
- QList< QString > **valueToStringOrStringList** ()

## Public Member Functions inherited from [Digikam::SearchXmlReader](#)

- SearchXml::Operator [defaultFieldOperator](#) () const
- QString **fieldName** () const
- SearchXml::Operator [fieldOperator](#) () const
- SearchXml::Relation **fieldRelation** () const
- QString [groupCaption](#) () const
- SearchXml::Operator [groupOperator](#) () const
- bool [isFieldElement](#) () const
- bool [isGroupElement](#) () const
- SearchXml::Element [readNext](#) ()
- void [readToEndOfElement](#) ()
- void [readToFirstField](#) ()
- bool [readToStartOfElement](#) (const QString &name)
- [SearchXmlReader](#) (const QString &xml)
- QString [value](#) ()
- QDateTime **valueToDateTime** ()
- QList< QDateTime > **valueToDateTimeList** ()
- double **valueToDouble** ()
- QList< double > **valueToDoubleList** ()
- QList< double > **valueToDoubleOrDoubleList** ()
- int **valueToInt** ()
- QList< int > **valueToIntList** ()
- QList< int > **valueToIntOrIntList** ()
- qlonglong **valueToLongLong** ()
- QList< qlonglong > **valueToLongLongList** ()
- QStringList **valueToStringList** ()
- QList< QString > **valueToStringOrStringList** ()

## Protected Attributes

- QString **m\_fieldName**
- SearchXml::Operator **m\_fieldOperator** = SearchXml::And
- SearchXml::Relation **m\_fieldRelation** = SearchXml::Equal
- QString **m\_groupCaption**
- SearchXml::Operator **m\_groupOperator** = SearchXml::And
- bool **m\_readValue** = false
- QVariant **m\_value**

## Protected Attributes inherited from [Digikam::SearchXmlReader](#)

- SearchXml::Operator `m_defaultFieldOperator`

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::SearchXmlReader](#)

- SearchXml::Operator `readOperator` (const QString &, SearchXml::Operator) const
- SearchXml::Relation `readRelation` (const QString &, SearchXml::Relation) const

## 6.1289.1 Constructor & Destructor Documentation

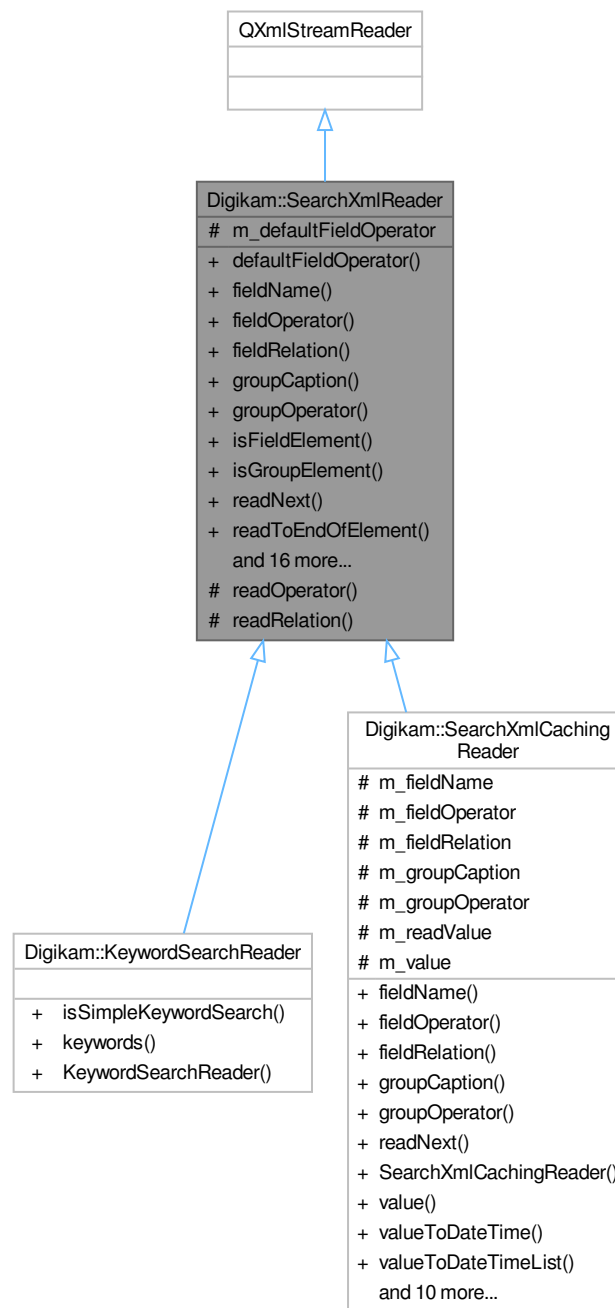
### 6.1289.1.1 SearchXmlCachingReader()

```
Digikam::SearchXmlCachingReader::SearchXmlCachingReader (  
    const QString & xml ) [explicit]
```

This class has the same semantics as [SearchXmlReader](#), but performs some caching and is thus much more relaxed than [SearchXmlReader](#) about the calling order of methods: With this class, you can access properties of a group until the next group is read, access properties and the value of a field until the next field is read, with all calls possible multiple times.

## 6.1290 Digikam::SearchXmlReader Class Reference

Inheritance diagram for Digikam::SearchXmlReader:



### Public Member Functions

- SearchXml::Operator `defaultFieldOperator` () const
- QString `fieldName` () const

- SearchXml::Operator [fieldOperator](#) () const
- SearchXml::Relation **fieldRelation** () const
- QString [groupCaption](#) () const
- SearchXml::Operator [groupOperator](#) () const
- bool [isFieldElement](#) () const
- bool [isGroupElement](#) () const
- SearchXml::Element [readNext](#) ()
- void [readToEndOfElement](#) ()
- void [readToFirstField](#) ()
- bool [readToStartOfElement](#) (const QString &name)
- **SearchXmlReader** (const QString &xml)
- QString [value](#) ()
- QDateTime [valueToDateTime](#) ()
- QList< QDateTime > [valueToDateTimeList](#) ()
- double [valueToDouble](#) ()
- QList< double > [valueToDoubleList](#) ()
- QList< double > [valueToDoubleOrDoubleList](#) ()
- int [valueToInt](#) ()
- QList< int > [valueToIntList](#) ()
- QList< int > [valueToIntOrIntList](#) ()
- qlonglong [valueToLongLong](#) ()
- QList< qlonglong > [valueToLongLongList](#) ()
- QStringList [valueToStringList](#) ()
- QList< QString > [valueToStringOrStringList](#) ()

### Protected Member Functions

- SearchXml::Operator **readOperator** (const QString &, SearchXml::Operator) const
- SearchXml::Relation **readRelation** (const QString &, SearchXml::Relation) const

### Protected Attributes

- SearchXml::Operator **m\_defaultFieldOperator**

## 6.1290.1 Member Function Documentation

### 6.1290.1.1 defaultFieldOperator()

```
SearchXml::Operator Digikam::SearchXmlReader::defaultFieldOperator ( ) const
```

Returns the default field operator. This operator can be overridden by a specific [fieldOperator\(\)](#).

### 6.1290.1.2 fieldOperator()

```
SearchXml::Operator Digikam::SearchXmlReader::fieldOperator ( ) const
```

Returns the field attributes. Only valid if the current element is a field. `fieldOperator` returns the default operator if the field has not specified any.

### 6.1290.1.3 groupCaption()

```
QString Digikam::SearchXmlReader::groupCaption ( ) const
```

Returns the (optional) group caption. Only valid if the current element is a group.

### 6.1290.1.4 groupOperator()

```
SearchXml::Operator Digikam::SearchXmlReader::groupOperator ( ) const
```

Returns the group operator. Only valid if the current element is a group.

### 6.1290.1.5 isFieldElement()

```
bool Digikam::SearchXmlReader::isFieldElement ( ) const
```

Returns if the current element is a field element (start or end element).

### 6.1290.1.6 isGroupElement()

```
bool Digikam::SearchXmlReader::isGroupElement ( ) const
```

Returns if the current element is a group element (start or end element).

### 6.1290.1.7 readNext()

```
SearchXml::Element Digikam::SearchXmlReader::readNext ( )
```

Continue parsing the document. Returns the type of the current element.

### 6.1290.1.8 readToEndOfElement()

```
void Digikam::SearchXmlReader::readToEndOfElement ( )
```

General helper method: Reads XML until the end element of the current start element is reached.

### 6.1290.1.9 readToFirstField()

```
void Digikam::SearchXmlReader::readToFirstField ( )
```

General helper method: Reads XML until the first field of the next or first found group is reached.

**6.1290.1.10 readToStartOfElement()**

```
bool Digikam::SearchXmlReader::readToStartOfElement (
    const QString & name )
```

General helper method: Reads XML a start element with the given name is found. The method goes to the next start element, and from there down the hierarchy, but not further up in the hierarchy. Returns false if the element is not found.

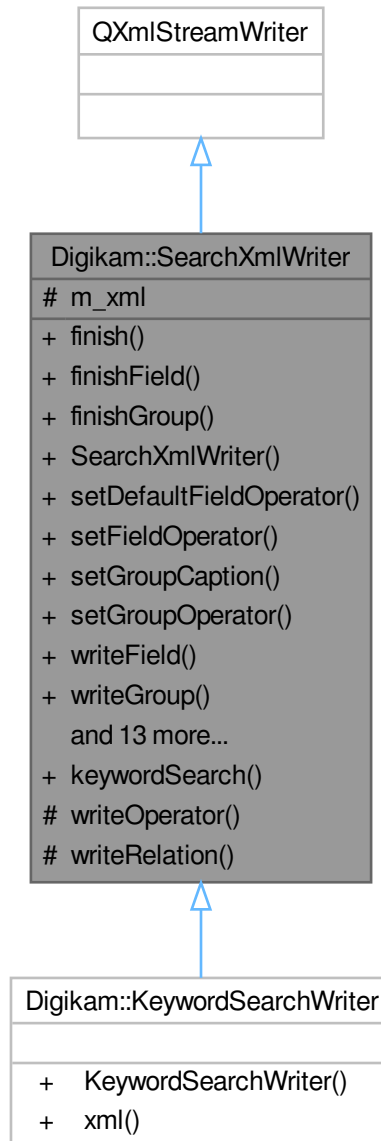
**6.1290.1.11 value()**

```
QString Digikam::SearchXmlReader::value ( )
```

Returns the field values. Only valid if the current element is a field. This reads to the end element of the field, and converts the found text/elements to the desired output.

## 6.1291 Digikam::SearchXmlWriter Class Reference

Inheritance diagram for Digikam::SearchXmlWriter:



### Public Member Functions

- void [finish](#) ()
- void [finishField](#) ()
- void [finishGroup](#) ()
- [SearchXmlWriter](#) ()
- void [setDefaultFieldOperator](#) (SearchXml::Operator op)

- void [setFieldOperator](#) (SearchXml::Operator op)
- void [setGroupCaption](#) (const QString &caption)
- void [setGroupOperator](#) (SearchXml::Operator op)
- void [writeField](#) (const QString &name, SearchXml::Relation relation)
- void [writeGroup](#) ()
- void [writeValue](#) (const QDateTime &dateTime)
- void [writeValue](#) (const QList< double > &valueList, int precision=8)
- void [writeValue](#) (const QList< float > &valueList, int precision=6)
- void [writeValue](#) (const QList< int > &valueList)
- void [writeValue](#) (const QList< QDateTime > &valueList)
- void [writeValue](#) (const QList< qlonglong > &valueList)
- void [writeValue](#) (const QString &value)
- void [writeValue](#) (const QStringList &valueList)
- void [writeValue](#) (double value, int precision=8)
- void [writeValue](#) (float value, int precision=6)
- void [writeValue](#) (int value)
- void [writeValue](#) (qlonglong value)
- QString [xml](#) () const

### Static Public Member Functions

- static QString [keywordSearch](#) (const QString &keyword)

### Protected Member Functions

- void [writeOperator](#) (const QString &, SearchXml::Operator)
- void [writeRelation](#) (const QString &, SearchXml::Relation)

### Protected Attributes

- QString [m\\_xml](#)

## 6.1291.1 Constructor & Destructor Documentation

### 6.1291.1.1 SearchXmlWriter()

```
Digikam::SearchXmlWriter::SearchXmlWriter ( ) [explicit]
```

Note that [SearchXmlWriter](#) and [SearchXmlGroupWriter](#) rely on you calling the methods following the restrictions set by the documentation; Otherwise you will not produce the desired output.

## 6.1291.2 Member Function Documentation

### 6.1291.2.1 finish()

```
void Digikam::SearchXmlWriter::finish ( )
```

Finish the XML. No further group can be added after calling this. You need to call this before you can get the resulting XML from [xml\(\)](#).



### 6.1291.2.2 finishField()

```
void Digikam::SearchXmlWriter::finishField ( )
```

Finish writing the current field. You shall call this method before adding another field, or closing the group.

### 6.1291.2.3 finishGroup()

```
void Digikam::SearchXmlWriter::finishGroup ( )
```

Finish the current group. You cannot add anymore fields after calling this. Note that you will want to call this before writing another group if you want the group on the same level. You can as well add nested groups and call this to close the group afterwards.

### 6.1291.2.4 keywordSearch()

```
QString Digikam::SearchXmlWriter::keywordSearch (
    const QString & keyword ) [static]
```

Returns ready-made XML for a query of type "keyword" with the specified text as keyword.

### 6.1291.2.5 setDefaultFieldOperator()

```
void Digikam::SearchXmlWriter::setDefaultFieldOperator (
    SearchXml::Operator op )
```

Sets the default operator for fields in this group "(field1 AND field2 AND ... fieldn)". The default operator can in each field be overridden. Default value is AND.

### 6.1291.2.6 setFieldOperator()

```
void Digikam::SearchXmlWriter::setFieldOperator (
    SearchXml::Operator op )
```

Adds an optional operator overriding the default field operator of the group.

### 6.1291.2.7 setGroupCaption()

```
void Digikam::SearchXmlWriter::setGroupCaption (
    const QString & caption )
```

Sets an optional caption.

### 6.1291.2.8 setGroupOperator()

```
void Digikam::SearchXmlWriter::setGroupOperator (
    SearchXml::Operator op )
```

Sets the operator applied to the group as a whole "OR (field1 ... fieldn)". Default value is OR.

### 6.1291.2.9 writeField()

```
void Digikam::SearchXmlWriter::writeField (
    const QString & name,
    SearchXml::Relation relation )
```

Adds a new field with the given name (entity) and relation, "Rating less than ...". Ensure that you closed the previous field with [finishField\(\)](#). For a reference of valid field names, look into [ItemQueryBuilder](#). The general rule is that names are like the database fields, but all lower-case.

### 6.1291.2.10 writeGroup()

```
void Digikam::SearchXmlWriter::writeGroup ( )
```

Adds a group. Use the returned group writer to add fields.

### 6.1291.2.11 writeValue()

```
void Digikam::SearchXmlWriter::writeValue (
    const QString & value )
```

Adds the value, "4" in the case of "Rating less than 4".

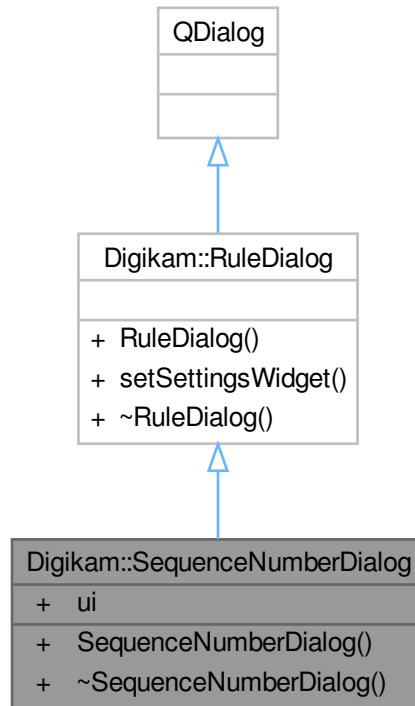
### 6.1291.2.12 xml()

```
QString Digikam::SearchXmlWriter::xml ( ) const
```

Get the created XML. The value is only valid if [finish\(\)](#) has been called.

## 6.1292 Digikam::SequenceNumberDialog Class Reference

Inheritance diagram for Digikam::SequenceNumberDialog:



### Public Member Functions

- **SequenceNumberDialog** ([Rule](#) \*const parent)

### Public Member Functions inherited from [Digikam::RuleDialog](#)

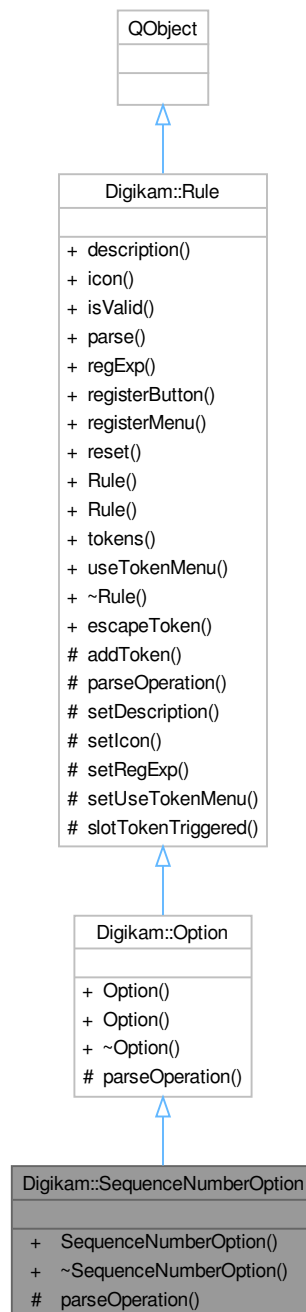
- **RuleDialog** ([Rule](#) \*const parent)
- void **setSettingsWidget** (QWidget \*const settingsWidget)

### Public Attributes

- Ui::SequenceNumberOptionDialogWidget \*const **ui** = nullptr

## 6.1293 Digikam::SequenceNumberOption Class Reference

Inheritance diagram for Digikam::SequenceNumberOption:



### Protected Member Functions

- QString `parseOperation` (`ParseSettings` &settings, const `QRegularExpressionMatch` &match) override

## Protected Member Functions inherited from Digikam::Rule

- bool [addToken](#) (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void [setUseTokenMenu](#) (bool value)

## Additional Inherited Members

## Public Types inherited from Digikam::Rule

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from Digikam::Rule

- void **signalTokenTriggered** (const QString &)

## Public Member Functions inherited from Digikam::Option

- **Option** (const QString &name, const QString &description)
- **Option** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from Digikam::Rule

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- **ParseResults** **parse** (**ParseSettings** &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Static Public Member Functions inherited from Digikam::Rule

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from Digikam::Rule

- virtual void **slotTokenTriggered** (const QString &)

## 6.1293.1 Member Function Documentation

### 6.1293.1.1 parseOperation()

```
QString Digikam::SequenceNumberOption::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [protected], [virtual]
```

TODO: describe me

**Parameters**

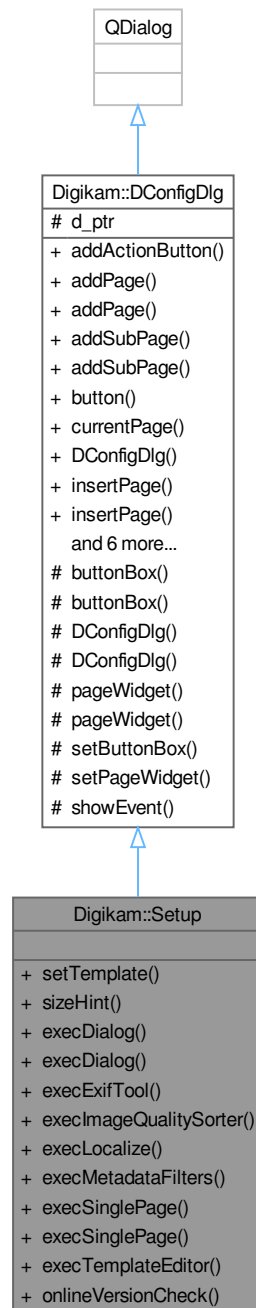
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

**Returns**

Implements [Digikam::Option](#).

## 6.1294 Digikam::Setup Class Reference

Inheritance diagram for Digikam::Setup:



### Public Types

- enum **Page** {  
**LastPageUsed** = -1 , **DatabasePage** = 0 , **CollectionsPage** , **AlbumViewPage** ,

**ToolTipPage** , **MetadataPage** , **TemplatePage** , **EditorPage** ,  
**ICCPPage** , **LightTablePage** , **ImageQualityPage** , **CameraPage** ,  
**PluginsPage** , **MiscellaneousPage** , **SetupPageEnumLast** }

## Public Types inherited from [Digikam::DConfigDlg](#)

- enum [FaceType](#) {  
**Auto** = DConfigDlgView::Auto , **Plain** = DConfigDlgView::Plain , **List** = DConfigDlgView::List , **Tree** =  
DConfigDlgView::Tree ,  
**Tabbed** = DConfigDlgView::Tabbed }

## Public Member Functions

- void **setTemplate** (const [Template](#) &t)
- QSize **sizeHint** () const override

## Public Member Functions inherited from [Digikam::DConfigDlg](#)

- void **addActionButton** (QAbstractButton \*const [button](#))
- void **addPage** ([DConfigDlgWdgItem](#) \*const [item](#))
- [DConfigDlgWdgItem](#) \* **addPage** (QWidget \*const [widget](#), const QString &[name](#))
- void **addSubPage** ([DConfigDlgWdgItem](#) \*const [parent](#), [DConfigDlgWdgItem](#) \*const [item](#))
- [DConfigDlgWdgItem](#) \* **addSubPage** ([DConfigDlgWdgItem](#) \*const [parent](#), QWidget \*const [widget](#), const  
QString &[name](#))
- QPushButton \* **button** (QDialogButtonBox::StandardButton which) const
- [DConfigDlgWdgItem](#) \* **currentPage** () const
- [DConfigDlg](#) (QWidget \*const [parent](#)=nullptr, Qt::WindowFlags [flags](#)=Qt::WindowFlags())
- void **insertPage** ([DConfigDlgWdgItem](#) \*const [before](#), [DConfigDlgWdgItem](#) \*const [item](#))
- [DConfigDlgWdgItem](#) \* **insertPage** ([DConfigDlgWdgItem](#) \*const [before](#), QWidget \*const [widget](#), const  
QString &[name](#))
- void **removePage** ([DConfigDlgWdgItem](#) \*const [item](#))
- void **setConfigGroup** (const QString &[group](#))
- void **setCurrentPage** ([DConfigDlgWdgItem](#) \*const [item](#))
- void **setFaceType** ([FaceType](#) [faceType](#))
- void **setStandardButtons** (QDialogButtonBox::StandardButtons [buttons](#))
- ~[DConfigDlg](#) () override

## Static Public Member Functions

- static bool **execDialog** (Page [page](#)=LastPageUsed)
- static bool **execDialog** (QWidget \*const [parent](#), Page [page](#)=LastPageUsed)
- static bool **execExifTool** (QWidget \*const [parent](#))
- static bool **execImageQualitySorter** (QWidget \*const [parent](#))
- static bool **execLocalize** (QWidget \*const [parent](#))
- static bool **execMetadataFilters** (QWidget \*const [parent](#), int [tab](#))
- static bool **execSinglePage** (Page [page](#))
- static bool **execSinglePage** (QWidget \*const [parent](#), Page [page](#))
- static bool **execTemplateEditor** (QWidget \*const [parent](#), const [Template](#) &t)
- static void **onlineVersionCheck** ()



## Additional Inherited Members

### Signals inherited from [Digikam::DConfigDlg](#)

- void [currentPageChanged](#) ([DConfigDlgWdgItem](#) \*current, [DConfigDlgWdgItem](#) \*before)
- void [pageRemoved](#) ([DConfigDlgWdgItem](#) \*page)

### Protected Member Functions inherited from [Digikam::DConfigDlg](#)

- QDialogButtonBox \* [buttonBox](#) ()
- const QDialogButtonBox \* [buttonBox](#) () const
- **DConfigDlg** ([DConfigDlgPrivate](#) &dd, [DConfigDlgWdg](#) \*const widget, QWidget \*const parent, Qt::Window↔Flags flags=Qt::WindowFlags())
- [DConfigDlg](#) ([DConfigDlgWdg](#) \*const widget, QWidget \*const parent, Qt::WindowFlags flags=Qt::Window↔Flags())
- [DConfigDlgWdg](#) \* [pageWidget](#) ()
- const [DConfigDlgWdg](#) \* [pageWidget](#) () const
- void [setButtonBox](#) (QDialogButtonBox \*const box)
- void [setPageWidget](#) ([DConfigDlgWdg](#) \*const widget)
- void **showEvent** (QShowEvent \*) override

### Protected Attributes inherited from [Digikam::DConfigDlg](#)

- [DConfigDlgPrivate](#) \*const **d\_ptr** = nullptr

## 6.1294.1 Member Function Documentation

### 6.1294.1.1 [execDialog\(\)](#)

```
bool Digikam::Setup::execDialog (
    Page page = LastPageUsed ) [static]
```

Show a setup dialog. The specified page will be selected. True is returned if the dialog was closed with Ok.

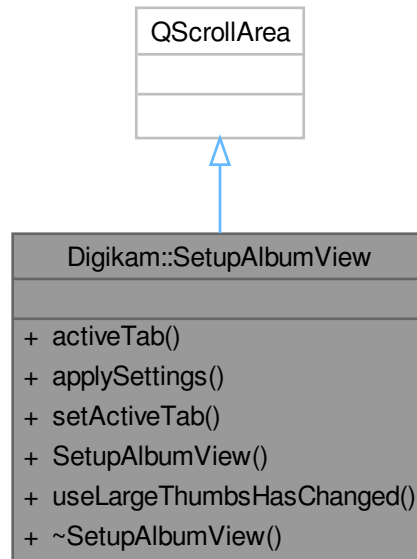
### 6.1294.1.2 [execSinglePage\(\)](#)

```
bool Digikam::Setup::execSinglePage (
    Page page ) [static]
```

Show a setup dialog. Only the specified page will be available.

## 6.1295 Digikam::SetupAlbumView Class Reference

Inheritance diagram for Digikam::SetupAlbumView:



### Public Types

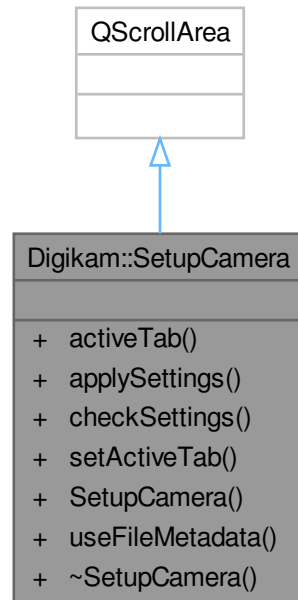
- enum `AlbumTab` {  
    `IconView = 0`, `FolderView`, `Preview`, `FullScreen`,  
    `MimeType`, `Category` }

### Public Member Functions

- `AlbumTab` `activeTab` () const
- void `applySettings` ()
- void `setActiveTab` (AlbumTab tab)
- `SetupAlbumView` (QWidget \*const parent=nullptr)
- bool `useLargeThumbsHasChanged` () const

## 6.1296 Digikam::SetupCamera Class Reference

Inheritance diagram for Digikam::SetupCamera:



### Public Types

- enum **CameraTab** { **Devices** = 0 , **Behavior** , **ImportFilters** , **ImportWindow** }
- enum **ConflictRule** { **OVERWRITE** = 0 , **DIFFNAME** , **SKIPFILE** }

### Signals

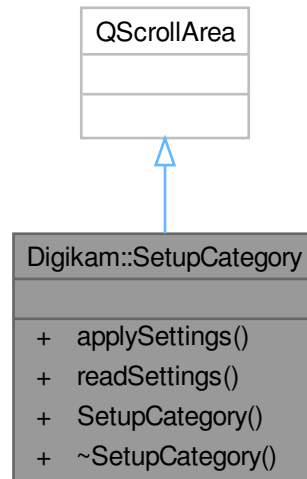
- void **signalUseFileMetadataChanged** (bool)

### Public Member Functions

- CameraTab **activeTab** () const
- void **applySettings** ()
- bool **checkSettings** ()
- void **setActiveTab** (CameraTab tab)
- **SetupCamera** (QWidget \*const parent=nullptr)
- bool **useFileMetadata** ()

## 6.1297 Digikam::SetupCategory Class Reference

Inheritance diagram for Digikam::SetupCategory:

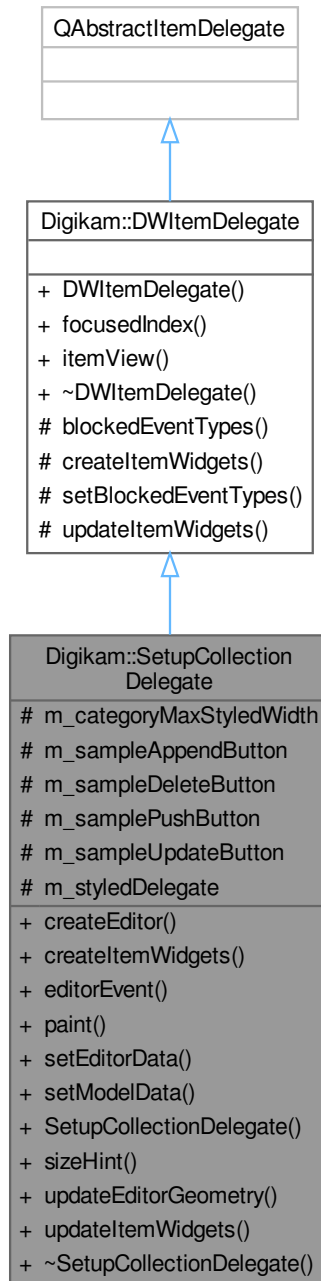


### Public Member Functions

- void **applySettings** ()
- void **readSettings** ()
- **SetupCategory** (QWidget \*const parent=nullptr)

## 6.1298 Digikam::SetupCollectionDelegate Class Reference

Inheritance diagram for Digikam::SetupCollectionDelegate:



### Signals

- void **appendPressed** (int mappedId) const
- void **categoryButtonPressed** (int mappedId) const
- void **deletePressed** (int mappedId) const
- void **updatePressed** (int mappedId) const

## Public Member Functions

- `QWidget * createEditor` (`QWidget *parent`, `const QStyleOptionViewItem &option`, `const QModelIndex &index`) `const` override
- `QList< QWidget * > createItemWidgets` (`const QModelIndex &index`) `const` override
- `bool editorEvent` (`QEvent *event`, `QAbstractItemModel *model`, `const QStyleOptionViewItem &option`, `const QModelIndex &index`) override
- `void paint` (`QPainter *painter`, `const QStyleOptionViewItem &option`, `const QModelIndex &index`) `const` override
- `void setEditorData` (`QWidget *editor`, `const QModelIndex &index`) `const` override
- `void setModelData` (`QWidget *editor`, `QAbstractItemModel *model`, `const QModelIndex &index`) `const` override
- `SetupCollectionDelegate` (`QAbstractItemView *const view`, `QObject *const parent=nullptr`)
- `QSize sizeHint` (`const QStyleOptionViewItem &option`, `const QModelIndex &index`) `const` override
- `void updateEditorGeometry` (`QWidget *editor`, `const QStyleOptionViewItem &option`, `const QModelIndex &index`) `const` override
- `void updateItemWidgets` (`const QList< QWidget * > &widgets`, `const QStyleOptionViewItem &option`, `const QPersistentModelIndex &index`) `const` override

## Public Member Functions inherited from [Digikam::DWItemDelegate](#)

- [DWItemDelegate](#) (`QAbstractItemView *const itemView`, `QObject *const parent=nullptr`)
- `QPersistentModelIndex focusedIndex` () `const`
- `QAbstractItemView * itemView` () `const`

## Protected Attributes

- `int m_categoryMaxStyledWidth` = 0
- `QToolButton * m_sampleAppendButton` = `nullptr`
- `QToolButton * m_sampleDeleteButton` = `nullptr`
- `QPushButton * m_samplePushButton` = `nullptr`
- `QToolButton * m_sampleUpdateButton` = `nullptr`
- `QStyledItemDelegate * m_styledDelegate` = `nullptr`

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DWItemDelegate](#)

- `QList< QEvent::Type > blockedEventTypes` (`QWidget *const widget`) `const`
- `void setBlockedEventTypes` (`QWidget *const widget`, `const QList< QEvent::Type > &types`) `const`

## 6.1298.1 Member Function Documentation

### 6.1298.1.1 createItemWidgets()

```
QList< QWidget * > Digikam::SetupCollectionDelegate::createItemWidgets (
    const QModelIndex & index ) const [override], [virtual]
```

Creates the list of widgets needed for an item.

#### Note

No initialization of the widgets is supposed to happen here. The widgets will be initialized based on needs for a given item.

If you want to connect some widget signals to any slot, you should do it here.

- index the index to create widgets for.

#### Note

If you want to know the index for which you are creating widgets, it is available as a QModelIndex Q\_↔ PROPERTY called "goya:creatingWidgetsForIndex". Ensure to add Q\_DECLARE\_METATYPE(QModelIndex) before your method definition to tell QVariant about QModelIndex.

#### Returns

the list of newly created widgets which will be used to interact with an item.

#### See also

[updateItemWidgets\(\)](#)

Implements [Digikam::DWItemDelegate](#).

### 6.1298.1.2 updateItemWidgets()

```
void Digikam::SetupCollectionDelegate::updateItemWidgets (
    const QList< QWidget * > & widgets,
    const QStyleOptionViewItem & option,
    const QPersistentModelIndex & index ) const [override], [virtual]
```

Updates a list of widgets for its use inside of the delegate (painting or event handling).

#### Note

All the positioning and sizing should be done in item coordinates.

#### Warning

Do not make widget connections in here, since this method will be called very regularly.

**Parameters**

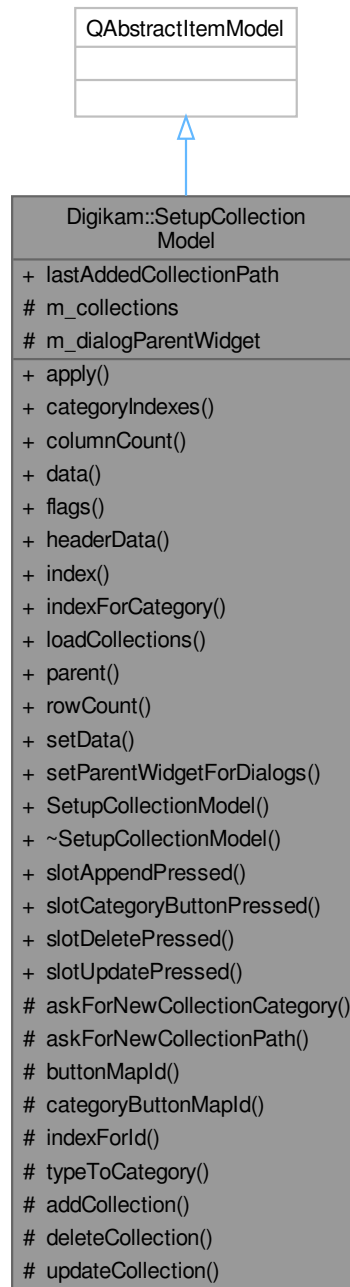
<i>widgets</i>	the widgets to update
<i>option</i>	the current set of style options for the view.
<i>index</i>	the model index of the item currently manipulated.

Implements [Digikam::DWItemDelegate](#).



## 6.1299 Digikam::SetupCollectionModel Class Reference

Inheritance diagram for Digikam::SetupCollectionModel:



### Classes

- class [Item](#)

## Public Types

- enum **Category** { **CategoryLocal** = 0 , **CategoryRemovable** = 1 , **CategoryRemote** = 2 , **NumberOfCategories** }
- enum **Columns** { **ColumnStatus** = 0 , **ColumnName** = 1 , **ColumnPath** = 2 , **ColumnAppendButton** = 3 , **ColumnUpdateButton** = 4 , **ColumnDeleteButton** = 5 , **NumberOfColumns** }
- enum **SetupCollectionDataRole** { **IsCategoryRole** = Qt::UserRole , **CategoryButtonDisplayRole** = Qt::UserRole + 1 , **CategoryButtonMapId** = Qt::UserRole + 2 , **IsAppendRole** = Qt::UserRole + 3 , **AppendDecorationRole** = Qt::UserRole + 4 , **AppendMapId** = Qt::UserRole + 5 , **IsUpdateRole** = Qt::UserRole + 6 , **UpdateDecorationRole** = Qt::UserRole + 7 , **UpdateMapId** = Qt::UserRole + 8 , **IsDeleteRole** = Qt::UserRole + 9 , **DeleteDecorationRole** = Qt::UserRole + 10 , **DeleteMapId** = Qt::UserRole + 11 }

## Public Slots

- void **slotAppendPressed** (int mappedId)
- void **slotCategoryButtonPressed** (int mappedId)
- void **slotDeletePressed** (int mappedId)
- void **slotUpdatePressed** (int mappedId)

## Signals

- void **collectionsLoaded** ()  
*Emitted when all collections were loaded and the model reset in loadCollections.*

## Public Member Functions

- void **apply** ()  
*Apply the changed settings to [CollectionManager](#).*
- QList< QModelIndex > **categoryIndexes** () const
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override  
*QAbstractItemModel implementation.*
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex **indexForCategory** (Category category) const
- void **loadCollections** ()  
*Read collections from [CollectionManager](#).*
- QModelIndex **parent** (const QModelIndex &index) const override
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override
- void **setParentWidgetForDialogs** (QWidget \*const widget)  
*Set a widget used as parent for dialogs and message boxes.*
- [SetupCollectionModel](#) (QObject \*const parent=nullptr)

## Public Attributes

- QString **lastAddedCollectionPath**

### Protected Slots

- void **addCollection** (int category)
- void **deleteCollection** (int internalId)
- void **updateCollection** (int internalId)

### Protected Member Functions

- bool **askForNewCollectionCategory** (int \*const category)
- bool **askForNewCollectionPath** (bool adding, int category, QString \*const newPath, QString \*const newLabel)
- int **buttonMapId** (const QModelIndex &index) const
- int **categoryButtonMapId** (const QModelIndex &index) const
- QModelIndex **indexForId** (int id, int column) const

### Static Protected Member Functions

- static Category **typeToCategory** ([CollectionLocation::Type](#) type)

### Protected Attributes

- QList< [Item](#) > **m\_collections**
- QWidget \* **m\_dialogParentWidget** = nullptr

## 6.1299.1 Member Enumeration Documentation

### 6.1299.1.1 SetupCollectionDataRole

```
enum Digikam::SetupCollectionModel::SetupCollectionDataRole
```

[SetupCollectionModel](#) is a model specialized for use in [SetupCollectionTreeView](#). It provides a reads the current collections from [CollectionManager](#), displays them in three categories, and supports adding and removing collections

#### Enumerator

IsCategoryRole	Returns true if the model index is the index of a category.
CategoryButtonDisplayRole	The text for the category button.
IsAppendRole	Returns true if the model index is the index of a button.
AppendDecorationRole	The pixmap of the button.
IsUpdateRole	Returns true if the model index is the index of a button.
UpdateDecorationRole	The pixmap of the button.
IsDeleteRole	Returns true if the model index is the index of a button.
DeleteDecorationRole	The pixmap of the button.

## 6.1299.2 Constructor & Destructor Documentation

### 6.1299.2.1 SetupCollectionModel()

```
Digikam::SetupCollectionModel::SetupCollectionModel (
    QObject *const parent = nullptr ) [explicit]
```

Internal data structure:

The category entries get a model index with INTERNALID and are identified by their row(). The item entries get the index in m\_collections as INTERNALID. No item is ever removed from m\_collections, deleted entries are only marked as such.

Items have a location, a parentId, and a name and label field. parentId always contains the category, needed to implement parent(). The location is the location if it exists, or is null if the item was added. Name and label are null if unchanged, then the values from location are used. They are valid if edited (label) or the location was added (both valid, location null).

## 6.1299.3 Member Function Documentation

### 6.1299.3.1 slotAppendPressed

```
void Digikam::SetupCollectionModel::slotAppendPressed (
    int mappedId ) [slot]
```

Forward button clicked signals to this slot. mappedId is retrieved with the ButtonMapId role for the model index of the button

### 6.1299.3.2 slotCategoryButtonPressed

```
void Digikam::SetupCollectionModel::slotCategoryButtonPressed (
    int mappedId ) [slot]
```

Forward category button clicked signals to this slot. mappedId is retrieved with the CategoryButtonMapId role for the model index of the button

## 6.1300 Digikam::SetupCollectionModel::Item Class Reference

### Public Member Functions

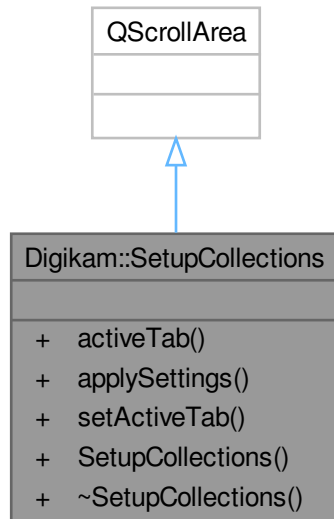
- **Item** (const [CollectionLocation](#) &location)
- **Item** (const QString &path, const QString &label, SetupCollectionModel::Category category)

### Public Attributes

- bool **appended** = false
- QStringList **childs**
- bool **deleted** = false
- QString **label**
- [CollectionLocation](#) **location**
- int **orgIndex** = 0
- int **parentId** = 0
- QString **path**
- bool **updated** = false

## 6.1301 Digikam::SetupCollections Class Reference

Inheritance diagram for Digikam::SetupCollections:



### Public Types

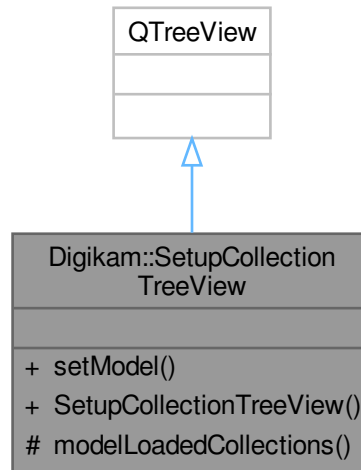
- enum `CollectionsTab` { `Collections = 0` , `IgnoreDirs` }

### Public Member Functions

- `CollectionsTab` `activeTab` () const
- void `applySettings` ()
- void `setActiveTab` (`CollectionsTab` tab)
- `SetupCollections` (`QWidget *const` parent=nullptr)

## 6.1302 Digikam::SetupCollectionTreeView Class Reference

Inheritance diagram for Digikam::SetupCollectionTreeView:



### Public Member Functions

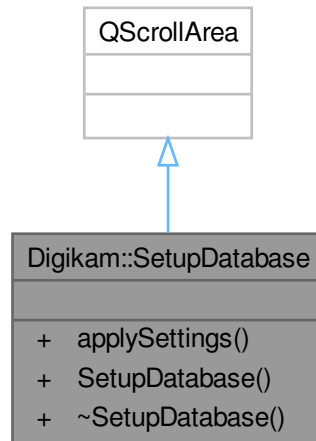
- void **setModel** ([SetupCollectionModel](#) \*model)
- **SetupCollectionTreeView** (QWidget \*const parent=nullptr)

### Protected Slots

- void **modelLoadedCollections** ()

## 6.1303 Digikam::SetupDatabase Class Reference

Inheritance diagram for Digikam::SetupDatabase:

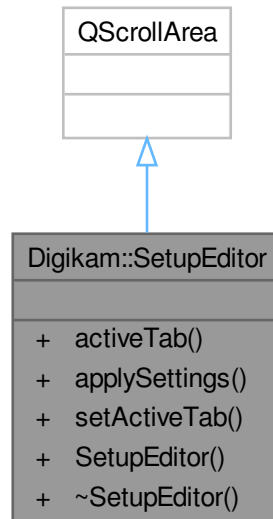


### Public Member Functions

- void **applySettings** ()
- **SetupDatabase** (QWidget \*const parent=nullptr)

## 6.1304 Digikam::SetupEditor Class Reference

Inheritance diagram for Digikam::SetupEditor:



### Public Types

- enum **EditorTab** {  
    **EditorWindow** = 0 , **Versioning** , **SaveSettings** , **RAWBehavior** ,  
    **RAWDefaultSettings** }

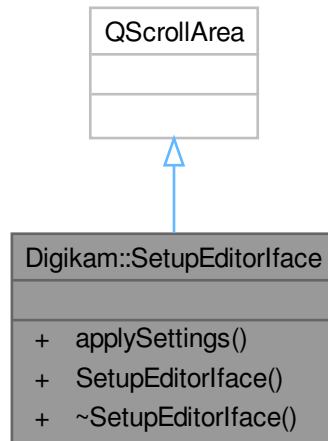
### Public Member Functions

- EditorTab **activeTab** () const
- void **applySettings** ()
- void **setActiveTab** (EditorTab tab)
- **SetupEditor** (QWidget \*const parent=nullptr)



## 6.1305 Digikam::SetupEditorIface Class Reference

Inheritance diagram for Digikam::SetupEditorIface:

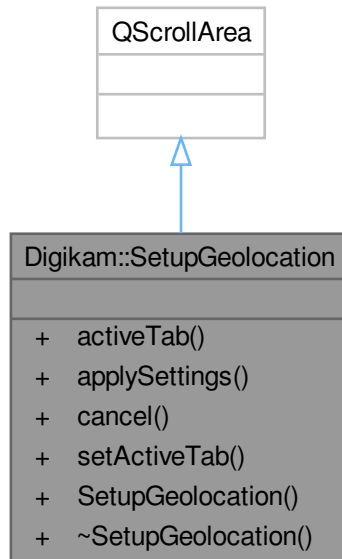


### Public Member Functions

- void **applySettings** ()
- **SetupEditorIface** (QWidget \*const parent=nullptr)

## 6.1306 Digikam::SetupGeolocation Class Reference

Inheritance diagram for Digikam::SetupGeolocation:



### Public Types

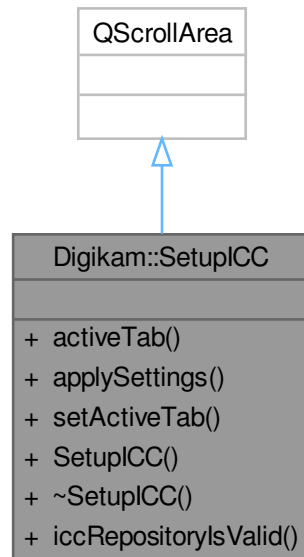
- enum **GeolocationTab** { **MarbleView** = 0 , **MarblePlugins** , **GoogleMaps** }

### Public Member Functions

- GeolocationTab **activeTab** () const
- void **applySettings** ()
- void **cancel** ()
- void **setActiveTab** (GeolocationTab tab)
- **SetupGeolocation** (QWidget \*const parent=nullptr)

## 6.1307 Digikam::SetupICC Class Reference

Inheritance diagram for Digikam::SetupICC:



### Public Types

- enum `ICCTab` { `Behavior` = 0 , `Profiles` , `Advanced` }

### Public Member Functions

- `ICCTab activeTab ()` const
- void `applySettings ()`
- void `setActiveTab (ICCTab tab)`
- `SetupICC (QDialogButtonBox *const dlgBtnBox, QWidget *const parent=nullptr)`

### Static Public Member Functions

- static bool `iccRepositoryIsValid ()`

## 6.1307.1 Constructor & Destructor Documentation

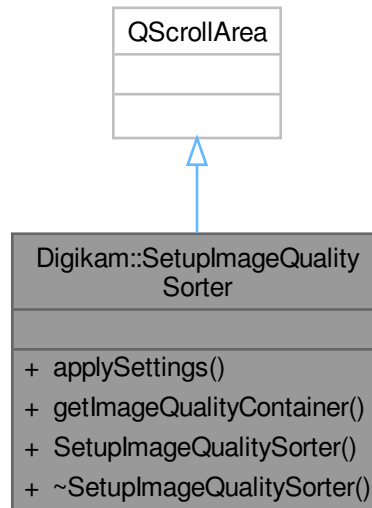
### 6.1307.1.1 SetupICC()

```

Digikam::SetupICC::SetupICC (
    QDialogButtonBox *const dlgBtnBox,
    QWidget *const parent = nullptr ) [explicit]
  
```

## 6.1308 Digikam::SetupImageQualitySorter Class Reference

Inheritance diagram for Digikam::SetupImageQualitySorter:

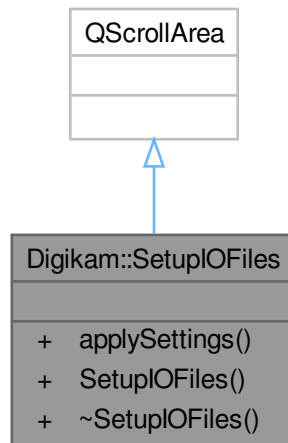


### Public Member Functions

- void **applySettings** ()
- [ImageQualityContainer](#) **getImageQualityContainer** () const
- **SetupImageQualitySorter** (QWidget \*const parent=nullptr)

## 6.1309 Digikam::SetupIOFiles Class Reference

Inheritance diagram for Digikam::SetupIOFiles:

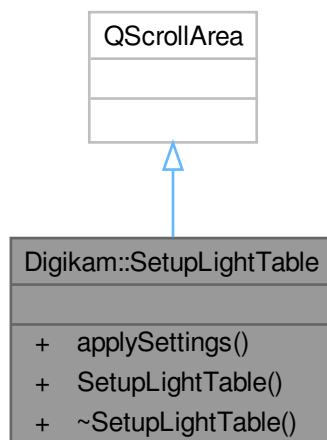


### Public Member Functions

- void **applySettings** ()
- **SetupIOFiles** (QWidget \*const parent=nullptr)

## 6.1310 Digikam::SetupLightTable Class Reference

Inheritance diagram for Digikam::SetupLightTable:

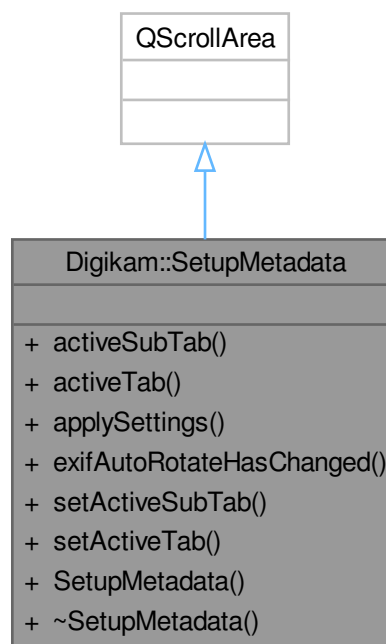


## Public Member Functions

- void **applySettings** ()
- **SetupLightTable** (QWidget \*const parent=nullptr)

## 6.1311 Digikam::SetupMetadata Class Reference

Inheritance diagram for Digikam::SetupMetadata:



## Classes

- class [Private](#)

## Public Types

- enum `MetadataSubTab` {  
`ExifViewer = 0`, `MakernotesViewer`, `IPTCViewer`, `XMPViewer`,  
`ExifToolViewer` }
- enum `MetadataTab` {  
`Behavior = 0`, `Sidecars`, `Rotation`, `Display`,  
`ExifTool`, `Baloo`, `AdvancedConfig` }

**Public Member Functions**

- MetadataSubTab **activeSubTab** () const
- MetadataTab **activeTab** () const
- void **applySettings** ()
- bool **exifAutoRotateHasChanged** () const
- void **setActiveSubTab** (MetadataSubTab tab)
- void **setActiveTab** (MetadataTab tab)
- **SetupMetadata** (QWidget \*const parent=nullptr)

**6.1312 Digikam::SetupMetadata::Private Class Reference****Public Member Functions**

- void **readSettings** ()

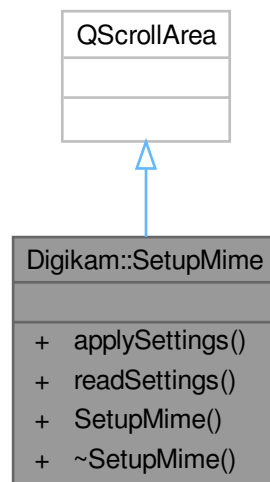
**Public Attributes**

- [AdvancedMetadataTab](#) \* **advTab** = nullptr
- QCheckBox \* **allowLossyRotate** = nullptr
- QCheckBox \* **allowRotateByMetadata** = nullptr
- QTabWidget \* **displaySubTab** = nullptr
- bool **exifAutoRotateOriginal** = false
- bool **exifAutoRotateShowedInfo** = false
- QCheckBox \* **exifRotateBox** = nullptr
- QCheckBox \* **exifSetOrientationBox** = nullptr
- [ExifToolConfPanel](#) \* **exifToolView** = nullptr
- QLineEdit \* **extensionsEdit** = nullptr
- int **extensionsMsgBoxResult** = QMessageBox::Help
- QGroupBox \* **fieldsGroup** = nullptr
- QCheckBox \* **readFromBalooBox** = nullptr
- QCheckBox \* **readWithExifToolBox** = nullptr
- QGroupBox \* **readWriteGroup** = nullptr
- QCheckBox \* **readXMPSidecarBox** = nullptr
- QCheckBox \* **rescanImageIfModifiedBox** = nullptr
- QRadioButton \* **rotateByContents** = nullptr
- QRadioButton \* **rotateByFlag** = nullptr
- QGroupBox \* **rotationAdvGroup** = nullptr
- QGroupBox \* **rotationGroup** = nullptr
- QCheckBox \* **saveColorLabelBox** = nullptr
- QCheckBox \* **saveCommentsBox** = nullptr
- QCheckBox \* **saveDateTimeBox** = nullptr
- QCheckBox \* **saveFaceTags** = nullptr
- QCheckBox \* **savePickLabelBox** = nullptr
- QCheckBox \* **savePosition** = nullptr
- QCheckBox \* **saveRatingBox** = nullptr
- QCheckBox \* **saveTagsBox** = nullptr
- QCheckBox \* **saveTemplateBox** = nullptr
- QCheckBox \* **saveToBalooBox** = nullptr
- QCheckBox \* **sidecarFileNameBox** = nullptr
- bool **sidecarFileNameShowedInfo** = false
- QTabWidget \* **tab** = nullptr

- `MetadataPanel * tagsCfgPanel = nullptr`
- `QCheckBox * updateFileTimeStampBox = nullptr`
- `QCheckBox * useLazySync = nullptr`
- `QCheckBox * writeDngFilesBox = nullptr`
- `QCheckBox * writeRawFilesBox = nullptr`
- `QCheckBox * writeWithExifToolBox = nullptr`
- `QLabel * writeWithExifToolLabel = nullptr`
- `QCheckBox * writeXMPSidecarBox = nullptr`
- `QComboBox * writingModeCombo = nullptr`

## 6.1313 Digikam::SetupMime Class Reference

Inheritance diagram for Digikam::SetupMime:



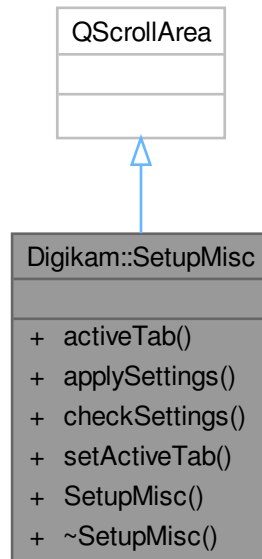
### Public Member Functions

- `void applySettings ()`
- `void readSettings ()`
- `SetupMime (QWidget *const parent=nullptr)`



## 6.1314 Digikam::SetupMisc Class Reference

Inheritance diagram for Digikam::SetupMisc:



### Classes

- class [Private](#)

### Public Types

- enum `MiscTab` {  
    **Behaviour** = 0 , **Appearance** , **Grouping** , **SpellCheck** ,  
    **Localize** , **System** }

### Public Member Functions

- `MiscTab` **activeTab** () const
- void **applySettings** ()
- bool **checkSettings** ()
- void **setActiveTab** (MiscTab tab)
- **SetupMisc** (QWidget \*const parent=nullptr)

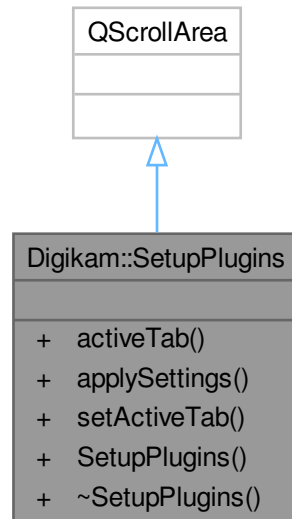
## 6.1315 Digikam::SetupMisc::Private Class Reference

### Public Attributes

- QComboBox \* **albumDateSource** = nullptr
- QLabel \* **albumDateSourceLabel** = nullptr
- [DFontSelect](#) \* **applicationFont** = nullptr
- QComboBox \* **applicationIcon** = nullptr
- QLabel \* **applicationIconLabel** = nullptr
- QComboBox \* **applicationStyle** = nullptr
- QLabel \* **applicationStyleLabel** = nullptr
- QCheckBox \* **cleanAtStart** = nullptr
- QCheckBox \* **detectFaces** = nullptr
- QCheckBox \* **drawFramesToGroupedCheck** = nullptr
- QCheckBox \* **expandNewCurrentItemCheck** = nullptr
- QHash< int, QButtonGroup \* > **groupingButtons** = QHash<int, QButtonGroup\*>()
- [LocalizeConfig](#) \* **localizeWidget** = nullptr
- QSpinBox \* **minimumSimilarityBound** = nullptr
- QLabel \* **minSimilarityBoundLabel** = nullptr
- QCheckBox \* **scanAtStart** = nullptr
- QCheckBox \* **scrollItemToCenterCheck** = nullptr
- QCheckBox \* **selectFirstAlbumItemCheck** = nullptr
- QCheckBox \* **showOnlyPersonTagsInPeopleSidebarCheck** = nullptr
- QCheckBox \* **showPermanentDeleteDialogCheck** = nullptr
- QCheckBox \* **showSplashCheck** = nullptr
- QCheckBox \* **showTrashDeleteDialogCheck** = nullptr
- QCheckBox \* **sidebarApplyDirectlyCheck** = nullptr
- QComboBox \* **sidebarType** = nullptr
- QLabel \* **sidebarTypeLabel** = nullptr
- QComboBox \* **stringComparisonType** = nullptr
- QLabel \* **stringComparisonTypeLabel** = nullptr
- [SystemSettingsWidget](#) \* **systemSettingsWidget** = nullptr
- QTabWidget \* **tab** = nullptr
- QComboBox \* **updateType** = nullptr
- QLabel \* **updateTypeLabel** = nullptr
- QCheckBox \* **updateWithDebug** = nullptr
- QCheckBox \* **useFastScan** = nullptr
- QCheckBox \* **useNativeFileDialogCheck** = nullptr

## 6.1316 Digikam::SetupPlugins Class Reference

Inheritance diagram for Digikam::SetupPlugins:



### Public Types

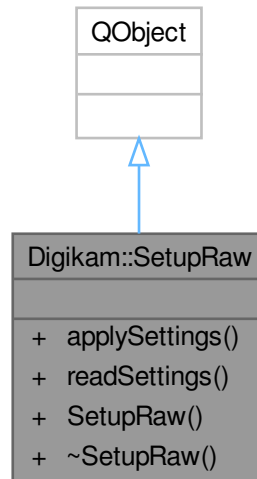
- enum `PluginTab` { `Generic = 0` , `Editor` , `Bqm` , `Loaders` }

### Public Member Functions

- `PluginTab` `activeTab` () const
- void `applySettings` ()
- void `setActiveTab` (`PluginTab` tab)
- `SetupPlugins` (`QWidget *const` parent=nullptr)

## 6.1317 Digikam::SetupRaw Class Reference

Inheritance diagram for Digikam::SetupRaw:

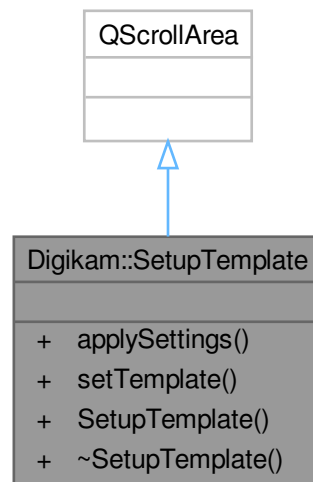


### Public Member Functions

- void **applySettings** ()
- void **readSettings** ()
- **SetupRaw** (QTabWidget \*const tab)

## 6.1318 Digikam::SetupTemplate Class Reference

Inheritance diagram for Digikam::SetupTemplate:

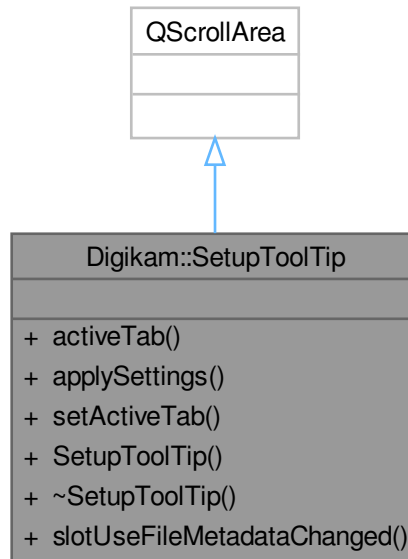


### Public Member Functions

- void **applySettings** ()
- void **setTemplate** (const [Template](#) &t)
- **SetupTemplate** (QWidget \*const parent=nullptr)

## 6.1319 Digikam::SetupToolTip Class Reference

Inheritance diagram for Digikam::SetupToolTip:



### Public Types

- enum `ToolTipTab` { `IconItems = 0` , `AlbumItems` , `ImportItems` }

### Public Slots

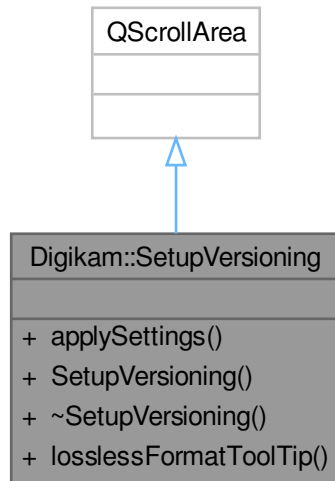
- void `slotUseFileMetadataChanged` (bool)

### Public Member Functions

- `ToolTipTab` `activeTab` () const
- void `applySettings` ()
- void `setActiveTab` (`ToolTipTab` tab)
- `SetupToolTip` (`QWidget *const` parent=nullptr)

## 6.1320 Digikam::SetupVersioning Class Reference

Inheritance diagram for Digikam::SetupVersioning:



### Public Member Functions

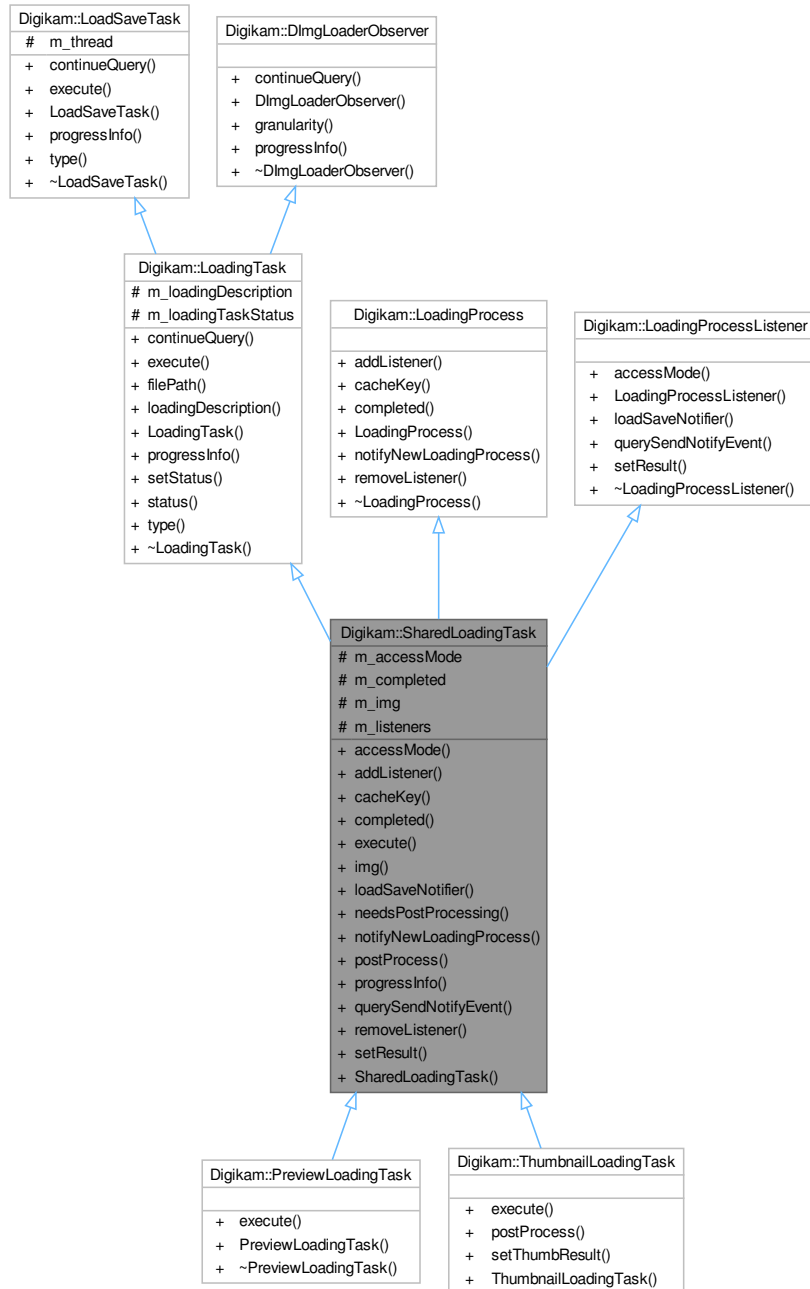
- void **applySettings** ()
- **SetupVersioning** (QWidget \*const parent=nullptr)

### Static Public Member Functions

- static void **losslessFormatToolTip** (QString &formatHelp, bool hasJXLSupport, bool hasWEBPSupport, bool hasAVIFSupport)

## 6.1321 Digikam::SharedLoadingTask Class Reference

Inheritance diagram for Digikam::SharedLoadingTask:



### Public Member Functions

- `LoadSaveThread::AccessMode accessMode ()` const override
- `void addListener (LoadingProcessListener *const listener)` override
- `QString cacheKey ()` const override
- `bool completed ()` const override



- void `execute` () override
- `DImg img` () const
- `LoadSaveNotifier * loadSaveNotifier` () const override
- bool `needsPostProcessing` () const
- void `notifyNewLoadingProcess` (`LoadingProcess *const process`, const `LoadingDescription &description`) override
- virtual void `postProcess` ()
- void `progressInfo` (float progress) override
- bool `querySendNotifyEvent` () const override
- void `removeListener` (`LoadingProcessListener *const listener`) override
- void `setResult` (const `LoadingDescription &loadingDescription`, const `DImg &img`) override
- `SharedLoadingTask` (`LoadSaveThread *const thread`, const `LoadingDescription &description`, `LoadSaveThread::AccessMode mode=LoadSaveThread::AccessModeReadWrite`, `LoadingTaskStatus loadingTaskStatus=LoadingTask←→ StatusLoading`)

### Public Member Functions inherited from [Digikam::LoadingTask](#)

- bool `continueQuery` () override
- `QString filePath` () const
- const `LoadingDescription & loadingDescription` () const
- `LoadingTask` (`LoadSaveThread *const thread`, const `LoadingDescription &description`, `LoadingTaskStatus loadingTaskStatus=LoadingTaskStatusLoading`)
- void `setStatus` (`LoadingTaskStatus status`)
- `LoadingTaskStatus status` () const
- `TaskType type` () override

### Public Member Functions inherited from [Digikam::LoadSaveTask](#)

- `LoadSaveTask` (`LoadSaveThread *const thread`)

### Public Member Functions inherited from [Digikam::DImgLoaderObserver](#)

- virtual float `granularity` ()

### Protected Attributes

- `LoadSaveThread::AccessMode m_accessMode` = `LoadSaveThread::AccessModeReadWrite`
- volatile bool `m_completed` = false
- `DImg m_img`
- `QList< LoadingProcessListener * > m_listeners`

### Protected Attributes inherited from [Digikam::LoadingTask](#)

- `LoadingDescription m_loadingDescription`
- volatile `LoadingTaskStatus m_loadingTaskStatus` = `LoadingTaskStatusLoading`

### Protected Attributes inherited from [Digikam::LoadSaveTask](#)

- `LoadSaveThread * m_thread` = nullptr

## Additional Inherited Members

### Public Types inherited from [Digikam::LoadingTask](#)

- enum **LoadingTaskStatus** { **LoadingTaskStatusLoading** , **LoadingTaskStatusPreloading** , **LoadingTaskStatusStopping** }

### Public Types inherited from [Digikam::LoadSaveTask](#)

- enum **TaskType** { **TaskTypeLoading** , **TaskTypeSaving** }

## 6.1321.1 Member Function Documentation

### 6.1321.1.1 `accessMode()`

```
LoadSaveThread::AccessMode Digikam::SharedLoadingTask::accessMode ( ) const [override], [virtual]
```

Implements [Digikam::LoadingProcessListener](#).

### 6.1321.1.2 `addListener()`

```
void Digikam::SharedLoadingTask::addListener (
    LoadingProcessListener *const listener ) [override], [virtual]
```

Implements [Digikam::LoadingProcess](#).

### 6.1321.1.3 `cacheKey()`

```
QString Digikam::SharedLoadingTask::cacheKey ( ) const [override], [virtual]
```

Implements [Digikam::LoadingProcess](#).

### 6.1321.1.4 `completed()`

```
bool Digikam::SharedLoadingTask::completed ( ) const [override], [virtual]
```

Implements [Digikam::LoadingProcess](#).

### 6.1321.1.5 `execute()`

```
void Digikam::SharedLoadingTask::execute ( ) [override], [virtual]
```

Reimplemented from [Digikam::LoadingTask](#).

### 6.1321.1.6 loadSaveNotifier()

```
LoadSaveNotifier * Digikam::SharedLoadingTask::loadSaveNotifier ( ) const [override], [virtual]
```

Implements [Digikam::LoadingProcessListener](#).

### 6.1321.1.7 notifyNewLoadingProcess()

```
void Digikam::SharedLoadingTask::notifyNewLoadingProcess (
    LoadingProcess *const process,
    const LoadingDescription & description ) [override], [virtual]
```

Implements [Digikam::LoadingProcess](#).

### 6.1321.1.8 progressInfo()

```
void Digikam::SharedLoadingTask::progressInfo (
    float progress ) [override], [virtual]
```

Reimplemented from [Digikam::LoadingTask](#).

### 6.1321.1.9 querySendNotifyEvent()

```
bool Digikam::SharedLoadingTask::querySendNotifyEvent ( ) const [override], [virtual]
```

Implements [Digikam::LoadingProcessListener](#).

### 6.1321.1.10 removeListener()

```
void Digikam::SharedLoadingTask::removeListener (
    LoadingProcessListener *const listener ) [override], [virtual]
```

Implements [Digikam::LoadingProcess](#).

### 6.1321.1.11 setResult()

```
void Digikam::SharedLoadingTask::setResult (
    const LoadingDescription & loadingDescription,
    const DImg & img ) [override], [virtual]
```

Implements [Digikam::LoadingProcessListener](#).

## 6.1322 Digikam::SharedLoadSaveThread Class Reference

Inheritance diagram for Digikam::SharedLoadSaveThread:



### Public Member Functions

- void **load** (const [LoadingDescription](#) &description, [AccessMode](#) mode, [LoadingPolicy](#) policy=[LoadingPolicyAppend](#))
- **SharedLoadSaveThread** ([QObject](#) \*const parent=nullptr)

## Public Member Functions inherited from Digikam::ManagedLoadSaveThread

- void **load** (const [LoadingDescription](#) &description)
- void **load** (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- [LoadingPolicy](#) **loadingPolicy** () const
- [ManagedLoadSaveThread](#) (QObject \*const parent=nullptr)
- void **save** (const [DImg](#) &image, const QString &filePath, const QString &format)
- void **setLoadingPolicy** ([LoadingPolicy](#) policy)
- void **setTerminationPolicy** ([TerminationPolicy](#) terminationPolicy)
- void **stopAllTasks** ()
- void **stopLoading** (const [LoadingDescription](#) &desc, [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void **stopLoading** (const QString &filePath=QString(), [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void **stopSaving** (const QString &filePath=QString())
- [TerminationPolicy](#) **terminationPolicy** () const

## Public Member Functions inherited from Digikam::LoadSaveThread

- void **imageLoaded** (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img) override
- void **imageSaved** (const QString &filePath, bool success) override
- void **imageStartedLoading** (const [LoadingDescription](#) &loadingDescription) override
- void **imageStartedSaving** (const QString &filePath) override
- void **load** (const [LoadingDescription](#) &description)
- void **loadingProgress** (const [LoadingDescription](#) &loadingDescription, float progress) override
- [LoadSaveThread](#) (QObject \*const parent=nullptr)
- void **moreCompleteLoadingAvailable** (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription) override
- virtual bool **querySendNotifyEvent** () const
- void **save** (const [DImg](#) &image, const QString &filePath, const QString &format)
- void **savingProgress** (const QString &filePath, float progress) override
- void **setNotificationPolicy** ([NotificationPolicy](#) notificationPolicy)
- virtual void **taskHasFinished** ()
- void **thumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QImage &img) override
- [~LoadSaveThread](#) () override

## Public Member Functions inherited from Digikam::DynamicThread

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Additional Inherited Members

## Public Types inherited from Digikam::ManagedLoadSaveThread

- enum [LoadingMode](#) { [LoadingModeNormal](#) , [LoadingModeShared](#) }
- enum [LoadingPolicy](#) { [LoadingPolicyFirstRemovePrevious](#) , [LoadingPolicyPrepend](#) , [LoadingPolicySimplePrepend](#) , [LoadingPolicyAppend](#) , [LoadingPolicySimpleAppend](#) , [LoadingPolicyPreload](#) }
- enum [LoadingTaskFilter](#) { [LoadingTaskFilterAll](#) , [LoadingTaskFilterPreloading](#) }
- enum [TerminationPolicy](#) { [TerminationPolicyTerminateLoading](#) , [TerminationPolicyTerminatePreloading](#) , [TerminationPolicyWait](#) , [TerminationPolicyTerminateAll](#) }

## Public Types inherited from [Digikam::LoadSaveThread](#)

- enum [AccessMode](#) { [AccessModeRead](#) , [AccessModeReadWrite](#) }
- enum [NotificationPolicy](#) { [NotificationPolicyDirect](#) , [NotificationPolicyTimeLimited](#) }

## Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void [start](#) ()
- void [stop](#) ()
- void [wait](#) ()

## Signals inherited from [Digikam::LoadSaveThread](#)

- void [signalImageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)
- void [signalImageSaved](#) (const QString &filePath, bool success)
- void [signalImageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription)
- void [signalImageStartedSaving](#) (const QString &filePath)
- void [signalLoadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress)
- void [signalMoreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription)
- void [signalSavingProgress](#) (const QString &filePath, float progress)
- void [signalThumbnailLoaded](#) (const [LoadingDescription](#) &loadingDescription, const QImage &img)

## Signals inherited from [Digikam::DynamicThread](#)

- void [finished](#) ()
- void [starting](#) ()

## Static Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- static int [exifOrientation](#) (const QString &filePath, const [DMetadata](#) &metadata, bool isRaw, bool fromRaw↔EmbeddedPreview)
- static [LoadSaveFileInfoProvider](#) \* [infoProvider](#) ()
- static void [setInfoProvider](#) ([LoadSaveFileInfoProvider](#) \*const infoProvider)

## Protected Member Functions inherited from [Digikam::ManagedLoadSaveThread](#)

- void [load](#) (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void [load](#) (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [LoadingPolicy](#) policy, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void [loadPreview](#) (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- void [loadThumbnail](#) (const [LoadingDescription](#) &description)
- void [preloadThumbnail](#) (const [LoadingDescription](#) &description)
- void [preloadThumbnailGroup](#) (const QList< [LoadingDescription](#) > &descriptions)
- void [prependThumbnailGroup](#) (const QList< [LoadingDescription](#) > &descriptions)
- void [shutDown](#) ()

**Protected Member Functions inherited from [Digikam::LoadSaveThread](#)**

- void **notificationReceived** ()
- void **run** () override

**Protected Member Functions inherited from [Digikam::DynamicThread](#)**

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

**Protected Attributes inherited from [Digikam::ManagedLoadSaveThread](#)**

- [LoadingPolicy](#) **m\_loadingPolicy** = [LoadingPolicyAppend](#)
- [TerminationPolicy](#) **m\_terminationPolicy** = [TerminationPolicyTerminateLoading](#)

**Protected Attributes inherited from [Digikam::LoadSaveThread](#)**

- [LoadSaveTask](#) \* **m\_currentTask** = nullptr
- QMutex **m\_mutex**
- [NotificationPolicy](#) **m\_notificationPolicy** = [NotificationPolicyTimeLimited](#)
- QList< [LoadSaveTask](#) \* > **m\_todo**

**6.1323 Digikam::SharedQueue< T > Class Template Reference****Public Member Functions**

- void **clear** ()
- bool **empty** ()
- T & **front** ()
- int **maxDepth** () const
- T & **pop\_front** ()
- void **push\_back** (T &&item)
- void **push\_back** (T &item)
- void **setMaxDepth** (int depth)
- int **size** ()

**6.1324 Digikam::SharpContainer Class Reference****Public Types**

- enum **SharpingMethods** { **SimpleSharp** = 0 , **UnsharpMask** , **Refocus** }

**Public Attributes**

- int **method** = SimpleSharp  
*Store SharpingMethods value.*
- double **rfCorrelation** = 0.5
- double **rfGauss** = 0.0
- int **rfMatrix** = 5
- double **rfNoise** = 0.03
- double **rfRadius** = 1.0  
*Refocus.*
- int **ssRadius** = 0  
*Simple sharp.*
- double **umAmount** = 1.0
- bool **umLumaOnly** = false
- double **umRadius** = 1.0  
*Unsharp mask.*
- double **umThreshold** = 0.05



## 6.1325 Digikam::SharpenFilter Class Reference

Inheritance diagram for Digikam::SharpenFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- void [readParameters](#) (const [FilterAction](#) &action) override
- **SharpenFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, double radius=0.0, double sigma=1.0)
- [SharpenFilter](#) ([DImgThreadedFilter](#) \*const parentFilter, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, double radius=0.0, double sigma=1.0)
- **SharpenFilter** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

## Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

## Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.1325.1 Constructor & Destructor Documentation

### 6.1325.1.1 SharpenFilter()

```
Digikam::SharpenFilter::SharpenFilter (
    DImgThreadedFilter *const parentFilter,
    const DImg & orgImage,
    const DImg & destImage,
    int progressBegin = 0,
    int progressEnd = 100,
    double radius = 0.0,
    double sigma = 1.0 )
```

Constructor for slave mode: execute immediately in current thread with specified master filter

## 6.1325.2 Member Function Documentation

### 6.1325.2.1 filterAction()

```
FilterAction Digikam::SharpenFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1325.2.2 filterIdentifier()

```
QString Digikam::SharpenFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

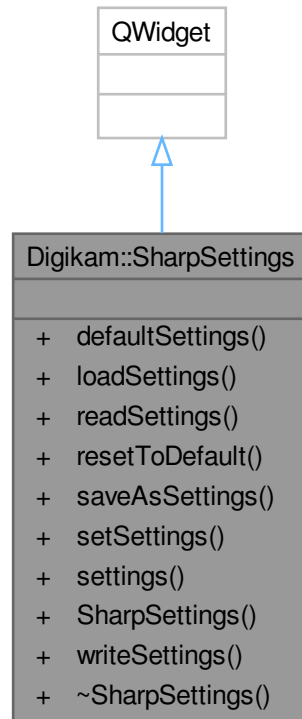
### 6.1325.2.3 readParameters()

```
void Digikam::SharpenFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1326 Digikam::SharpSettings Class Reference

Inheritance diagram for Digikam::SharpSettings:



### Signals

- void `signalSettingsChanged()`

### Public Member Functions

- [SharpContainer](#) `defaultSettings()` const
- void `loadSettings()`
- void `readSettings()` (const `KConfigGroup` &group)
- void `resetToDefault()`
- void `saveAsSettings()`
- void `setSettings()` (const [SharpContainer](#) &settings)
- [SharpContainer](#) `settings()` const
- `SharpSettings` (`QWidget` \*const parent)
- void `writeSettings()` (`KConfigGroup` &group)

## 6.1327 Digikam::ShearFilter Class Reference

Inheritance diagram for Digikam::ShearFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- QSize **getNewSize** () const
- void **readParameters** (const [FilterAction](#) &action) override
- **ShearFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, float hAngle=0.0, float vAngle=0.0, bool antialiasing=true, const QColor &background-color=Qt::black, int orgW=0, int orgH=0)
- **ShearFilter** (QObject \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cancelFilter** ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > **multithreadedSteps** (int stop, int start=0) const
- virtual bool **parametersSuccessfullyRead** () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void **setFilterVersion** (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void **setupAndStartDirectly** (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void **setupFilter** (const [DImg](#) &orgImage)
- virtual void **startFilter** ()
- virtual void **startFilterDirectly** ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false



## 6.1327.1 Member Function Documentation

### 6.1327.1.1 filterAction()

```
FilterAction Digikam::ShearFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1327.1.2 filterIdentifier()

```
QString Digikam::ShearFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1327.1.3 readParameters()

```
void Digikam::ShearFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1328 Digikam::ShowHideVersionsOverlay Class Reference

Inheritance diagram for Digikam::ShowHideVersionsOverlay:



### Public Member Functions

- void [setActive](#) (bool active) override
- void [setSettings](#) (const [VersionManagerSettings](#) &settings)
- [ShowHideVersionsOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- [ItemViewHoverButton](#) \* **button** () const
- [HoverButtonDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotClicked** (bool checked)

## Protected Slots inherited from [Digikam::HoverButtonDelegateOverlay](#)

- void **slotEntered** (const QModelIndex &index) override
- void **slotReset** () override

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotEntered** (const QModelIndex &index)
- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

## Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- [ItemViewHoverButton](#) \* **createButton** () override
- void **updateButton** (const QModelIndex &index) override

### Protected Member Functions inherited from [Digikam::HoverButtonDelegateOverlay](#)

- `QWidget * createWidget ()` override
- `void visualChange ()` override

### Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- `bool checkIndexOnEnter (const QModelIndex &index) const`
- `bool eventFilter (QObject *obj, QEvent *event)` override
- `virtual void hide ()`
- `virtual QString notifyMultipleMessage (const QModelIndex &, int number)`
- `QWidget * parentWidget () const`
- `virtual void viewportLeaveEvent (QObject *obj, QEvent *event)`
- `virtual void widgetEnterEvent ()`
- `void widgetEnterNotifyMultiple (const QModelIndex &index)`
- `virtual void widgetLeaveEvent ()`
- `void widgetLeaveNotifyMultiple ()`

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- `QList< QModelIndex > affectedIndexes (const QModelIndex &index) const`
- `bool affectsMultiple (const QModelIndex &index) const`
- `int numberOfAffectedIndexes (const QModelIndex &index) const`
- `bool viewHasMultiSelection () const`

### Protected Attributes

- `VersionItemFilterSettings m_filter`

### Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- `bool m_mouseButtonPressedOnWidget = false`
- `QWidget * m_widget = nullptr`

### Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- `QAbstractItemDelegate * m_delegate = nullptr`
- `QAbstractItemView * m_view = nullptr`

### Additional Inherited Members

### Signals inherited from [Digikam::ItemDelegateOverlay](#)

- `void hideNotification ()`
- `void requestNotification (const QModelIndex &index, const QString &message)`
- `void update (const QModelIndex &index)`

## 6.1328.1 Member Function Documentation

### 6.1328.1.1 checkIndex()

```
bool Digikam::ShowHideVersionsOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1328.1.2 createButton()

```
ItemViewHoverButton * Digikam::ShowHideVersionsOverlay::createButton ( ) [override], [protected],
[virtual]
```

Create your widget here. Pass view() as parent.

Implements [Digikam::HoverButtonDelegateOverlay](#).

### 6.1328.1.3 setActive()

```
void Digikam::ShowHideVersionsOverlay::setActive (
    bool active ) [override], [virtual]
```

Will call [createButton\(\)](#).

Reimplemented from [Digikam::HoverButtonDelegateOverlay](#).

### 6.1328.1.4 updateButton()

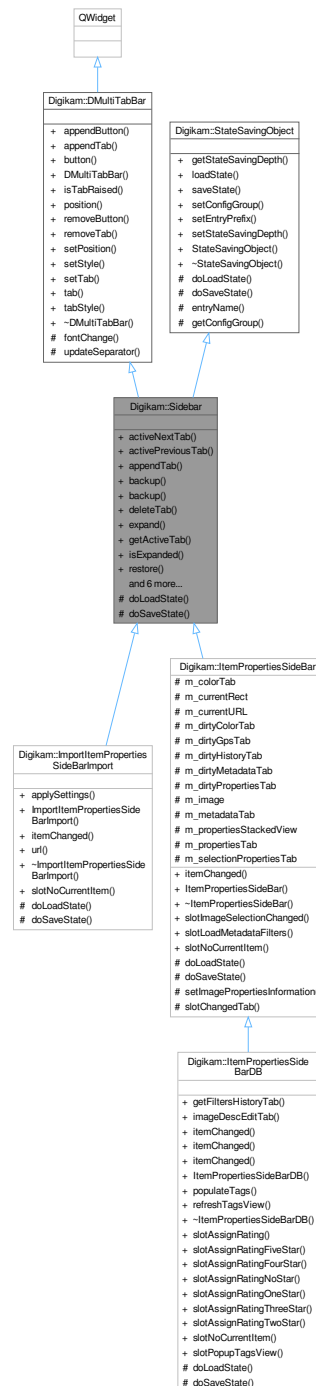
```
void Digikam::ShowHideVersionsOverlay::updateButton (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Called when a new index is entered. Reposition your button here, adjust and store state.

Implements [Digikam::HoverButtonDelegateOverlay](#).

## 6.1329 Digikam::Sidebar Class Reference

Inheritance diagram for Digikam::Sidebar:



### Classes

- class [Private](#)

## Signals

- void [signalChangedTab](#) (QWidget \*w)
- void [signalViewChanged](#) ()

## Public Member Functions

- void [activeNextTab](#) ()
- void [activePreviousTab](#) ()
- void [appendTab](#) (QWidget \*const w, const QIcon &pic, const QString &title)
- void [backup](#) ()
- void [backup](#) (const QList< QWidget \* > &thirdWidgetsToBackup, QList< int > \*const sizes)
- void [deleteTab](#) (QWidget \*const w)
- void [expand](#) ()
- QWidget \* [getActiveTab](#) () const
- bool [isExpanded](#) () const
- void [restore](#) ()
- void [restore](#) (const QList< QWidget \* > &thirdWidgetsToRestore, const QList< int > &sizes)
- void [setActiveTab](#) (QWidget \*const w)
- void [shrink](#) ()
- [Sidebar](#) (QWidget \*const parent, [SidebarSplitter](#) \*const sp, Qt::Edge side=Qt::LeftEdge, bool minimized↔ Default=false)
- [SidebarSplitter](#) \* [splitter](#) () const

## Public Member Functions inherited from [Digikam::DMultiTabBar](#)

- void [appendButton](#) (const QIcon &pic, int id=-1, QMenu \*const popup=nullptr, const QString &not\_used\_↔ yet=QString())
- void [appendTab](#) (const QIcon &pic, int id=-1, const QString &text=QString())
- [DMultiTabBarButton](#) \* [button](#) (int id) const
- [DMultiTabBar](#) (Qt::Edge pos, QWidget \*const parent=nullptr)
- bool [isTabRaised](#) (int id) const
- Qt::Edge [position](#) () const
- void [removeButton](#) (int id)
- void [removeTab](#) (int id)
- void [setPosition](#) (Qt::Edge pos)
- void [setStyle](#) (TextStyle style)
- void [setTab](#) (int id, bool state)
- [DMultiTabBarTab](#) \* [tab](#) (int id) const
- TextStyle [tabStyle](#) () const

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

**Protected Member Functions**

- void [doLoadState](#) () override
- void [doSaveState](#) () override

**Protected Member Functions inherited from [Digikam::DMultiTabBar](#)**

- virtual void **fontChange** (const QFont &)
- void **updateSeparator** ()

**Protected Member Functions inherited from [Digikam::StateSavingObject](#)**

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

**Friends**

- class **SidebarSplitter**

**Additional Inherited Members****Public Types inherited from [Digikam::DMultiTabBar](#)**

- enum [TextStyle](#) { [ActiveIconText](#) = 0 , [AllIconsText](#) = 2 }

**Public Types inherited from [Digikam::StateSavingObject](#)**

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

**6.1329.1 Detailed Description**

This class handles a sidebar view

Since this class derives from [StateSavingObject](#), you can call [StateSavingObject::loadState\(\)](#) and [StateSavingObject::saveState\(\)](#) for loading/saving of settings. However, if you use multiple sidebar instances in your program, you have to remember to either call [QObject::setObjectName\(\)](#), [StateSavingObject::setEntryPrefix\(\)](#) or [StateSavingObject::setConfigGroup\(\)](#) first.

**6.1329.2 Constructor & Destructor Documentation****6.1329.2.1 Sidebar()**

```
Digikam::Sidebar::Sidebar (
    QWidget *const parent,
    SidebarSplitter *const sp,
    Qt::Edge side = Qt::LeftEdge,
    bool minimizedDefault = false ) [explicit]
```

Creates a new sidebar



## Parameters

<i>parent</i>	sidebar's parent
<i>sp</i>	sets the splitter, which should handle the width. The splitter normally is part of the main view. Internally, the width of the widget stack can be changed by a QSplitter.
<i>side</i>	where the sidebar should be displayed. At the left or right border. Use Qt::LeftEdge or Qt::RightEdge.
<i>minimizedDefault</i>	hide the sidebar when the program is started the first time.

### 6.1329.3 Member Function Documentation

#### 6.1329.3.1 activeNextTab()

```
void Digikam::Sidebar::activeNextTab ( )
```

Activates a next tab from current one. If current one is last, first one is activated.

#### 6.1329.3.2 activePreviousTab()

```
void Digikam::Sidebar::activePreviousTab ( )
```

Activates a previous tab from current one. If current one is first, last one is activated.

#### 6.1329.3.3 appendTab()

```
void Digikam::Sidebar::appendTab (
    QWidget *const w,
    const QIcon & pic,
    const QString & title )
```

Appends a new tab to the sidebar

## Parameters

<i>w</i>	widget which is activated by this tab
<i>pic</i>	icon which is shown in this tab
<i>title</i>	text which is shown it this tab

#### 6.1329.3.4 backup() [1/2]

```
void Digikam::Sidebar::backup ( )
```

Hide sidebar and backup minimized state.

### 6.1329.3.5 backup() [2/2]

```
void Digikam::Sidebar::backup (
    const QList< QWidget * > & thirdWidgetsToBackup,
    QList< int > *const sizes )
```

Hide sidebar and backup minimized state. If there are other widgets in this splitter, stores their sizes in the provided list.

### 6.1329.3.6 deleteTab()

```
void Digikam::Sidebar::deleteTab (
    QWidget *const w )
```

Deletes a tab from the tabbar

### 6.1329.3.7 doLoadState()

```
void Digikam::Sidebar::doLoadState ( ) [override], [protected], [virtual]
```

Load the last view state from disk - called by [StateSavingObject::loadState\(\)](#)

Implements [Digikam::StateSavingObject](#).

### 6.1329.3.8 doSaveState()

```
void Digikam::Sidebar::doSaveState ( ) [override], [protected], [virtual]
```

Save the view state to disk - called by [StateSavingObject::saveState\(\)](#)

Implements [Digikam::StateSavingObject](#).

### 6.1329.3.9 expand()

```
void Digikam::Sidebar::expand ( )
```

Redisplays the whole sidebar

### 6.1329.3.10 getActiveTab()

```
QWidget * Digikam::Sidebar::getActiveTab ( ) const
```

Returns the currently activated tab, or 0 if no tab is active

### 6.1329.3.11 isExpanded()

```
bool Digikam::Sidebar::isExpanded ( ) const
```

Return the visible status of current sidebar tab.

**6.1329.3.12 restore()** [1/2]

```
void Digikam::Sidebar::restore ( )
```

Show sidebar and restore minimized state.

**6.1329.3.13 restore()** [2/2]

```
void Digikam::Sidebar::restore (
    const QList< QWidget * > & thirdWidgetsToRestore,
    const QList< int > & sizes )
```

Show sidebar and restore minimized state. Restores other widgets' sizes in splitter.

**6.1329.3.14 setActiveTab()**

```
void Digikam::Sidebar::setActiveTab (
    QWidget *const w )
```

Activates a tab

**6.1329.3.15 shrink()**

```
void Digikam::Sidebar::shrink ( )
```

Hides the sidebar (display only the activation buttons)

**6.1329.3.16 signalChangedTab**

```
void Digikam::Sidebar::signalChangedTab (
    QWidget * w ) [signal]
```

Is emitted, when another tab is activated

**6.1329.3.17 signalViewChanged**

```
void Digikam::Sidebar::signalViewChanged ( ) [signal]
```

Is emitted, when tab is shrink or expanded

## 6.1330 Digikam::Sidebar::Private Class Reference

### Public Attributes

- int **activeTab** = -1
- QHash< QWidget \*, [SidebarState](#) > **appendedTabsStateCache**
- int **dragSwitchId** = -1
- QTimer \* **dragSwitchTimer** = nullptr
- bool **isMinimized** = false
- bool **minimized** = false
- bool **minimizedDefault** = false
- const QString **optionActiveTabEntry** = QLatin1String("ActiveTab")
- const QString **optionMinimizedEntry** = QLatin1String("Minimized")
- const QString **optionRestoreSizeEntry** = QLatin1String("RestoreSize")
- int **restoreSize** = 0
- [SidebarSplitter](#) \* **splitter** = nullptr
- QStackedWidget \* **stack** = nullptr
- int **tabs** = 0

### 6.1330.1 Member Data Documentation

#### 6.1330.1.1 isMinimized

```
bool Digikam::Sidebar::Private::isMinimized = false
```

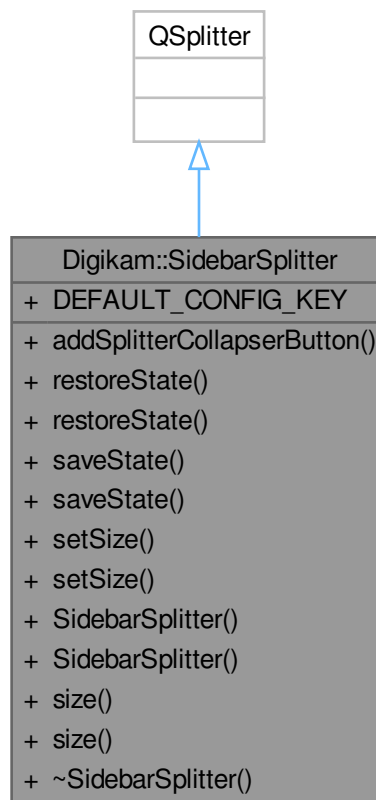
Backup of shrinked status before [backup\(\)](#), restored by [restore\(\)](#)

**Note**

when sidebar is hidden, only icon bar is affected. If sidebar view is visible, this one must be shrink and restored accordingly.

**6.1331 Digikam::SidebarSplitter Class Reference**

Inheritance diagram for Digikam::SidebarSplitter:

**Classes**

- class [Private](#)

**Public Member Functions**

- void **addSplitterCollapserButton** (QWidget \*const widget)
- void [restoreState](#) (KConfigGroup &group)
- void [restoreState](#) (KConfigGroup &group, const QString &key)
- void [saveState](#) (KConfigGroup &group)
- void [saveState](#) (KConfigGroup &group, const QString &key)

- void **setSize** (QWidget \*const widget, int [size](#))
- void [setSize](#) ([Sidebar](#) \*const bar, int [size](#))
- **SidebarSplitter** (Qt::Orientation orientation, QWidget \*const parent=nullptr)
- [SidebarSplitter](#) (QWidget \*const parent=nullptr)
- int [size](#) (QWidget \*const widget) const
- int [size](#) ([Sidebar](#) \*const bar) const

### Static Public Attributes

- static const QString **DEFAULT\_CONFIG\_KEY** = QLatin1String("SplitterState")

### Friends

- class [Sidebar](#)

## 6.1331.1 Constructor & Destructor Documentation

### 6.1331.1.1 SidebarSplitter()

```
Digikam::SidebarSplitter::SidebarSplitter (
    QWidget *const parent = nullptr ) [explicit]
```

This is a QSplitter with better support for storing its state in config files, especially if Sidebars are contained in the splitter.

## 6.1331.2 Member Function Documentation

### 6.1331.2.1 restoreState() [1/2]

```
void Digikam::SidebarSplitter::restoreState (
    KConfigGroup & group )
```

Restores the splitter state from group, handling minimized sidebars correctly. DEFAULT\_CONFIG\_KEY is used for restoring the state.

### 6.1331.2.2 restoreState() [2/2]

```
void Digikam::SidebarSplitter::restoreState (
    KConfigGroup & group,
    const QString & key )
```

Restores the splitter state from group, handling minimized sidebars correctly. This version uses a specified key in the config group.

### 6.1331.2.3 saveState() [1/2]

```
void Digikam::SidebarSplitter::saveState (
    KConfigGroup & group )
```

Saves the splitter state to group, handling minimized sidebars correctly. DEFAULT\_CONFIG\_KEY is used for storing the state.

### 6.1331.2.4 saveState() [2/2]

```
void Digikam::SidebarSplitter::saveState (
    KConfigGroup & group,
    const QString & key )
```

Saves the splitter state to group, handling minimized sidebars correctly. This version uses a specified key in the config group.

### 6.1331.2.5 setSize()

```
void Digikam::SidebarSplitter::setSize (
    Sidebar *const bar,
    int size )
```

Sets the splitter size for the given sidebar or splitter child widget to size. Special value -1: Sets the minimum size hint of the widget.

### 6.1331.2.6 size()

```
int Digikam::SidebarSplitter::size (
    Sidebar *const bar ) const
```

Returns the value of sizes() that corresponds to the given [Sidebar](#) or splitter child widget.

## 6.1332 Digikam::SidebarSplitter::Private Class Reference

### Public Attributes

- `QList< Sidebar * >` **sidebars**

## 6.1333 Digikam::SidebarState Class Reference

### Public Member Functions

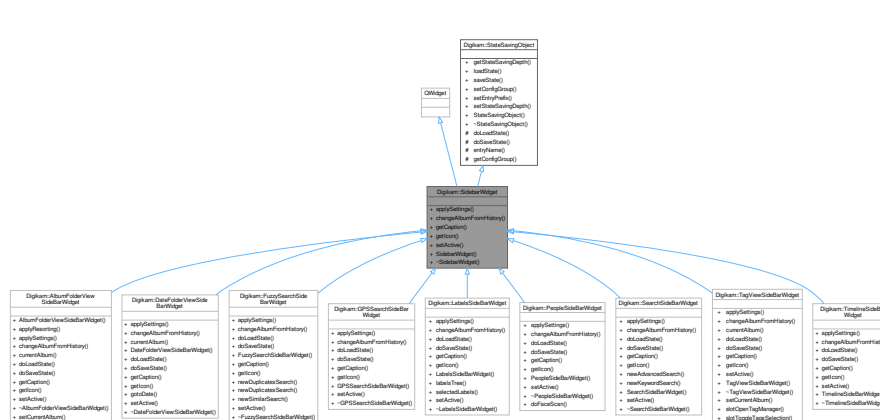
- **SidebarState** (`QWidget *const w, int size`)

## Public Attributes

- QWidget \* **activeWidget** = nullptr
- int **size** = 0

## 6.1334 Digikam::SidebarWidget Class Reference

Inheritance diagram for Digikam::SidebarWidget:



## Signals

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

## Public Member Functions

- virtual void [applySettings](#) ()=0
- virtual void [changeAlbumFromHistory](#) (const QList< [Album](#) \* > &album)=0
- virtual const QString [getCaption](#) ()=0
- virtual const QIcon [getIcon](#) ()=0
- virtual void [setActive](#) (bool active)=0
- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

## Public Member Functions inherited from Digikam::StateSavingObject

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()



## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- virtual void [doLoadState](#) ()=0
- virtual void [doSaveState](#) ()=0
- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

### 6.1334.1 Detailed Description

Abstract base class for widgets that are use in one of digikams's sidebars.

### 6.1334.2 Constructor & Destructor Documentation

#### 6.1334.2.1 SidebarWidget()

```
Digikam::SidebarWidget::SidebarWidget (
    QWidget *const parent ) [explicit]
```

Constructor.

Parameters

<i>parent</i>	the parent of this widget, may be null
---------------	--

#### 6.1334.2.2 ~SidebarWidget()

```
Digikam::SidebarWidget::~~SidebarWidget ( ) [override], [default]
```

Destructor.

### 6.1334.3 Member Function Documentation

#### 6.1334.3.1 applySettings()

```
virtual void Digikam::SidebarWidget::applySettings ( ) [pure virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implemented in [Digikam::AlbumFolderViewSideBarWidget](#), [Digikam::DateFolderViewSideBarWidget](#), [Digikam::FuzzySearchSideBarV](#), [Digikam::GPSSearchSideBarWidget](#), [Digikam::LabelsSideBarWidget](#), [Digikam::PeopleSideBarWidget](#), [Digikam::SearchSideBarWidg](#), [Digikam::TagViewSideBarWidget](#), and [Digikam::TimelineSideBarWidget](#).

### 6.1334.3.2 `changeAlbumFromHistory()`

```
virtual void Digikam::SidebarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [pure virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implemented in [Digikam::AlbumFolderViewSideBarWidget](#), [Digikam::DateFolderViewSideBarWidget](#), [Digikam::FuzzySearchSideBarV](#), [Digikam::GPSSearchSideBarWidget](#), [Digikam::LabelsSideBarWidget](#), [Digikam::PeopleSideBarWidget](#), [Digikam::SearchSideBarWidg](#), [Digikam::TagViewSideBarWidget](#), and [Digikam::TimelineSideBarWidget](#).

### 6.1334.3.3 `getCaption()`

```
virtual const QString Digikam::SidebarWidget::getCaption ( ) [pure virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implemented in [Digikam::AlbumFolderViewSideBarWidget](#), [Digikam::DateFolderViewSideBarWidget](#), [Digikam::FuzzySearchSideBarV](#), [Digikam::GPSSearchSideBarWidget](#), [Digikam::LabelsSideBarWidget](#), [Digikam::PeopleSideBarWidget](#), [Digikam::SearchSideBarWidg](#), [Digikam::TagViewSideBarWidget](#), and [Digikam::TimelineSideBarWidget](#).

### 6.1334.3.4 `getIcon()`

```
virtual const QIcon Digikam::SidebarWidget::getIcon ( ) [pure virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implemented in [Digikam::AlbumFolderViewSideBarWidget](#), [Digikam::DateFolderViewSideBarWidget](#), [Digikam::FuzzySearchSideBarV](#), [Digikam::GPSSearchSideBarWidget](#), [Digikam::LabelsSideBarWidget](#), [Digikam::PeopleSideBarWidget](#), [Digikam::SearchSideBarWidg](#), [Digikam::TagViewSideBarWidget](#), and [Digikam::TimelineSideBarWidget](#).

### 6.1334.3.5 `requestActiveTab`

```
void Digikam::SidebarWidget::requestActiveTab (
    SidebarWidget * ) [signal]
```

This signal can be emitted if this sidebar widget wants to be the one that is active.

### 6.1334.3.6 `setActive()`

```
virtual void Digikam::SidebarWidget::setActive (
    bool active ) [pure virtual]
```

This method is called if the visible sidebar widget is changed.

## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implemented in [Digikam::AlbumFolderViewSideBarWidget](#), [Digikam::DateFolderViewSideBarWidget](#), [Digikam::FuzzySearchSideBarWidget](#), [Digikam::GPSSearchSideBarWidget](#), [Digikam::LabelsSideBarWidget](#), [Digikam::PeopleSideBarWidget](#), [Digikam::SearchSideBarWidget](#), [Digikam::TagViewSideBarWidget](#), and [Digikam::TimelineSideBarWidget](#).

## 6.1334.3.7 signalNotificationError

```
void Digikam::SidebarWidget::signalNotificationError (
    const QString & message,
    int type ) [signal]
```

To dispatch error message to temporized pop-up notification widget hosted with icon-view.

## 6.1335 Digikam::SidecarFinder Class Reference

## Public Member Functions

- **SidecarFinder** (const QList< QUrl > &files)

## Public Attributes

- QList< bool > **localFileModes**
- QList< QUrl > **localFiles**
- QList< QString > **localFileSuffixes**

## 6.1336 Digikam::SimilarityDb Class Reference

## Public Member Functions

- void [clearImageSimilarity](#) (FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)
- void [copySimilarityAttributes](#) (qulonglong srcId, qulonglong destId)
- QList< qulonglong > [getDirtyOrMissingFingerprints](#) (const QList< [ItemInfo](#) > &imageInfos, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)
- QStringList [getDirtyOrMissingFingerprintURLs](#) (const QList< [ItemInfo](#) > &imageInfos, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)
- double [getImageSimilarity](#) (qulonglong imageID1, qulonglong imageID2, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)
- QList< FuzzyAlgorithm > [getImageSimilarityAlgorithms](#) (qulonglong imageID1, qulonglong imageID2)
- QString [getLegacySetting](#) (const QString &keyword)
- QString [getSetting](#) (const QString &keyword)
- bool [hasDirtyOrMissingFingerprint](#) (const [ItemInfo](#) &imageInfo, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar) const
- bool [hasFingerprint](#) (qulonglong imageId, FuzzyAlgorithm algorithm) const
- bool [hasFingerprints](#) ()
- bool [hasFingerprints](#) (FuzzyAlgorithm algorithm) const

- bool `integrityCheck ()`
- `QSet< qlonglong > registeredImageIds () const`
- void `removeImageFingerprint (qlonglong imageID, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)`
- void `removeImageSimilarity (qlonglong imageID, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)`
- void `removeImageSimilarity (qlonglong imageID1, qlonglong imageID2, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)`
- void `setImageSimilarity (qlonglong imageID1, qlonglong imageID2, double value, FuzzyAlgorithm algorithm=FuzzyAlgorithm::Haar)`
- bool `setSetting (const QString &keyword, const QString &value)`
- void `vacuum ()`

## Friends

- class `SimilarityDbAccess`

## 6.1336.1 Member Function Documentation

### 6.1336.1.1 clearImageSimilarity()

```
void Digikam::SimilarityDb::clearImageSimilarity (
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar )
```

This method removes all image similarity entries for the algorithm.

#### Parameters

<i>algorithm</i>	The algorithm.
------------------	----------------

### 6.1336.1.2 copySimilarityAttributes()

```
void Digikam::SimilarityDb::copySimilarityAttributes (
    qlonglong srcId,
    qlonglong destId )
```

Copies all similarity-specific information, from image srcId to destId.

### 6.1336.1.3 getDirtyOrMissingFingerprints()

```
QList< qlonglong > Digikam::SimilarityDb::getDirtyOrMissingFingerprints (
    const QList< ItemInfo > & imageInfos,
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar )
```

Returns a list of all item ids (images, videos,...) where either no fingerprint for the given algorithm exists or is outdated because the file is identified as changed since the generation of the fingerprint.

#### Parameters

<i>imageInfos</i>	The image info objects representing the items.
<i>algorithm</i>	The algorithm.

**Returns**

The ids of the items whose fingerprints are dirty or missing.

**6.1336.1.4 getDirtyOrMissingFingerprintURLs()**

```
QStringList Digikam::SimilarityDb::getDirtyOrMissingFingerprintURLs (
    const QList< ItemInfo > & imageInfos,
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar )
```

Returns a list of the URLs of all items (images, videos,...) where either no fingerprint for the given algorithm exists or is outdated because the file is identified as changed since the generation of the fingerprint.

**Parameters**

<i>imageInfos</i>	The image info objects representing the items.
<i>algorithm</i>	The algorithm.

**Returns**

The URLs of the items whose fingerprints are dirty or missing.

**6.1336.1.5 getImageSimilarity()**

```
double Digikam::SimilarityDb::getImageSimilarity (
    qulonglong imageID1,
    qulonglong imageID2,
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar )
```

Returns the similarity value for two images. A value of -1 means nonexistence. A value of -2 means that there is a value that cannot be converted into a double

**6.1336.1.6 getImageSimilarityAlgorithms()**

```
QList< FuzzyAlgorithm > Digikam::SimilarityDb::getImageSimilarityAlgorithms (
    qulonglong imageID1,
    qulonglong imageID2 )
```

Returns the algorithms for which a similarity value exists for the given image ids.

**Parameters**

<i>imageID1</i>	The first image id.
<i>imageID2</i>	The second image id.

**Returns**

a list of all algorithms for which a similarity value exists.

### 6.1336.1.7 getLegacySetting()

```
QString Digikam::SimilarityDb::getLegacySetting (
    const QString & keyword )
```

Returns the legacy settings with the keyword name.

#### Parameters

<i>keyword</i>	The setting entry name.
----------------	-------------------------

#### Returns

The setting value.

### 6.1336.1.8 getSetting()

```
QString Digikam::SimilarityDb::getSetting (
    const QString & keyword )
```

Returns the setting with the keyword name.

#### Parameters

<i>keyword</i>	The setting entry name.
----------------	-------------------------

#### Returns

The setting value.

### 6.1336.1.9 hasDirtyOrMissingFingerprint()

```
bool Digikam::SimilarityDb::hasDirtyOrMissingFingerprint (
    const ItemInfo & imageInfo,
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar ) const
```

Checks if the given image has a dirty fingerprint or even none for the given algorithm.

#### Parameters

<i>imageInfo</i>	The image info object representing the item.
<i>algorithm</i>	The algorithm used for the fingerprint.

#### Returns

True, if the image either has no or a dirty fingerprint.

**6.1336.1.10 hasFingerprint()**

```
bool Digikam::SimilarityDb::hasFingerprint (
    qlonglong imageId,
    FuzzyAlgorithm algorithm ) const
```

This method checks if the given image has a fingerprint for the given algorithm.

**Parameters**

<i>imageId</i>	The Id of the image to check.
<i>algorithm</i>	The algorithm.

**Returns**

True, if there is a fingerprint.

**6.1336.1.11 hasFingerprints() [1/2]**

```
bool Digikam::SimilarityDb::hasFingerprints ( )
```

This method checks if there are any fingerprints for any algorithm present.

**Returns**

True, if fingerprints exist.

**6.1336.1.12 hasFingerprints() [2/2]**

```
bool Digikam::SimilarityDb::hasFingerprints (
    FuzzyAlgorithm algorithm ) const
```

This method checks if there are any fingerprints for the given algorithm.

**Parameters**

<i>algorithm</i>	The algorithm.
------------------	----------------

**Returns**

true, if there are fingerprints and false, otherwise.

**6.1336.1.13 integrityCheck()**

```
bool Digikam::SimilarityDb::integrityCheck ( )
```

This method checks the integrity of the similarity database.

**Returns**

true, if the integrity check was passed and false, else.

**6.1336.1.14 registeredImageIds()**

```
QSet< qulonglong > Digikam::SimilarityDb::registeredImageIds ( ) const
```

This method returns all image ids that are present in the similarity db tables.

**Returns**

a set of all present image ids.

**6.1336.1.15 removeImageFingerprint()**

```
void Digikam::SimilarityDb::removeImageFingerprint (
    qulonglong imageID,
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar )
```

This method removes the fingerprint entry for the given imageId and algorithm. Also, this automatically removes the entries in the ImageSimilarities table for the given algorithm and image id.

**Parameters**

<i>imageID</i>	The image id.
<i>algorithm</i>	The algorithm.

**6.1336.1.16 removeImageSimilarity() [1/2]**

```
void Digikam::SimilarityDb::removeImageSimilarity (
    qulonglong imageID,
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar )
```

This method removes the image similarity entries for the imageID and algorithm.

**Parameters**

<i>imageID</i>	The image id.
<i>algorithm</i>	The algorithm.

**6.1336.1.17 removeImageSimilarity() [2/2]**

```
void Digikam::SimilarityDb::removeImageSimilarity (
    qulonglong imageID1,
    qulonglong imageID2,
    FuzzyAlgorithm algorithm = FuzzyAlgorithm::Haar )
```



This method removes the image similarity entry for the imageIDs and algorithm.

#### Parameters

<i>imageID1</i>	The first image id.
<i>imageID2</i>	The second image id.
<i>algorithm</i>	The algorithm.

#### 6.1336.1.18 setSetting()

```
bool Digikam::SimilarityDb::setSetting (
    const QString & keyword,
    const QString & value )
```

Set the database setting entry given by keyword to the given value.

#### Parameters

<i>keyword</i>	The keyword, i.e. setting name.
<i>value</i>	The value.

#### Returns

True, if the value was set and false, else..

#### 6.1336.1.19 vacuum()

```
void Digikam::SimilarityDb::vacuum ( )
```

This method shrinks the database.

## 6.1337 Digikam::SimilarityDbAccess Class Reference

### Public Member Functions

- [SimilarityDbBackend](#) \* **backend** () const
- [SimilarityDb](#) \* **db** () const
- QString **lastError** () const
- void **setLastError** (const QString &error)
- [SimilarityDbAccess](#) ()

### Static Public Member Functions

- static bool **checkReadyForUse** ([InitializationObserver](#) \*const observer)
- static void **cleanUpDatabase** ()
- static void **initDbEngineErrorHandler** ([DbEngineErrorHandler](#) \*const errorhandler)
- static bool **isInitialized** ()
- static [DbEngineParameters](#) **parameters** ()
- static void **setParameters** (const [DbEngineParameters](#) &parameters)

## 6.1337.1 Constructor & Destructor Documentation

### 6.1337.1.1 SimilarityDbAccess()

```
Digikam::SimilarityDbAccess::SimilarityDbAccess ( )
```

This class is written in analogy to [CoreDbAccess](#) (some features stripped off). For documentation, see [coredbaccess.h](#)

## 6.1337.2 Member Function Documentation

### 6.1337.2.1 checkReadyForUse()

```
bool Digikam::SimilarityDbAccess::checkReadyForUse (
    InitializationObserver *const observer ) [static]
```

This static method checks if the similarity db is ready for use.

#### Parameters

<i>observer</i>	the observer.
-----------------	---------------

#### Returns

true, if the database is ready for use.

### 6.1337.2.2 cleanUpDatabase()

```
void Digikam::SimilarityDbAccess::cleanUpDatabase ( ) [static]
```

This static method removes the connection to the similarity database.

### 6.1337.2.3 initDbEngineErrorHandler()

```
void Digikam::SimilarityDbAccess::initDbEngineErrorHandler (
    DbEngineErrorHandler *const errorhandler ) [static]
```

This static method initialises the error handler for the similarity db.

#### Parameters

<i>errorhandler</i>	The error handler.
---------------------	--------------------

### 6.1337.2.4 isInitialized()

```
bool Digikam::SimilarityDbAccess::isInitialized ( ) [static]
```

This static method returns if the similarity db is initialised.

**Returns**

true, if the similarityDb is initialised.

**6.1337.2.5 parameters()**

```
DbEngineParameters Digikam::SimilarityDbAccess::parameters ( ) [static]
```

This static method returns the current db parameters.

**Returns**

the current db parameters.

**6.1337.2.6 setLastError()**

```
void Digikam::SimilarityDbAccess::setLastError (
    const QString & error )
```

Set the "last error" message. This method is not for public use.

**6.1337.2.7 setParameters()**

```
void Digikam::SimilarityDbAccess::setParameters (
    const DbEngineParameters & parameters ) [static]
```

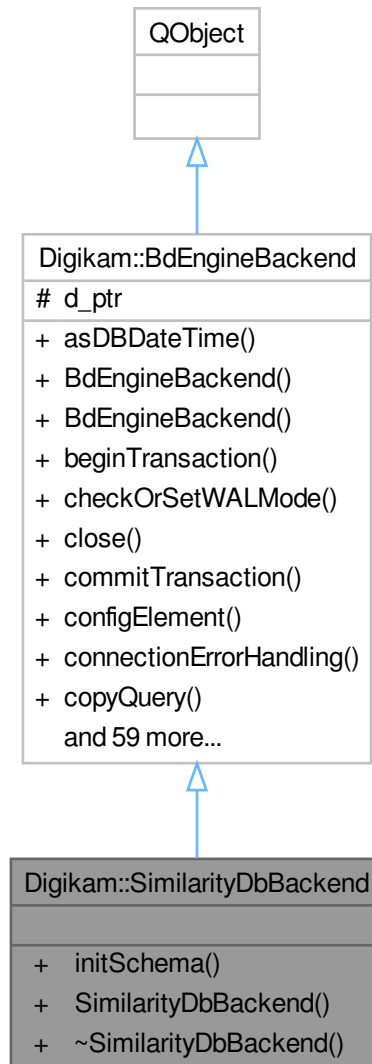
This static method sets the database parameters that are needed to initialise the db connection.

**Parameters**

<i>parameters</i>	The db parameters.
-------------------	--------------------

## 6.1338 Digikam::SimilarityDbBackend Class Reference

Inheritance diagram for Digikam::SimilarityDbBackend:



### Public Member Functions

- bool `initSchema` ([SimilarityDbSchemaUpdater](#) \*const updater)
- **SimilarityDbBackend** ([DbEngineLocking](#) \*const locking, const QString &backendName=QLatin1↳String("similarityDatabase-"))

### Public Member Functions inherited from [Digikam::BdEngineBackend](#)

- QDateTime `asDBDateTime` (const QDateTime &dateTime) const

- [BdEngineBackend](#) (const QString &backendName, [DbEngineLocking](#) \*const locking)
- **BdEngineBackend** (const QString &backendName, [DbEngineLocking](#) \*const locking, [BdEngineBackendPrivate](#) &dd)
- [BdEngineBackend::QueryState](#) [beginTransaction](#) ()
- bool [checkOrSetWALMode](#) ()
- void [close](#) ()
- [BdEngineBackend::QueryState](#) [commitTransaction](#) ()
- [DbEngineConfigSettings](#) [configElement](#) () const
- bool [connectionErrorHandling](#) (int retries)
- [DbEngineSqlQuery](#) [copyQuery](#) (const [DbEngineSqlQuery](#) &old)
- DbType [databaseType](#) () const
- bool [exec](#) ([DbEngineSqlQuery](#) &query)
- bool **execBatch** ([DbEngineSqlQuery](#) &query)
- [QueryState](#) [execDBAction](#) (const [DbEngineAction](#) &action, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) [execDBAction](#) (const [DbEngineAction](#) &action, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execDBAction** (const QString &action, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execDBAction** (const QString &action, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- QSqlQuery [execDBActionQuery](#) (const [DbEngineAction](#) &action, const QMap< QString, QVariant > &bindingMap)
- QSqlQuery **execDBActionQuery** (const QString &action, const QMap< QString, QVariant > &bindingMap)
- [QueryState](#) [execDirectSql](#) (const QString &query)
- [QueryState](#) [execDirectSqlWithResult](#) (const QString &query, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [DbEngineSqlQuery](#) [execQuery](#) (const QString &sql)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QList< QVariant > &boundValues)
- [DbEngineSqlQuery](#) [execQuery](#) (const QString &sql, const QMap< QString, QVariant > &bindingMap)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QList< QVariant > &boundValues)
- void [execQuery](#) ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)
- [QueryState](#) **execSql** (const QString &sql, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) [execSql](#) (const QString &sql, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)

- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execUpsertDBAction** (const [DbEngineAction](#) &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [QueryState](#) **execUpsertDBAction** (const QString &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [DbEngineAction](#) **getDBAction** (const QString &actionName) const
- [DbEngineSqlQuery](#) **getQuery** ()
- [QueryState](#) **handleQueryResult** ([DbEngineSqlQuery](#) &query, QList< QVariant > \*const values, QVariant \*const lastInsertId)
- bool **isCompatible** (const [DbEngineParameters](#) &parameters)
- bool **isInTransaction** () const
- bool **isOpen** () const
- bool **isReady** () const
- QString **lastError** ()
- QSqlError **lastSQLError** ()
- int **maximumBoundValues** () const
- bool **open** (const [DbEngineParameters](#) &parameters)
- [DbEngineSqlQuery](#) **prepareQuery** (const QString &sql)
- bool **queryErrorHandling** ([DbEngineSqlQuery](#) &query, int retries)
- QList< QVariant > **readToList** ([DbEngineSqlQuery](#) &query)
- void **rollbackTransaction** ()
- void **setDbEngineErrorHandler** ([DbEngineErrorHandler](#) \*const handler)
- void **setForeignKeyChecks** (bool check)
- [Status](#) **status** () const
- QStringList **tables** ()
- bool **transactionErrorHandling** (const QSqlError &lastError, int retries)

### Additional Inherited Members

### Public Types inherited from [Digikam::BdEngineBackend](#)

- enum **DbType** { [SQLite](#) , [MySQL](#) }
- enum **QueryOperationStatus** { [ExecuteNormal](#) , [Wait](#) , [AbortQueries](#) }
- enum **QueryStateEnum** { [NoErrors](#) , [SQLError](#) , [ConnectionError](#) }
- enum **Status** { [Unavailable](#) , [Open](#) , [OpenSchemaChecked](#) }

## Protected Attributes inherited from Digikam::BdEngineBackend

- `BdEngineBackendPrivate` \*const `d_ptr` = nullptr

### 6.1338.1 Member Function Documentation

#### 6.1338.1.1 initSchema()

```
bool Digikam::SimilarityDbBackend::initSchema (
    SimilarityDbSchemaUpdater *const updater )
```

Initialize the database schema to the current version, carry out upgrades if necessary. Shall only be called from the thread that called `open()`.

## 6.1339 Digikam::SimilarityDbSchemaUpdater Class Reference

### Public Member Functions

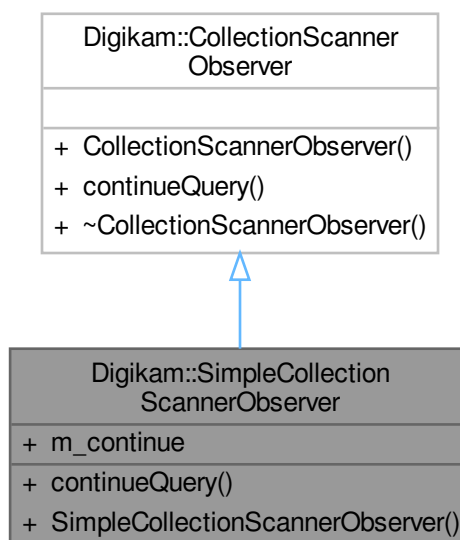
- void `setObserver` (`InitializationObserver` \*const observer)
- `SimilarityDbSchemaUpdater` (`SimilarityDbAccess` \*const dbAccess)
- bool `update` ()

### Static Public Member Functions

- static int `schemaVersion` ()

## 6.1340 Digikam::SimpleCollectionScannerObserver Class Reference

Inheritance diagram for Digikam::SimpleCollectionScannerObserver:



**Public Member Functions**

- bool `continueQuery` () override
- `SimpleCollectionScannerObserver` (bool \*const var)

**Public Attributes**

- bool \* `m_continue` = nullptr

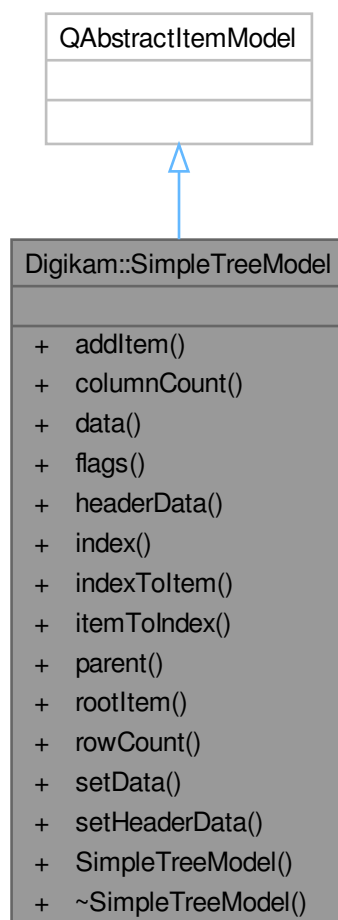
**6.1340.1 Member Function Documentation****6.1340.1.1 continueQuery()**

bool Digikam::SimpleCollectionScannerObserver::continueQuery ( ) [override], [virtual]

Implements [Digikam::CollectionScannerObserver](#).

**6.1341 Digikam::SimpleTreeModel Class Reference**

Inheritance diagram for Digikam::SimpleTreeModel:





## Classes

- class [Item](#)

## Public Member Functions

- [Item](#) \* **addItem** ([Item](#) \*const parentItem=nullptr, const int rowNumber=-1)
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override  
*QAbstractItemModel:*
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- QVariant **headerData** (int section, Qt::Orientation orientation, int role) const override
- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- [Item](#) \* **indexToItem** (const QModelIndex &itemIndex) const
- QModelIndex **itemToIndex** (const [Item](#) \*const item) const
- QModelIndex **parent** (const QModelIndex &index) const override
- [Item](#) \* **rootItem** () const
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role) override
- bool **setHeaderData** (int section, Qt::Orientation orientation, const QVariant &value, int role) override
- **SimpleTreeModel** (const int [columnCount](#), QObject \*const parent=nullptr)

## 6.1342 Digikam::SimpleTreeModel::Item Class Reference

### Public Attributes

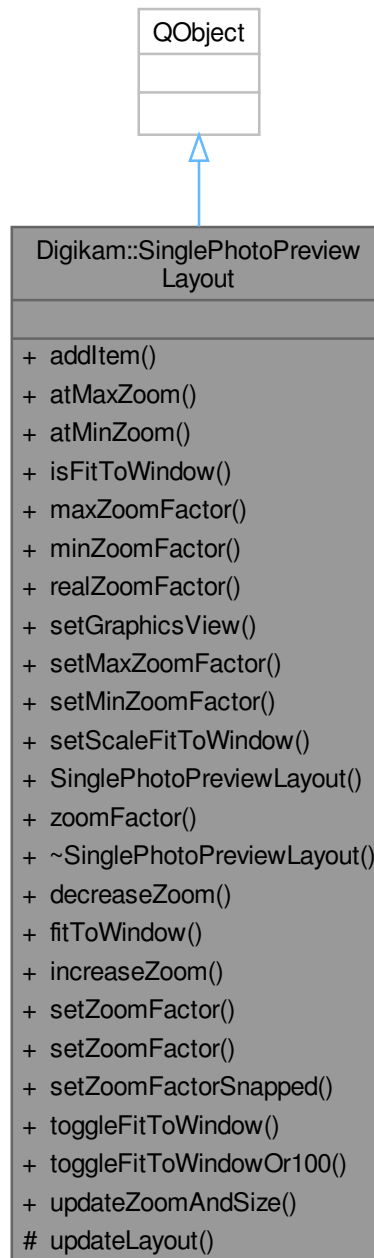
- QString **data**

### Friends

- class **SimpleTreeModel**

## 6.1343 Digikam::SinglePhotoPreviewLayout Class Reference

Inheritance diagram for Digikam::SinglePhotoPreviewLayout:



### Public Types

- enum `SetZoomFlag` { `JustSetFactor` = 0 , `CenterView` = 1 << 0 , `SnapZoomFactor` = 1 << 1 }

## Public Slots

- void **decreaseZoom** (const QPoint &viewportAnchor=QPoint())
- void **fitToWindow** ()
- void **increaseZoom** (const QPoint &viewportAnchor=QPoint())
- void **setZoomFactor** (double z, const QPoint &viewportAnchor=QPoint(), SetZoomFlags flags=JustSetFactor)
- void **setZoomFactor** (double z, SetZoomFlags flags)
- void **setZoomFactorSnapped** (double z)
- void **toggleFitToWindow** ()  
*Toggle between fitToWindow and previous zoom factor.*
- void **toggleFitToWindowOr100** ()  
*Toggle between fitToWindow and zoom factor 1.*
- void **updateZoomAndSize** ()  
*Update settings when size of image or view changed.*

## Signals

- void **fitToWindowToggled** (bool fitToWindow)
- void **zoomFactorChanged** (double)

## Public Member Functions

- void **addItem** (GraphicsDImgItem \*const item)
- bool **atMaxZoom** () const
- bool **atMinZoom** () const
- bool **isFitToWindow** () const
- double **maxZoomFactor** () const
- double **minZoomFactor** () const
- double **realZoomFactor** () const
- void **setGraphicsView** (GraphicsDImgView \*const view)
- void **setMaxZoomFactor** (double z)
- void **setMinZoomFactor** (double z)
- void **setScaleFitToWindow** (bool value)
- **SinglePhotoPreviewLayout** (QObject \*const parent)
- double **zoomFactor** () const

## Protected Member Functions

- void **updateLayout** ()

## 6.1343.1 Member Function Documentation

### 6.1343.1.1 addItem()

```
void Digikam::SinglePhotoPreviewLayout::addItem (
    GraphicsDImgItem *const item )
```

Set the item to layout. For a SinglePhoto layout, typically, you can add only one item.

**6.1343.1.2 maxZoomFactor()**

```
double Digikam::SinglePhotoPreviewLayout::maxZoomFactor ( ) const
```

The zoom range for incrementing and decrementing.

**6.1343.1.3 setGraphicsView()**

```
void Digikam::SinglePhotoPreviewLayout::setGraphicsView (
    GraphicsDImgView *const view )
```

Set the graphics view, and associated scene, to operate on.

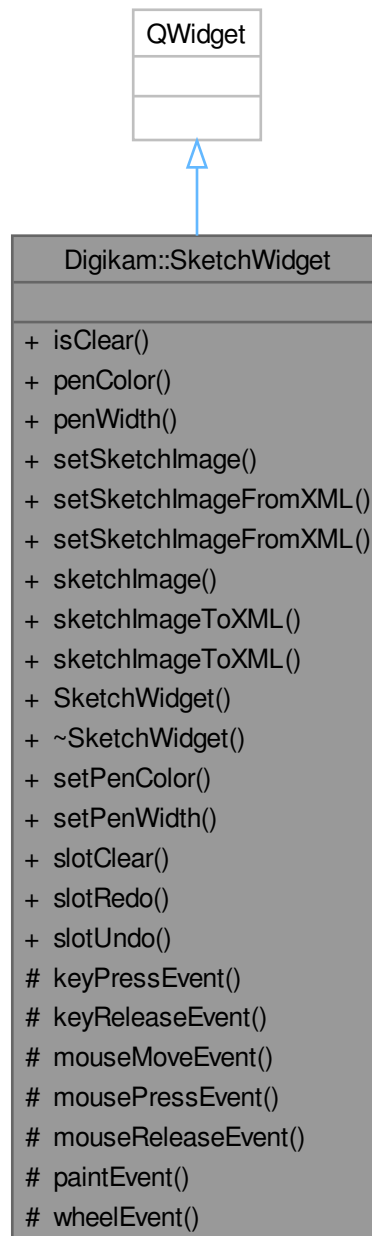
**6.1343.1.4 setScaleFitToWindow()**

```
void Digikam::SinglePhotoPreviewLayout::setScaleFitToWindow (
    bool value )
```

Set to true to scale small images to fit to window.

## 6.1344 Digikam::SketchWidget Class Reference

Inheritance diagram for Digikam::SketchWidget:



### Public Slots

- void **setPenColor** (const QColor &newColor)
- void **setPenWidth** (int newWidth)
- void **slotClear** ()
- void **slotRedo** ()
- void **slotUndo** ()

## Signals

- void **signalPenColorChanged** (const QColor &)
- void **signalPenSizeChanged** (int)
- void **signalSketchChanged** (const QImage &)
- void **signalUndoRedoStateChanged** (bool hasUndo, bool hasRedo)

## Public Member Functions

- bool **isClear** () const
- QColor **penColor** () const
- int **penWidth** () const
- void **setSketchImage** (const QImage &image)
- bool **setSketchImageFromXML** (const QString &xml)
- bool **setSketchImageFromXML** (QXmlStreamReader &reader)
- QImage **sketchImage** () const
- QString **sketchImageToXML** ()
- void **sketchImageToXML** (QXmlStreamWriter &writer)
- **SketchWidget** (QWidget \*const parent=nullptr)

## Protected Member Functions

- void **keyPressEvent** (QKeyEvent \*) override
- void **keyReleaseEvent** (QKeyEvent \*) override
- void **mouseMoveEvent** (QMouseEvent \*) override
- void **mousePressEvent** (QMouseEvent \*) override
- void **mouseReleaseEvent** (QMouseEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **wheelEvent** (QWheelEvent \*) override

## 6.1344.1 Member Function Documentation

### 6.1344.1.1 setSketchImageFromXML()

```
bool Digikam::SketchWidget::setSketchImageFromXML (
    QXmlStreamReader & reader )
```

This method set sketch image using XML data based on drawing line history. Return true if data are imported successfully.

### 6.1344.1.2 sketchImageToXML()

```
void Digikam::SketchWidget::sketchImageToXML (
    QXmlStreamWriter & writer )
```

This method return the drawing line history as XML, to be stored in database as [SAAlbum](#) data.

## 6.1345 Digikam::SlideVideo Class Reference

Inheritance diagram for Digikam::SlideVideo:



### Public Slots

- void **slotPositionChanged** (int position)
- void **slotVolumeChanged** (int volume)

### Signals

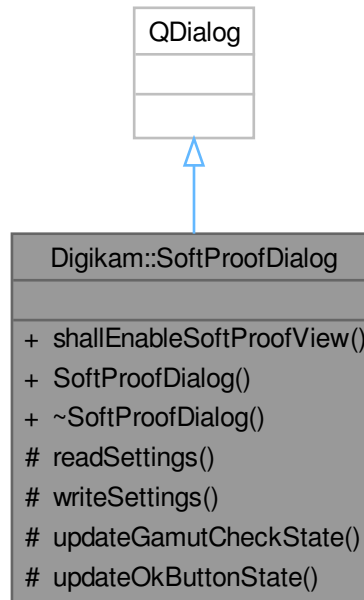
- void **signalVideoDuration** (qint64)
- void **signalVideoFinished** ()
- void **signalVideoLoaded** (bool)
- void **signalVideoPosition** (qint64)
- void **signalVideoVolume** (int)

### Public Member Functions

- void **backward** ()
- void **forward** ()
- void **pause** (bool)
- void **setCurrentUrl** (const QUrl &url)
- void **setInfoInterface** ([DInfoInterface](#) \*const iface)
- **SlideVideo** (QWidget \*const parent)
- void **stop** ()

## 6.1346 Digikam::SoftProofDialog Class Reference

Inheritance diagram for Digikam::SoftProofDialog:



### Public Member Functions

- bool **shallEnableSoftProofView** () const
- **SoftProofDialog** (QWidget \*const parent)

### Protected Slots

- void **updateGamutCheckState** ()
- void **updateOkButtonState** ()

### Protected Member Functions

- void **readSettings** ()
- void **writeSettings** ()



## 6.1347 Digikam::SolidHardwareDlg Class Reference

Inheritance diagram for Digikam::SolidHardwareDlg:



### Public Member Functions

- **SolidHardwareDlg** (QWidget \*const parent)

### Public Member Functions inherited from [Digikam::InfoDlg](#)

- QDialogButtonBox \* **buttonBox** () const
- **InfoDlg** (QWidget \*const parent)
- QTreeWidget \* **listView** () const
- QWidget \* **mainWidget** () const
- virtual void **setInfoMap** (const QMap< QString, QString > &list)
- QTabWidget \* **tabView** () const

## 6.1348 Digikam::SolidVolumeInfo Class Reference

### Public Member Functions

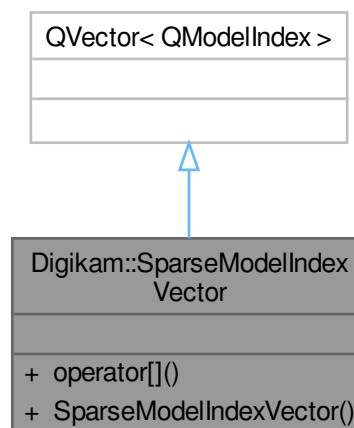
- bool **isNull** () const

### Public Attributes

- bool **isMounted** = false  
*is mounted on File System.*
- bool **isOpticalDisc** = false  
*is an optical disk device as CD/DVD/BR*
- bool **isRemovable** = false  
*may be removed*
- QString **label**  
*volume label (think of CDs)*
- QString **path**  
*mount path of volume, with trailing slash*
- QString **udi**  
*Solid device UDI of the StorageAccess device.*
- QString **uuid**  
*UUID as from Solid.*

## 6.1349 Digikam::SparseModelIndexVector Class Reference

Inheritance diagram for Digikam::SparseModelIndexVector:

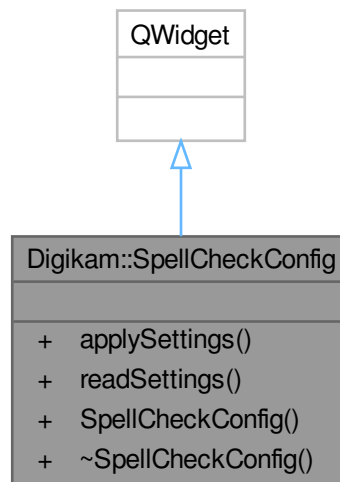


**Public Member Functions**

- `QModelIndex & operator[]` (int i)
- `SparseModelIndexVector` (int rowCount, QAbstractItemModel \*const model\_, int column\_)

## 6.1350 Digikam::SpellCheckConfig Class Reference

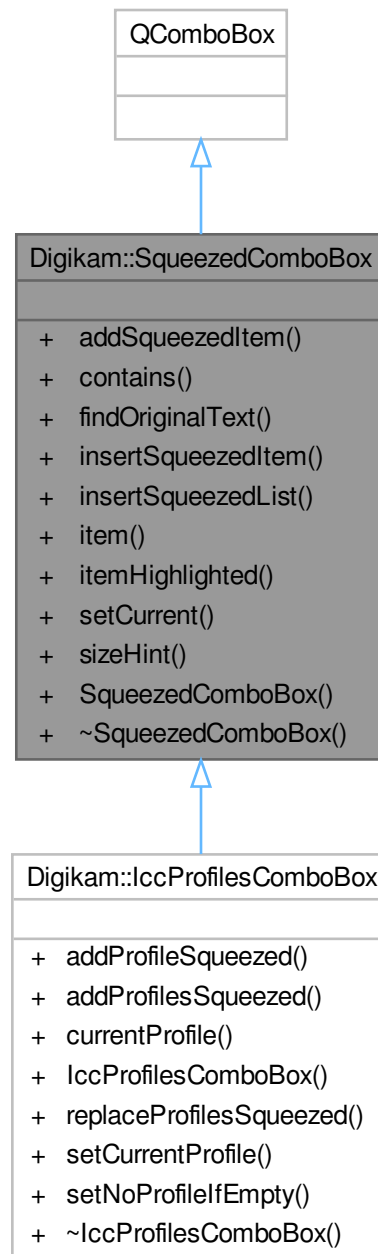
Inheritance diagram for Digikam::SpellCheckConfig:

**Public Member Functions**

- void `applySettings` ()
- void `readSettings` ()
- `SpellCheckConfig` (QWidget \*const parent=nullptr)

## 6.1351 Digikam::SqueezedComboBox Class Reference

Inheritance diagram for Digikam::SqueezedComboBox:



### Signals

- void **signalItemActivated** (const QString &)

## Public Member Functions

- void [addSqueezedItem](#) (const QString &newItem, const QVariant &userData=QVariant())
- bool [contains](#) (const QString &text) const
- int [findOriginalText](#) (const QString &text, Qt::CaseSensitivity cs=Qt::CaseSensitive) const
- void [insertSqueezedItem](#) (const QString &newItem, int index, const QVariant &userData=QVariant())
- void [insertSqueezedList](#) (const QStringList &newItems, int index)
- QString [item](#) (int index) const
- QString [itemHighlighted](#) () const
- void [setCurrent](#) (const QString &itemText)
- QSize [sizeHint](#) () const override
- [SqueezedComboBox](#) (QWidget \*const parent=nullptr, const char \*name=nullptr)
- [~SqueezedComboBox](#) () override

### 6.1351.1 Detailed Description

This widget is a QComboBox, but then a little bit different. It only shows the right part of the items depending on de size of the widget. When it is not possible to show the complete item, it will be shortened and "..." will be prepended.

### 6.1351.2 Constructor & Destructor Documentation

#### 6.1351.2.1 SqueezedComboBox()

```
Digikam::SqueezedComboBox::SqueezedComboBox (
    QWidget *const parent = nullptr,
    const char * name = nullptr ) [explicit]
```

Constructor

#### Parameters

<i>parent</i>	the parent widget
<i>name</i>	the name to give to the widget

#### 6.1351.2.2 ~SqueezedComboBox()

```
Digikam::SqueezedComboBox::~SqueezedComboBox ( ) [override]
```

destructor

### 6.1351.3 Member Function Documentation

#### 6.1351.3.1 addSqueezedItem()

```
void Digikam::SqueezedComboBox::addSqueezedItem (
    const QString & newItem,
    const QVariant & userData = QVariant() )
```

Append an item.

## Parameters

<i>newItem</i>	the original (long version) of the item which needs to be added to the combobox
<i>userData</i>	custom meta-data assigned to new item.

**6.1351.3.2 contains()**

```
bool Digikam::SqueezedComboBox::contains (
    const QString & text ) const
```

Returns true if the combobox contains the original (not-squeezed) version of text.

## Parameters

<i>text</i>	the original (not-squeezed) text to check for
-------------	---

**6.1351.3.3 findOriginalText()**

```
int Digikam::SqueezedComboBox::findOriginalText (
    const QString & text,
    Qt::CaseSensitivity cs = Qt::CaseSensitive ) const
```

Returns the index of the combobox if found the original (not-squeezed) version of text.

## Parameters

<i>text</i>	the original (not-squeezed) text to find for
<i>cs</i>	case sensitive or case insensitive search

**6.1351.3.4 insertSqueezedItem()**

```
void Digikam::SqueezedComboBox::insertSqueezedItem (
    const QString & newItem,
    int index,
    const QVariant & userData = QVariant() )
```

This inserts a item to the list. See `QComboBox::insertItem()` for details. Please do not use `QComboBox::insertItem()` to this widget, as that will fail.

## Parameters

<i>newItem</i>	the original (long version) of the item which needs to be added to the combobox
<i>index</i>	the position in the widget.
<i>userData</i>	custom meta-data assigned to new item.

### 6.1351.3.5 insertSqueezedList()

```
void Digikam::SqueezedComboBox::insertSqueezedList (
    const QStringList & newItems,
    int index )
```

This inserts items to the list. See `QComboBox::insertItems()` for details. Please do not use `QComboBox::insertItems()` to this widget, as that will fail.

#### Parameters

<i>newItems</i>	the originals (long version) of the items which needs to be added to the combobox
<i>index</i>	the position in the widget.

### 6.1351.3.6 item()

```
QString Digikam::SqueezedComboBox::item (
    int index ) const
```

This method returns the full text (not squeezed) for the index.

#### Parameters

<i>index</i>	the position in the widget.
--------------	-----------------------------

#### Returns

full text of the item

### 6.1351.3.7 itemHighlighted()

```
QString Digikam::SqueezedComboBox::itemHighlighted ( ) const
```

This method returns the full text (not squeezed) of the currently highlighted item.

#### Returns

full text of the highlighted item

### 6.1351.3.8 setCurrent()

```
void Digikam::SqueezedComboBox::setCurrent (
    const QString & itemText )
```

Set the current item to the one matching the given text.

## Parameters

<i>itemText</i>	the original (long version) of the item text
-----------------	--

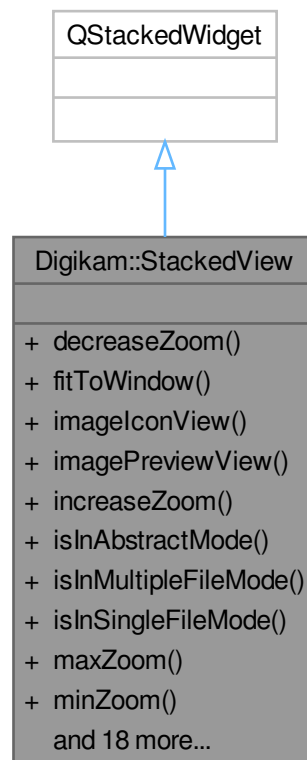
**6.1351.3.9 sizeHint()**

```
QSize Digikam::SqueezedComboBox::sizeHint ( ) const [override]
```

Sets the [sizeHint\(\)](#) of this widget.

**6.1352 Digikam::StackedView Class Reference**

Inheritance diagram for Digikam::StackedView:

**Public Types**

- enum **StackedViewMode** {  
**StackedViewModeFirst** = 0 , **IconViewMode** = 0 , **PreviewImageMode** = 1 , **WelcomePageMode** = 2 ,  
**TableViewMode** = 3 , **TrashViewMode** = 4 , **MapWidgetMode** = 5 , **MediaPlayerMode** = 6 ,  
**StackedViewModeLast** = 6 }



## Signals

- void **signalAddToExistingQueue** (int)
- void **signalDeleteItem** ()
- void **signalEscapePreview** ()
- void **signalGotoAlbumAndItem** (const [ItemInfo](#) &)
- void **signalGotoDateAndItem** (const [ItemInfo](#) &)
- void **signalGotoTagAndItem** (int)
- void **signalNextItem** ()
- void **signalPopupTagsView** ()
- void **signalPrevItem** ()
- void **signalViewModeChanged** ()
- void **signalZoomFactorChanged** (double)

## Public Member Functions

- void **decreaseZoom** ()
- void **fitToWindow** ()
- [DigikamItemView](#) \* **imageIconView** () const
- [ItemPreviewView](#) \* **imagePreviewView** () const
- void **increaseZoom** ()
- bool **isInAbstractMode** () const
- bool **isInMultipleFileMode** () const
- bool **isInSingleFileMode** () const
- bool **maxZoom** ()
- bool **minZoom** ()
- void **previewLoaded** ()
- void **setDockArea** (QMainWindow \*)
- void **setPreviewItem** (const [ItemInfo](#) &info=[ItemInfo](#)(), const [ItemInfo](#) &previous=[ItemInfo](#)(), const [ItemInfo](#) &next=[ItemInfo](#)())
- void **setViewMode** (const StackedViewMode mode, bool focus=false)
- void **setZoomFactor** (double z)
- void **setZoomFactorSnapped** (double z)
- [StackedView](#) (QWidget \*const parent=nullptr)
- [TableView](#) \* **tableView** () const
- [ItemThumbnailBar](#) \* **thumbBar** () const
- [ThumbBarDock](#) \* **thumbBarDock** () const
- void **toggleFitToWindowOr100** ()
- [TrashView](#) \* **trashView** () const
- StackedViewMode **viewMode** () const
- double **zoomFactor** ()
- double **zoomMax** ()
- double **zoomMin** ()
- void **zoomTo100Percents** ()

## 6.1352.1 Member Function Documentation

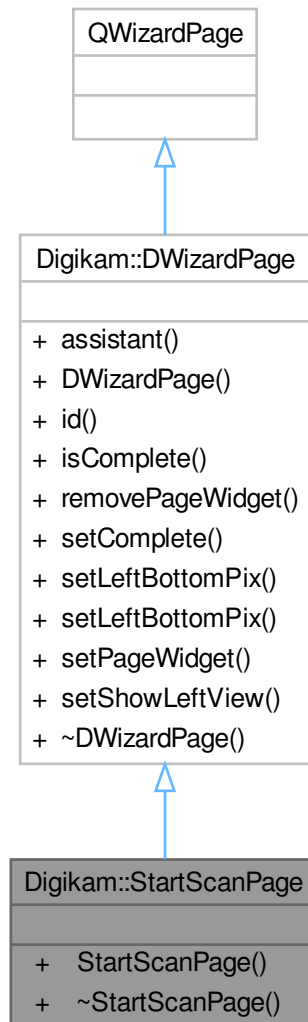
### 6.1352.1.1 isInSingleFileMode()

```
bool Digikam::StackedView::isInSingleFileMode ( ) const
```

Single-file mode is image preview or media player, multi-file is icon view or map, abstract modes do not handle files (welcome page)

## 6.1353 Digikam::StartScanPage Class Reference

Inheritance diagram for Digikam::StartScanPage:



### Public Member Functions

- **StartScanPage** (QWizard \*const dlg)

### Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)
- int **id** () const
- bool **isComplete** () const override

- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

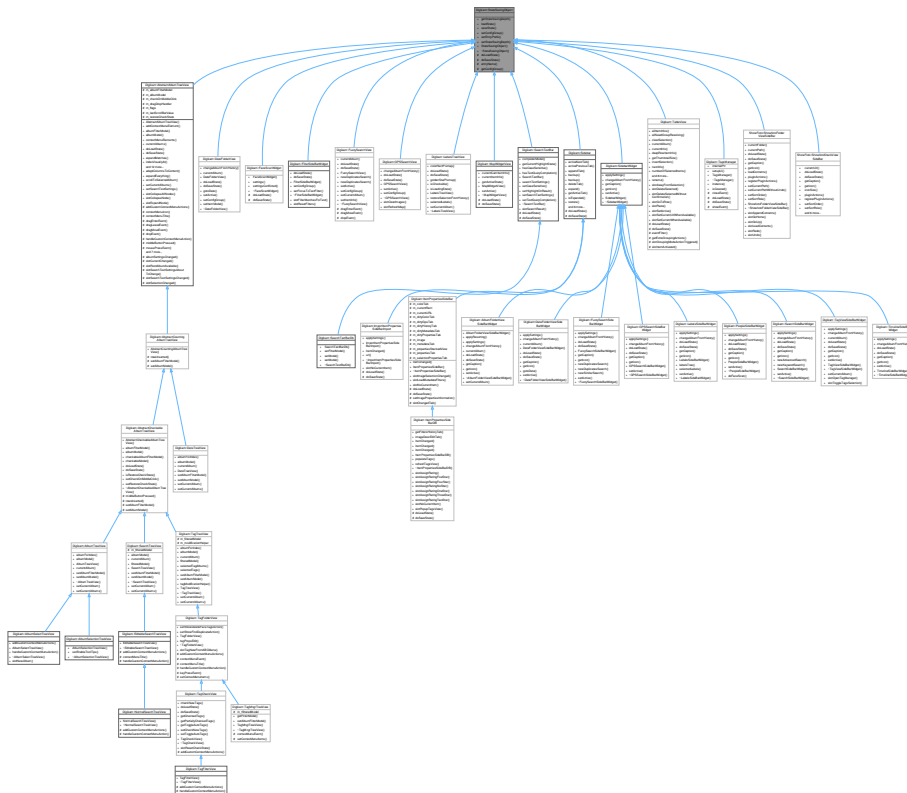
## 6.1354 Digikam::State Struct Reference

### Public Attributes

- bool **currentIndex** = false
- bool **expanded** = false
- bool **selected** = false

## 6.1355 Digikam::StateSavingObject Class Reference

Inheritance diagram for Digikam::StateSavingObject:



### Public Types

- enum **StateSavingDepth** { **INSTANCE** , **DIRECT\_CHILDREN** , **RECURSIVE** }

## Public Member Functions

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Protected Member Functions

- virtual void [doLoadState](#) ()=0
- virtual void [doSaveState](#) ()=0
- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

### 6.1355.1 Detailed Description

An interface-like class with utility methods and a general public interface to support state saving and restoring for objects via KConfig. Use this class as a Mixin.

The public interface for loading and saving state is implemented designed as template methods. To store or restore the state of a class, inherit from this class via multiple inheritance and implement [doLoadState\(\)](#) and [doSaveState\(\)](#). In these methods always use the protected method [getConfigGroup\(\)](#) to access a config group. Also always use the [entryName\(\)](#) method for generating keys in the config (for prefixes, see below).

Ensure that this class is inherited after a QObject-based class and pass "this" as constructor argument.

By default a config group based on Qt's object name of the class is used. This behaviour can be changed by setting a dedicated config group via [setConfigGroup\(\)](#). This is useful for to externally control the config group and shouldn't be used inside the implementing class.

Additionally to setting the config group, also a prefix for each config group entry can be defined via [setEntryPrefix\(\)](#). This may be useful if multiple instances of the same class shall be stored in the same config group or can generally be a good idea to make the config more readable and recognizable. By default this prefix is empty.

This class also supports recursive saving / loading invocations based on the QT object hierarchy. As default, calls to [loadState\(\)](#) or [saveState\(\)](#) only invoke the [doLoadState\(\)](#) or [doSaveState\(\)](#) method of the called instance. This behaviour can be changed with [setStateSavingDepth\(\)](#) to automatically call children of the instance. Various modes are supported as documented in [StateSavingDepth](#).

#### Author

jwienke

### 6.1355.2 Member Enumeration Documentation

#### 6.1355.2.1 StateSavingDepth

```
enum Digikam::StateSavingObject::StateSavingDepth
```

This enum defines the "depth" of the [StateSavingObject::loadState\(\)](#) and [StateSavingObject::saveState\(\)](#) methods.

## Enumerator

INSTANCE	Only the instance the saving / restoring was invoked on is saved / restored.
DIRECT_CHILDREN	The instance itself and all direct children of this instance implementing <a href="#">StateSavingObject</a> are saved / restored.
RECURSIVE	The instance and all children in the complete hierarchy are saved / restored.

## 6.1355.3 Constructor & Destructor Documentation

### 6.1355.3.1 StateSavingObject()

```
Digikam::StateSavingObject::StateSavingObject (
    QObject *const host ) [explicit]
```

Constructor. Must be called after any QObject-based constructor.

## Parameters

<i>host</i>	self-reference to access the object name, simply pass "this" as argument
-------------	--

### 6.1355.3.2 ~StateSavingObject()

```
Digikam::StateSavingObject::~~StateSavingObject ( ) [virtual]
```

Destructor.

## 6.1355.4 Member Function Documentation

### 6.1355.4.1 doLoadState()

```
virtual void Digikam::StateSavingObject::doLoadState ( ) [protected], [pure virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implemented in [Digikam::DateFolderView](#), [Digikam::FilterSideBarWidget](#), [Digikam::AlbumFolderViewSideBarWidget](#), [Digikam::DateFolderViewSideBarWidget](#), [Digikam::FuzzySearchSideBarWidget](#), [Digikam::GPSSearchSideBarWidget](#), [Digikam::LabelsSideBarWidget](#), [Digikam::PeopleSideBarWidget](#), [Digikam::SearchSideBarWidget](#), [Digikam::TagViewSideBarWidget](#), [Digikam::TimelineSideBarWidget](#), [Digikam::MapWidgetView](#), [Digikam::TableView](#), [Digikam::AbstractAlbumTreeView](#), [Digikam::AbstractCheckableAlbumTreeView](#), [Digikam::LabelsTreeView](#), [Digikam::ImportItemPropertiesSideBarImport](#), [Digikam::ItemPropertiesSideBar](#), [Digikam::ItemPropertiesSideBarDB](#), [Digikam::TagsManager](#), [Digikam::TagCheckView](#), [Digikam::Sidebar](#), [Digikam::SearchTextBar](#), [ShowFoto::ShowfotoFolderViewSideBar](#), [ShowFoto::ShowfotoStackViewSideBar](#), [Digikam::FaceScanWidget](#), [Digikam::FuzzySearchView](#), and [Digikam::GPSSearchView](#).

### 6.1355.4.2 doSaveState()

```
virtual void Digikam::StateSavingObject::doSaveState ( ) [protected], [pure virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implemented in [Digikam::DateFolderView](#), [Digikam::FilterSideBarWidget](#), [Digikam::AlbumFolderViewSideBarWidget](#), [Digikam::DateFolderViewSideBarWidget](#), [Digikam::FuzzySearchSideBarWidget](#), [Digikam::GPSSearchSideBarWidget](#), [Digikam::LabelsSideBarWidget](#), [Digikam::PeopleSideBarWidget](#), [Digikam::SearchSideBarWidget](#), [Digikam::TagViewSideBarWidget](#), [Digikam::TimelineSideBarWidget](#), [Digikam::MapWidgetView](#), [Digikam::TableView](#), [Digikam::AbstractAlbumTreeView](#), [Digikam::AbstractCheckableAlbumTreeView](#), [Digikam::LabelsTreeView](#), [Digikam::ImportItemPropertiesSideBarImport](#), [Digikam::ItemPropertiesSideBar](#), [Digikam::ItemPropertiesSideBarDB](#), [Digikam::TagsManager](#), [Digikam::TagCheckView](#), [Digikam::Sidebar](#), [Digikam::SearchTextBar](#), [ShowFoto::ShowfotoFolderViewSideBar](#), [ShowFoto::ShowfotoStackViewSideBar](#), [Digikam::FaceScanWidget](#), [Digikam::FuzzySearchView](#), and [Digikam::GPSSearchView](#).

### 6.1355.4.3 entryName()

```
QString Digikam::StateSavingObject::entryName (
    const QString & base ) const [protected]
```

Always use this method to create config group entry names. This allows to manipulate the entry keys externally by eg. setting a prefix.

#### Parameters

<i>base</i>	original name planned for the config group entry
-------------	--

#### Returns

entry name after manipulating it with externally set parameters

### 6.1355.4.4 getConfigGroup()

```
KConfigGroup Digikam::StateSavingObject::getConfigGroup ( ) const [protected]
```

Returns the config group that must be used for state saving and loading.

#### Returns

config group for state saving and loading

### 6.1355.4.5 getStateSavingDepth()

```
StateSavingObject::StateSavingDepth Digikam::StateSavingObject::getStateSavingDepth ( ) const
```

Returns the depth used for state saving or loading. Default is [StateSavingDepth::INSTANCE](#).

#### Returns

state saving / restoring depth

#### 6.1355.4.6 loadState()

```
void Digikam::StateSavingObject::loadState ( )
```

Invokes loading the class' state.

#### 6.1355.4.7 saveState()

```
void Digikam::StateSavingObject::saveState ( )
```

Invokes saving the class' state.

#### 6.1355.4.8 setConfigGroup()

```
void Digikam::StateSavingObject::setConfigGroup (
    const KConfigGroup & group ) [virtual]
```

Sets a dedicated config group that will be used to store and reload the state from. If this method is not called, a group based on the object name is used.

You can re-implement this method to pass the group set here to child objects. Don't forget to call this method in your implementation.

##### Parameters

<i>group</i>	config group to use for state saving and restoring
--------------	--

Reimplemented in [Digikam::DateFolderView](#), [Digikam::FilterSideBarWidget](#), [Digikam::FuzzySearchView](#), and [Digikam::GPSSearchView](#).

#### 6.1355.4.9 setEntryPrefix()

```
void Digikam::StateSavingObject::setEntryPrefix (
    const QString & prefix ) [virtual]
```

Define a prefix that will be used for every entry in the config group. The default prefix is empty.

You can re-implement this method to pass the prefix set here to child objects. Don't forget to call this method in your implementation.

##### Parameters

<i>prefix</i>	the prefix to use for the config entries
---------------	--

#### 6.1355.4.10 setStateSavingDepth()

```
void Digikam::StateSavingObject::setStateSavingDepth (
    const StateSavingDepth depth )
```

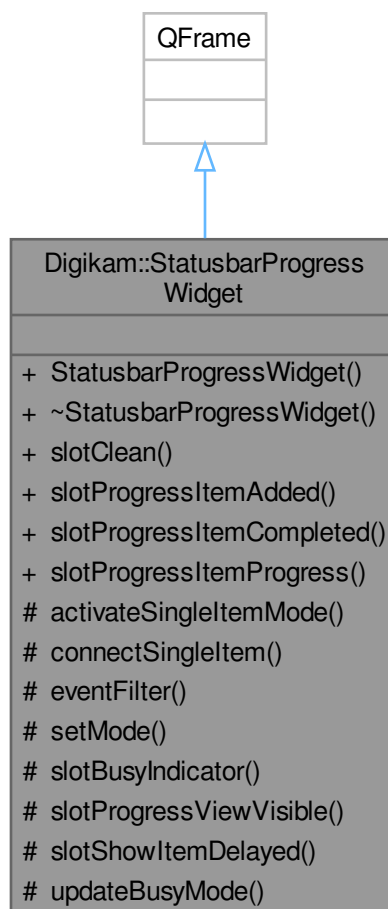
Sets the depth used for state saving or loading.

#### Parameters

<i>depth</i>	new depth to use
--------------	------------------

## 6.1356 Digikam::StatusBarProgressWidget Class Reference

Inheritance diagram for Digikam::StatusBarProgressWidget:



#### Public Slots

- void `slotClean` ()
- void `slotProgressItemAdded` ([ProgressItem](#) \*i)
- void `slotProgressItemCompleted` ([ProgressItem](#) \*i)
- void `slotProgressItemProgress` ([ProgressItem](#) \*i, unsigned int value)



**Public Member Functions**

- **StatusbarProgressWidget** ([ProgressView](#) \*const progressView, QWidget \*const parent, bool button=true)

**Protected Slots**

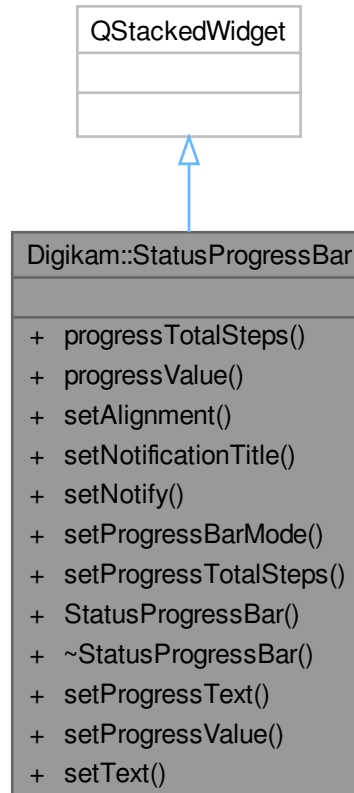
- void **slotBusyIndicator** ()
- void **slotProgressViewVisible** (bool)
- void **slotShowItemDelayed** ()
- void **updateBusyMode** ()

**Protected Member Functions**

- void **activateSingleItemMode** ()
- void **connectSingleItem** ()
- bool **eventFilter** (QObject \*, QEvent \*) override
- void **setMode** ()

**6.1357 Digikam::StatusProgressBar Class Reference**

Inheritance diagram for Digikam::StatusProgressBar:



### Public Types

- enum **StatusProgressBarMode** { **TextMode** = 0 , **ProgressBarMode** , **CancelProgressBarMode** }

### Public Slots

- void **setProgressText** (const QString &text)
- void **setProgressValue** (int v)
- void **setText** (const QString &text)

### Signals

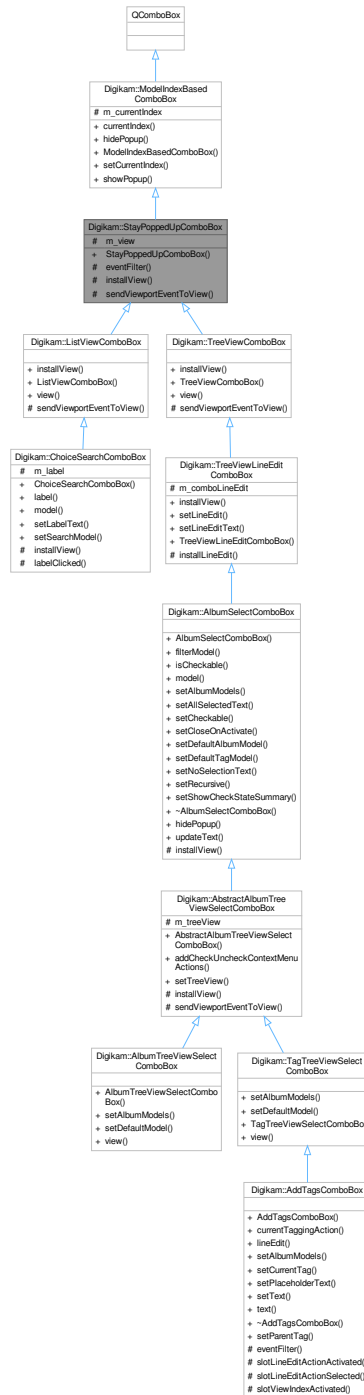
- void **signalCancelButtonPressed** ()

### Public Member Functions

- int **progressTotalSteps** () const
- int **progressValue** () const
- void **setAlignment** (Qt::Alignment a)
- void **setNotificationTitle** (const QString &title, const QIcon &icon)
- void **setNotify** (bool b)
- void **setProgressBarMode** (int mode, const QString &text=QString())
- void **setProgressTotalSteps** (int v)
- **StatusProgressBar** (QWidget \*const parent=nullptr)

## 6.1358 Digikam::StayPoppedUpComboBox Class Reference

Inheritance diagram for Digikam::StayPoppedUpComboBox:



### Public Member Functions

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex **currentIndex** () const
- void **hidePopup** () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

## Protected Member Functions

- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- void **installView** (QAbstractItemView \*view)
- virtual void **sendViewportEventToView** (QEvent \*e)=0

## Protected Attributes

- QAbstractItemView \* **m\_view** = nullptr

## Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex **m\_currentIndex**

## 6.1358.1 Constructor & Destructor Documentation

### 6.1358.1.1 StayPoppedUpComboBox()

```
Digikam::StayPoppedUpComboBox::StayPoppedUpComboBox (
    QWidget *const parent = nullptr ) [explicit]
```

This class provides an abstract QComboBox with a custom view (which is created by implementing subclasses) instead of the usual QListView. The Pop-up of the combo box will stay open after selecting an item; it will be closed by clicking outside, but not inside the widget. You need three steps: Construct the object, call `setModel()` with an appropriate QAbstractItemModel, then call [installView\(\)](#) to replace the standard combo box view with a view.

## 6.1358.2 Member Function Documentation

### 6.1358.2.1 installView()

```
void Digikam::StayPoppedUpComboBox::installView (
    QAbstractItemView * view ) [protected]
```

Replace the standard combo box list view with the given view. The view will be set as the view of the combo box (including re-parenting) and be stored in the `m_view` variable.

### 6.1358.2.2 sendViewportEventToView()

```
virtual void Digikam::StayPoppedUpComboBox::sendViewportEventToView (  
    QEvent * e ) [protected], [pure virtual]
```

Implement in subclass: Send the given event to the viewportEvent() method of m\_view. This method is protected for a usual QAbstractItemView. You can override, pass a view, and call parent implementation. The existing view will be used. You must then also reimplement sendViewportEventToView.

Implemented in [Digikam::AbstractAlbumTreeViewSelectComboBox](#), [Digikam::TreeViewComboBox](#), and [Digikam::ListViewComboBox](#).

## 6.1359 Digikam::StretchFilter Class Reference

Inheritance diagram for Digikam::StretchFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- void [readParameters](#) (const [FilterAction](#) &action) override
- [StretchFilter](#) ([DImg](#) \*const orgImage, const [DImg](#) \*const reflmage, [QObject](#) \*const parent=nullptr)
- [StretchFilter](#) ([QObject](#) \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, [QObject](#) \*const parent, const [QString](#) &name=[QString](#)())
- [DImgThreadedFilter](#) ([QObject](#) \*const parent=nullptr, const [QString](#) &name=[QString](#)())
- const [QString](#) & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- [QList](#)< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual [QString](#) [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const [QString](#) &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual [QList](#)< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) ([QObject](#) \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- [QThread::Priority](#) [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static [QString](#) [DisplayableName](#) ()
- static [QString](#) [FilterIdentifier](#) ()
- static [QList](#)< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false



## 6.1359.1 Member Function Documentation

### 6.1359.1.1 filterAction()

```
FilterAction Digikam::StretchFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1359.1.2 filterIdentifier()

```
QString Digikam::StretchFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

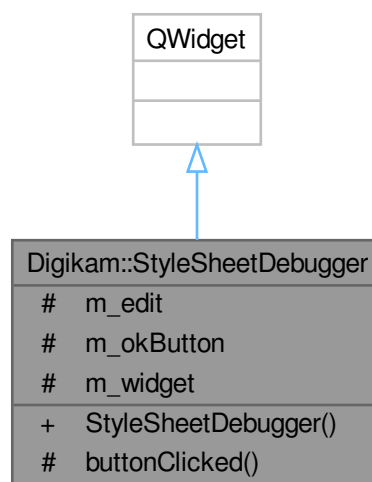
### 6.1359.1.3 readParameters()

```
void Digikam::StretchFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1360 Digikam::StyleSheetDebugger Class Reference

Inheritance diagram for Digikam::StyleSheetDebugger:



## Public Member Functions

- [StyleSheetDebugger](#) (QWidget \*const object)

## Protected Slots

- void **buttonClicked** ()

## Protected Attributes

- QTextEdit \* **m\_edit** = nullptr
- QPushButton \* **m\_okButton** = nullptr
- QWidget \* **m\_widget** = nullptr

## 6.1360.1 Constructor & Destructor Documentation

### 6.1360.1.1 StyleSheetDebugger()

```
Digikam::StyleSheetDebugger::StyleSheetDebugger (  
    QWidget *const object ) [explicit]
```

This widget is for development purpose only: It allows the developer to change the style sheet on a widget dynamically. If you want to develop or debug the stylesheet on your widget, add temporary code: `new StyleSheetDebugger(myWidget);` That's all. Change the style sheet by editing it and pressing Ok.

## 6.1361 Digikam::SubjectData Class Reference

### Public Member Functions

- **SubjectData** (const QString &n, const QString &m, const QString &d)

### Public Attributes

- QString **detail**  
*English and Detail Name of subject.*
- QString **matter**  
*English and Matter Name of subject.*
- QString **name**  
*English and Name of subject.*

## 6.1362 Digikam::SubjectEdit Class Reference

Inheritance diagram for Digikam::SubjectEdit:



### Public Member Functions

- **SubjectEdit** (QWidget \*const parent)

## Public Member Functions inherited from [Digikam::SubjectWidget](#)

- void **setSubjectsList** (const QStringList &list)
- QStringList **subjectsList** () const
- **SubjectWidget** (QWidget \*const parent, bool sizesLimited=false)

## Additional Inherited Members

## Signals inherited from [Digikam::SubjectWidget](#)

- void **signalModified** ()

## Protected Slots inherited from [Digikam::SubjectWidget](#)

- virtual void **slotAddSubject** ()
- virtual void **slotDelSubject** ()
- virtual void **slotEditOptionChanged** (int)
- virtual void **slotRefChanged** ()
- virtual void **slotRepSubject** ()
- virtual void **slotSubjectSelectionChanged** ()
- virtual void **slotSubjectsToggled** (bool)

## Protected Member Functions inherited from [Digikam::SubjectWidget](#)

- virtual QString **buildSubject** () const
- virtual bool **loadSubjectCodesFromXML** (const QUrl &url)

## Protected Attributes inherited from [Digikam::SubjectWidget](#)

- [DTextEdit](#) \* **m\_detailEdit** = nullptr
- QString **m\_iprDefault**
- [QLineEdit](#) \* **m\_iprEdit** = nullptr
- [DTextEdit](#) \* **m\_matterEdit** = nullptr
- [DTextEdit](#) \* **m\_nameEdit** = nullptr
- [QLabel](#) \* **m\_note** = nullptr
- [QLineEdit](#) \* **m\_refEdit** = nullptr
- [QCheckBox](#) \* **m\_subjectsCheck** = nullptr

## 6.1363 Digikam::SubjectWidget Class Reference

Inheritance diagram for Digikam::SubjectWidget:



### Signals

- void **signalModified** ()

### Public Member Functions

- void **setSubjectsList** (const QStringList &list)
- QStringList **subjectsList** () const
- **SubjectWidget** (QWidget \*const parent, bool sizesLimited=false)

### Protected Slots

- virtual void **slotAddSubject** ()
- virtual void **slotDelSubject** ()
- virtual void **slotEditOptionChanged** (int)
- virtual void **slotRefChanged** ()
- virtual void **slotRepSubject** ()
- virtual void **slotSubjectSelectionChanged** ()
- virtual void **slotSubjectsToggled** (bool)

### Protected Member Functions

- virtual QString **buildSubject** () const
- virtual bool **loadSubjectCodesFromXML** (const QUrl &url)

### Protected Attributes

- [DTextEdit](#) \* **m\_detailEdit** = nullptr
- QString **m\_iprDefault**
- [QLineEdit](#) \* **m\_iprEdit** = nullptr
- [DTextEdit](#) \* **m\_matterEdit** = nullptr
- [DTextEdit](#) \* **m\_nameEdit** = nullptr
- [QLabel](#) \* **m\_note** = nullptr
- [QLineEdit](#) \* **m\_refEdit** = nullptr
- [QCheckBox](#) \* **m\_subjectsCheck** = nullptr

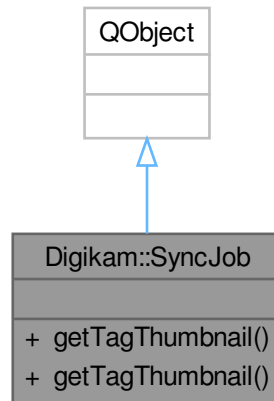
## 6.1364 Digikam::SubQueryBuilder Class Reference

### Public Member Functions

- QString **build** (enum SKey key, enum SOperator op, const QString &passedVal, QList< QVariant > \*bound↔ Values) const

## 6.1365 Digikam::SyncJob Class Reference

Inheritance diagram for Digikam::SyncJob:



### Static Public Member Functions

- static QPixmap **getTagThumbnail** (const QString &name, int size)
- static QPixmap [getTagThumbnail](#) (TAlbum \*const album)

### 6.1365.1 Member Function Documentation

#### 6.1365.1.1 getTagThumbnail()

```

QPixmap Digikam::SyncJob::getTagThumbnail (
    TAlbum *const album ) [static]
  
```

Load the image or icon for the tag thumbnail.

## 6.1366 Digikam::SystemSettings Class Reference

### Public Types

- enum [ProxyType](#) { [HttpProxy](#) = 0 , [Socks5Proxy](#) }
- This enum is used to specify the proxy that is used.*

### Public Member Functions

- void **saveSettings** ()
- **SystemSettings** (const QString &name)

## Public Attributes

- bool **enableAesthetic** = false
- bool **enableAutoTags** = false
- bool **enableFaceEngine** = false
- bool **enableHWTCnv** = false
- bool **enableHWVideo** = false
- bool **enableLogging** = false
- bool **enableOpenCL** = false
- bool **proxyAuth** = false
- QString **proxyPass**
- int **proxyPort** = 8080
- int **proxyType** = [HttpProxy](#)
- QString **proxyUrl**
- QString **proxyUser**
- bool **softwareOpenGL** = false
- QString **videoBackend** = QLatin1String("ffmpeg")

## 6.1366.1 Member Enumeration Documentation

### 6.1366.1.1 ProxyType

enum [Digikam::SystemSettings::ProxyType](#)

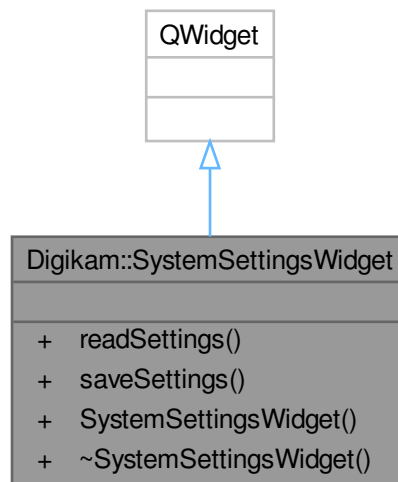
#### Enumerator

HttpProxy	Uses an Http proxy.
Socks5Proxy	Uses a Socks5 proxy.



## 6.1367 Digikam::SystemSettingsWidget Class Reference

Inheritance diagram for Digikam::SystemSettingsWidget:

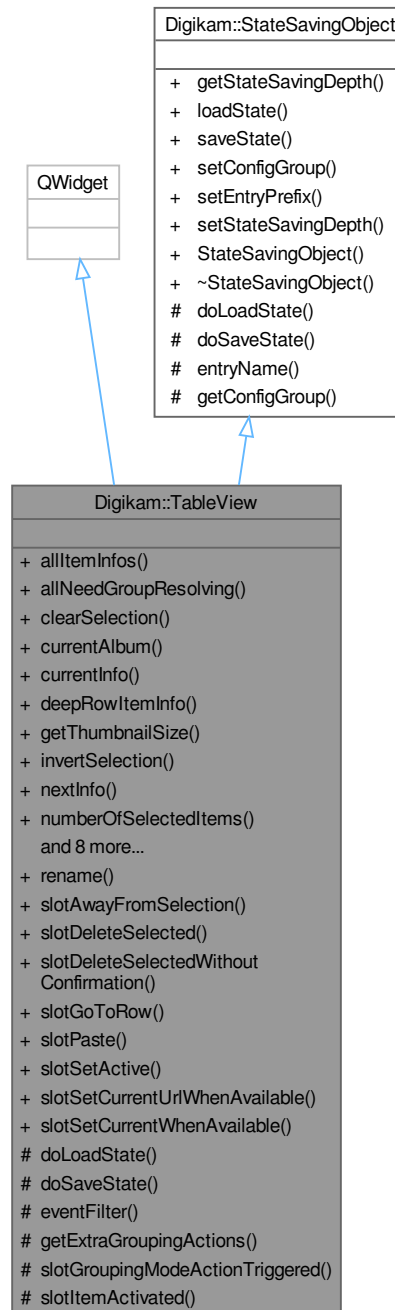


### Public Member Functions

- void **readSettings** ()
- void **saveSettings** ()
- **SystemSettingsWidget** (QWidget \*const parent)

## 6.1368 Digikam::TableView Class Reference

Inheritance diagram for Digikam::TableView:



### Public Slots

- void `rename()`
- void `slotAwayFromSelection()`

*Unselects the current selection and changes the current item.*

- void [slotDeleteSelected](#) (const [ItemViewUtilities::DeleteMode](#) deleteMode=[ItemViewUtilities::DeleteUseTrash](#))
- void [slotDeleteSelectedWithoutConfirmation](#) (const [ItemViewUtilities::DeleteMode](#) deleteMode=[ItemViewUtilities::DeleteUseTrash](#))
- void [slotGoToRow](#) (const int rowNumber, const bool relativeMove)
- void [slotPaste](#) ()
- void [slotSetActive](#) (const bool isActive)
- void [slotSetCurrentUrlWhenAvailable](#) (const [QUrl](#) &url)
- void [slotSetCurrentWhenAvailable](#) (const [qulonglong](#) id)

## Signals

- void [signalInsertSelectedToExistingQueue](#) (int queue)
- void [signalItemsChanged](#) ()
- void [signalPopupTagsView](#) ()
- void [signalPreviewRequested](#) (const [ItemInfo](#) &info)
- void [signalShowContextMenu](#) ([QContextMenuEvent](#) \*event, const [QList](#)< [QAction](#) \* > &actions)
- void [signalShowContextMenuOnInfo](#) ([QContextMenuEvent](#) \*event, const [ItemInfo](#) &info, const [QList](#)< [QAction](#) \* > &actions, [ItemFilterModel](#) \*filterModel=nullptr)
- void [signalZoomInStep](#) ()
- void [signalZoomOutStep](#) ()

## Public Member Functions

- [ItemInfoList](#) [allItemInfos](#) (bool grouping=false) const
- bool [allNeedGroupResolving](#) (const [OperationType](#) type) const
- void [clearSelection](#) ()
- [Album](#) \* [currentAlbum](#) () const
- [ItemInfo](#) [currentInfo](#) () const
- [ItemInfo](#) [deepRowItemInfo](#) (const int rowNumber, const bool relative) const
- [ThumbnailSize](#) [getThumbnailSize](#) () const
- void [invertSelection](#) ()
- [ItemInfo](#) [nextInfo](#) () const
- int [numberOfSelectedItems](#) () const
- [ItemInfo](#) [previousInfo](#) () const
- void [selectAll](#) ()
- [ItemInfoList](#) [selectedItemInfos](#) (bool grouping=false) const
- [ItemInfoList](#) [selectedItemInfosCurrentFirst](#) (bool grouping=false) const
- bool [selectedNeedGroupResolving](#) (const [OperationType](#) type) const
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &size)
- [TableView](#) ([QItemSelectionModel](#) \*const selectionModel, [DCategorizedSortFilterProxyModel](#) \*const imageFilterModel, [QWidget](#) \*const parent)

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const [KConfigGroup](#) &group)
- virtual void [setEntryPrefix](#) (const [QString](#) &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) ([QObject](#) \*const host)
- virtual [~StateSavingObject](#) ()

### Protected Slots

- void **slotGroupingModeActionTriggered** ()
- void **slotItemActivated** (const QModelIndex &tableViewIndex)

### Protected Member Functions

- void **doLoadState** () override
- void **doSaveState** () override
- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- QList< QAction \* > **getExtraGroupingActions** ()

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString **entryName** (const QString &base) const
- KConfigGroup **getConfigGroup** () const

### Additional Inherited Members

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## 6.1368.1 Member Function Documentation

### 6.1368.1.1 doLoadState()

```
void Digikam::TableView::doLoadState ( ) [override], [protected], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1368.1.2 doSaveState()

```
void Digikam::TableView::doSaveState ( ) [override], [protected], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1368.1.3 invertSelection()

```
void Digikam::TableView::invertSelection ( )
```

## 6.1368.1.4 selectAll()

```
void Digikam::TableView::selectAll ( )
```

## 6.1368.1.5 slotAwayFromSelection

```
void Digikam::TableView::slotAwayFromSelection ( ) [slot]
```

## 6.1368.1.6 slotDeleteSelected

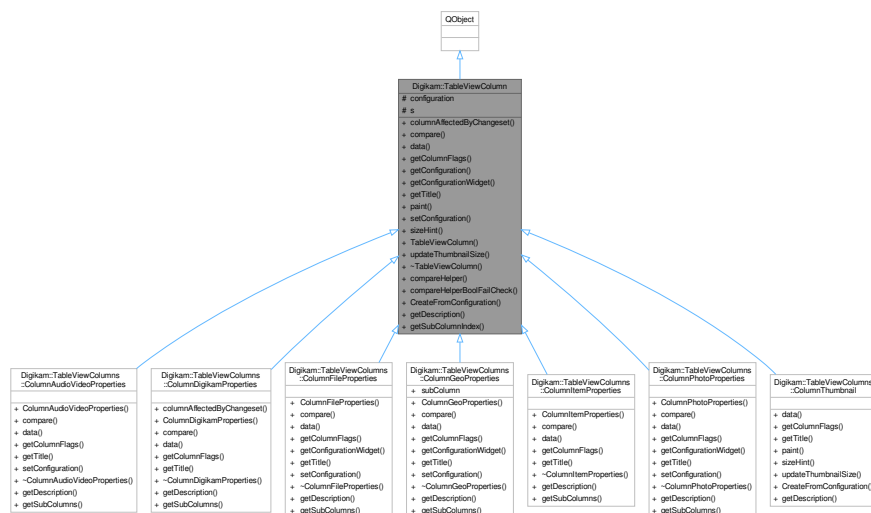
```
void Digikam::TableView::slotDeleteSelected (
    const ItemViewUtilities::DeleteMode deleteMode = ItemViewUtilities::DeleteUseTrash
) [slot]
```

## 6.1368.1.7 slotSetCurrentWhenAvailable

```
void Digikam::TableView::slotSetCurrentWhenAvailable (
    const qulonglong id ) [slot]
```

## 6.1369 Digikam::TableViewColumn Class Reference

Inheritance diagram for Digikam::TableViewColumn:



## Public Types

- enum **ColumnCompareResult** { **CmpEqual** = 0 , **CmpABiggerB** = 1 , **CmpALessB** = 2 }
- enum **ColumnFlag** { **ColumnNoFlags** = 0 , **ColumnCustomPainting** = 1 , **ColumnCustomSorting** = 2 , **ColumnHasConfigurationWidget** = 4 }

## Signals

- void **signalAllDataChanged** ()
- void **signalDataChanged** (const qlonglong imageId)

## Public Member Functions

- virtual bool **columnAffectedByChangeset** (const [ImageChangeset](#) &imageChangeset) const
- virtual ColumnCompareResult **compare** ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const
- virtual QVariant **data** ([TableViewModel::Item](#) \*const item, const int role) const
- virtual ColumnFlags **getColumnFlags** () const
- virtual [TableViewColumnConfiguration](#) **getConfiguration** () const
- virtual [TableViewColumnConfigurationWidget](#) \* **getConfigurationWidget** (QWidget \*const parentWidget) const
- virtual QString **getTitle** () const =0
- virtual bool **paint** (QPainter \*const painter, const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- virtual void **setConfiguration** (const [TableViewColumnConfiguration](#) &newConfiguration)
- virtual QSize **sizeHint** (const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- **TableViewColumn** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, QObject \*const parent=nullptr)
- virtual void **updateThumbnailSize** ()

## Static Public Member Functions

- template<class MyType >  
static ColumnCompareResult **compareHelper** (const MyType &A, const MyType &B)
- static bool **compareHelperBoolFailCheck** (const bool okA, const bool okB, ColumnCompareResult \*const result)
- template<typename columnClass >  
static bool **CreateFromConfiguration** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, QObject \*const parent)
- static [TableViewColumnDescription](#) **getDescription** ()
- template<typename columnClass >  
static bool **getSubColumnIndex** (const QString &subColumnId, typename columnClass::SubColumn \*const subColumn)

## Protected Attributes

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1369.1 Member Function Documentation

### 6.1369.1.1 columnAffectedByChangeset()

```
bool Digikam::TableViewColumn::columnAffectedByChangeset (
    const ImageChangeset & imageChangeset ) const [virtual]
```

Reimplemented in [Digikam::TableViewColumns::ColumnDigikamProperties](#).

### 6.1369.1.2 compare()

```
TableViewColumn::ColumnCompareResult Digikam::TableViewColumn::compare (
    TableViewModel::Item *const itemA,
    TableViewModel::Item *const itemB ) const [virtual]
```

This function should never be called, because subclasses have to do the comparison on their own. But it can not be pure, since then every subclass which does not do custom comparison would have to implement an empty stub.

Reimplemented in [Digikam::TableViewColumns::ColumnAudioVideoProperties](#), [Digikam::TableViewColumns::ColumnDigikamProperties](#), [Digikam::TableViewColumns::ColumnFileProperties](#), [Digikam::TableViewColumns::ColumnGeoProperties](#), [Digikam::TableViewColumns::ColumnItemProperties](#) and [Digikam::TableViewColumns::ColumnPhotoProperties](#).

### 6.1369.1.3 data()

```
QVariant Digikam::TableViewColumn::data (
    TableViewModel::Item *const item,
    const int role ) const [virtual]
```

Reimplemented in [Digikam::TableViewColumns::ColumnDigikamProperties](#), [Digikam::TableViewColumns::ColumnFileProperties](#), [Digikam::TableViewColumns::ColumnGeoProperties](#), [Digikam::TableViewColumns::ColumnItemProperties](#), and [Digikam::TableViewColumns::ColumnPhotoProperties](#).

### 6.1369.1.4 getColumnFlags()

```
TableViewColumn::ColumnFlags Digikam::TableViewColumn::getColumnFlags ( ) const [virtual]
```

Reimplemented in [Digikam::TableViewColumns::ColumnAudioVideoProperties](#).

### 6.1369.1.5 paint()

```
bool Digikam::TableViewColumn::paint (
    QPainter *const painter,
    const QStyleOptionViewItem & option,
    TableViewModel::Item *const item ) const [virtual]
```

Reimplemented in [Digikam::TableViewColumns::ColumnThumbnail](#).

### 6.1369.1.6 sizeHint()

```
QSize Digikam::TableViewColumn::sizeHint (
    const QStyleOptionViewItem & option,
    TableViewModel::Item *const item ) const [virtual]
```

Reimplemented in [Digikam::TableViewColumns::ColumnThumbnail](#).

### 6.1369.1.7 updateThumbnailSize()

```
void Digikam::TableViewColumn::updateThumbnailSize ( ) [virtual]
```

Reimplemented in [Digikam::TableViewColumns::ColumnThumbnail](#).

## 6.1370 Digikam::TableViewColumnConfiguration Class Reference

### Public Member Functions

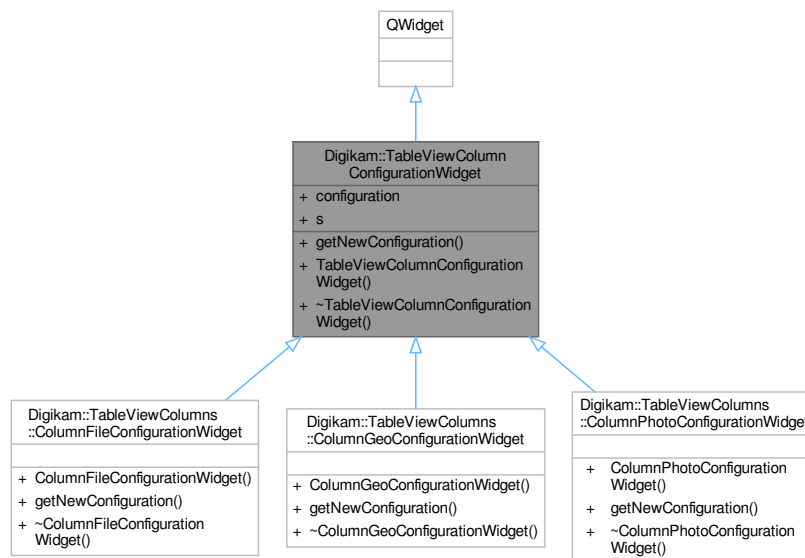
- QString **getSetting** (const QString &key, const QString &defaultValue=QString()) const
- void **loadSettings** (const KConfigGroup &configGroup)
- void **saveSettings** (KConfigGroup &configGroup) const
- **TableViewColumnConfiguration** (const QString &id=QString())

### Public Attributes

- QString **columnId**
- QHash< QString, QString > **columnSettings**

## 6.1371 Digikam::TableViewColumnConfigurationWidget Class Reference

Inheritance diagram for Digikam::TableViewColumnConfigurationWidget:



### Public Member Functions

- virtual [TableViewColumnConfiguration](#) **getNewConfiguration** ()=0
- **TableViewColumnConfigurationWidget** ([TableViewShared](#) \*const sharedObject, const [TableViewColumnConfiguration](#) &currentConfiguration, QWidget \*const parent=nullptr)

### Public Attributes

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr



## 6.1372 Digikam::TableViewColumnDescription Class Reference

### Public Types

- typedef QList< [TableViewColumnDescription](#) > **List**

### Public Member Functions

- void **addSetting** (const QString &key, const QString &value)
- void **addSubColumn** (const [TableViewColumnDescription](#) &subColumnDescription)
- [TableViewColumnDescription](#) **setIcon** (const QString &iconName)
- **TableViewColumnDescription** (const QString &id, const QString &title, const QString &setting←Key=QString(), const QString &settingValue=QString())
- [TableViewColumnConfiguration](#) **toConfiguration** () const

### Static Public Member Functions

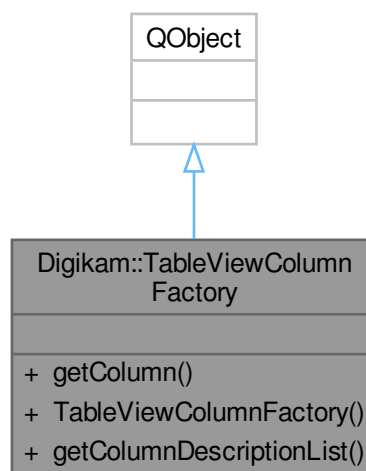
- static bool **FindInListById** (const [TableViewColumnDescription::List](#) &listToSearch, const QString &targetId, [TableViewColumnDescription](#) \*const resultDescription)

### Public Attributes

- QString **columnIcon**
- QString **columnId**
- QHash< QString, QString > **columnSettings**
- QString **columnTitle**
- QList< [TableViewColumnDescription](#) > **subColumns**

## 6.1373 Digikam::TableViewColumnFactory Class Reference

Inheritance diagram for Digikam::TableViewColumnFactory:



### Public Member Functions

- [TableViewColumn](#) \* **getColumn** (const [TableViewColumnConfiguration](#) &columnConfiguration)
- [TableViewColumnFactory](#) ([TableViewShared](#) \*const tableViewShared, QWidget \*const parent)

### Static Public Member Functions

- static QList< [TableViewColumnDescription](#) > **getColumnDescriptionList** ()

## 6.1374 Digikam::TableViewColumnProfile Class Reference

### Public Member Functions

- void **loadSettings** (const KConfigGroup &configGroup)
- void **saveSettings** (KConfigGroup &configGroup)

### Public Attributes

- QList< [TableViewColumnConfiguration](#) > **columnConfigurationList**
- QByteArray **headerState**
- QString **name**

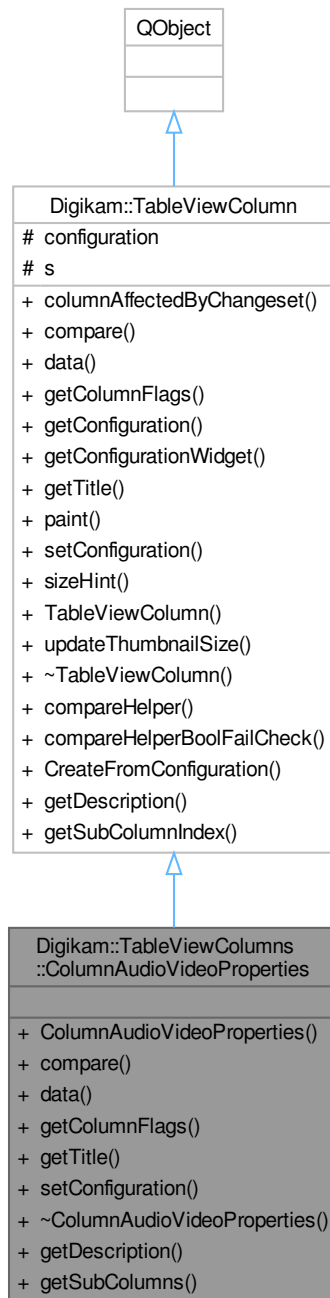
### 6.1374.1 Member Function Documentation

#### 6.1374.1.1 loadSettings()

```
void Digikam::TableViewColumnProfile::loadSettings (  
    const KConfigGroup & configGroup )
```

## 6.1375 Digikam::TableViewColumns::ColumnAudioVideoProperties Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnAudioVideoProperties:



### Public Types

- enum **SubColumn** {  
**SubColumnAudioBitRate** = 0 , **SubColumnAudioChannelType** = 1 , **SubColumnAudioCodec** = 2 , **Sub**↔

```
ColumnDuration = 3 ,
SubColumnFrameRate = 4 , SubColumnVideoCodec = 5 }
```

## Public Types inherited from [Digikam::TableViewColumn](#)

- enum **ColumnCompareResult** { **CmpEqual** = 0 , **CmpABiggerB** = 1 , **CmpALessB** = 2 }
- enum **ColumnFlag** { **ColumnNoFlags** = 0 , **ColumnCustomPainting** = 1 , **ColumnCustomSorting** = 2 , **ColumnHasConfigurationWidget** = 4 }

## Public Member Functions

- **ColumnAudioVideoProperties** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, const SubColumn pSubColumn, QObject \*const parent=nullptr)
- ColumnCompareResult **compare** ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const override
- QVariant **data** ([TableViewModel::Item](#) \*const item, const int role) const override
- ColumnFlags **getColumnFlags** () const override
- QString **getTitle** () const override
- void **setConfiguration** (const [TableViewColumnConfiguration](#) &newConfiguration) override

## Public Member Functions inherited from [Digikam::TableViewColumn](#)

- virtual bool **columnAffectedByChangeset** (const [ImageChangeset](#) &imageChangeset) const
- virtual [TableViewColumnConfiguration](#) **getConfiguration** () const
- virtual [TableViewColumnConfigurationWidget](#) \* **getConfigurationWidget** (QWidget \*const parentWidget) const
- virtual bool **paint** (QPainter \*const painter, const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- virtual QSize **sizeHint** (const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- **TableViewColumn** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, QObject \*const parent=nullptr)
- virtual void **updateThumbnailSize** ()

## Static Public Member Functions

- static [TableViewColumnDescription](#) **getDescription** ()
- static QStringList **getSubColumns** ()

## Static Public Member Functions inherited from [Digikam::TableViewColumn](#)

- template<class MyType >  
static ColumnCompareResult **compareHelper** (const MyType &A, const MyType &B)
- static bool **compareHelperBoolFailCheck** (const bool okA, const bool okB, ColumnCompareResult \*const result)
- template<typename columnClass >  
static bool **CreateFromConfiguration** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, QObject \*const parent)
- static [TableViewColumnDescription](#) **getDescription** ()
- template<typename columnClass >  
static bool **getSubColumnIndex** (const QString &subColumnId, typename columnClass::SubColumn \*const subColumn)

## Additional Inherited Members

## Signals inherited from [Digikam::TableViewColumn](#)

- void **signalAllDataChanged** ()
- void **signalDataChanged** (const qlonglong imageId)

## Protected Attributes inherited from [Digikam::TableViewColumn](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1375.1 Member Function Documentation

### 6.1375.1.1 compare()

```
TableViewColumn::ColumnCompareResult Digikam::TableViewColumns::ColumnAudioVideoProperties↔
::compare (
    TableViewModel::Item *const itemA,
    TableViewModel::Item *const itemB ) const [override], [virtual]
```

This function should never be called, because subclasses have to do the comparison on their own. But it can not be pure, since then every subclass which does not do custom comparison would have to implement an empty stub.

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1375.1.2 data()

```
QVariant Digikam::TableViewColumns::ColumnAudioVideoProperties::data (
    TableViewModel::Item *const item,
    const int role ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1375.1.3 getColumnFlags()

```
TableViewColumn::ColumnFlags Digikam::TableViewColumns::ColumnAudioVideoProperties::get↔
ColumnFlags ( ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1375.1.4 getTitle()

```
QString Digikam::TableViewColumns::ColumnAudioVideoProperties::getTitle ( ) const [override],
[virtual]
```

Implements [Digikam::TableViewColumn](#).

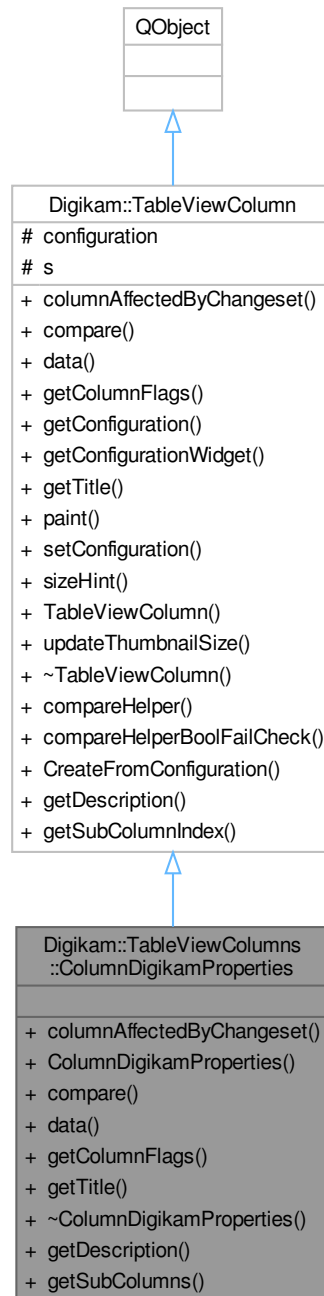
### 6.1375.1.5 setConfiguration()

```
void Digikam::TableViewColumns::ColumnAudioVideoProperties::setConfiguration (
    const TableViewColumnConfiguration & newConfiguration ) [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

## 6.1376 Digikam::TableViewColumns::ColumnDigikamProperties Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnDigikamProperties:



### Public Types

- enum **SubColumn** {  
**SubColumnRating** = 0 , **SubColumnPickLabel** = 1 , **SubColumnColorLabel** = 2 , **SubColumnTitle** = 3 ,  
**SubColumnCaption** = 4 , **SubColumnTags** = 5 }

## Public Types inherited from [Digikam::TableViewColumn](#)

- enum **ColumnCompareResult** { **CmpEqual** = 0 , **CmpABiggerB** = 1 , **CmpALessB** = 2 }
- enum **ColumnFlag** { **ColumnNoFlags** = 0 , **ColumnCustomPainting** = 1 , **ColumnCustomSorting** = 2 , **ColumnHasConfigurationWidget** = 4 }

## Public Member Functions

- bool [columnAffectedByChangeset](#) (const [ImageChangeset](#) &imageChangeset) const override
- **ColumnDigikamProperties** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, const [SubColumn](#) pSubColumn, [QObject](#) \*const parent=nullptr)
- [ColumnCompareResult](#) [compare](#) ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const override
- [QVariant](#) [data](#) ([TableViewModel::Item](#) \*const item, const int role) const override
- [ColumnFlags](#) [getColumnFlags](#) () const override
- [QString](#) [getTitle](#) () const override

## Public Member Functions inherited from [Digikam::TableViewColumn](#)

- virtual [TableViewColumnConfiguration](#) [getConfiguration](#) () const
- virtual [TableViewColumnConfigurationWidget](#) \* [getConfigurationWidget](#) ([QWidget](#) \*const parentWidget) const
- virtual bool [paint](#) ([QPainter](#) \*const painter, const [QStyleOptionViewItem](#) &option, [TableViewModel::Item](#) \*const item) const
- virtual void [setConfiguration](#) (const [TableViewColumnConfiguration](#) &newConfiguration)
- virtual [QSize](#) [sizeHint](#) (const [QStyleOptionViewItem](#) &option, [TableViewModel::Item](#) \*const item) const
- **TableViewColumn** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [QObject](#) \*const parent=nullptr)
- virtual void [updateThumbnailSize](#) ()

## Static Public Member Functions

- static [TableViewColumnDescription](#) [getDescription](#) ()
- static [QStringList](#) [getSubColumns](#) ()

## Static Public Member Functions inherited from [Digikam::TableViewColumn](#)

- template<class MyType >  
static [ColumnCompareResult](#) [compareHelper](#) (const MyType &A, const MyType &B)
- static bool [compareHelperBoolFailCheck](#) (const bool okA, const bool okB, [ColumnCompareResult](#) \*const result)
- template<typename columnClass >  
static bool [CreateFromConfiguration](#) ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, [QObject](#) \*const parent)
- static [TableViewColumnDescription](#) [getDescription](#) ()
- template<typename columnClass >  
static bool [getSubColumnIndex](#) (const [QString](#) &subColumnId, typename columnClass::SubColumn \*const subColumn)



## Additional Inherited Members

## Signals inherited from [Digikam::TableViewColumn](#)

- void **signalAllDataChanged** ()
- void **signalDataChanged** (const qlonglong imageId)

## Protected Attributes inherited from [Digikam::TableViewColumn](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1376.1 Member Function Documentation

### 6.1376.1.1 columnAffectedByChangeset()

```
bool Digikam::TableViewColumns::ColumnDigikamProperties::columnAffectedByChangeset (
    const ImageChangeset & imageChangeset ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1376.1.2 compare()

```
TableViewColumn::ColumnCompareResult Digikam::TableViewColumns::ColumnDigikamProperties↔
::compare (
    TableViewModel::Item *const itemA,
    TableViewModel::Item *const itemB ) const [override], [virtual]
```

This function should never be called, because subclasses have to do the comparison on their own. But it can not be pure, since then every subclass which does not do custom comparison would have to implement an empty stub.

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1376.1.3 data()

```
QVariant Digikam::TableViewColumns::ColumnDigikamProperties::data (
    TableViewModel::Item *const item,
    const int role ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1376.1.4 getColumnFlags()

```
TableViewColumn::ColumnFlags Digikam::TableViewColumns::ColumnDigikamProperties::getColumn↔
Flags ( ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

**6.1376.1.5 getDescription()**

```
TableViewColumnDescription Digikam::TableViewColumns::ColumnDigikamProperties::getDescription
( ) [static]
```

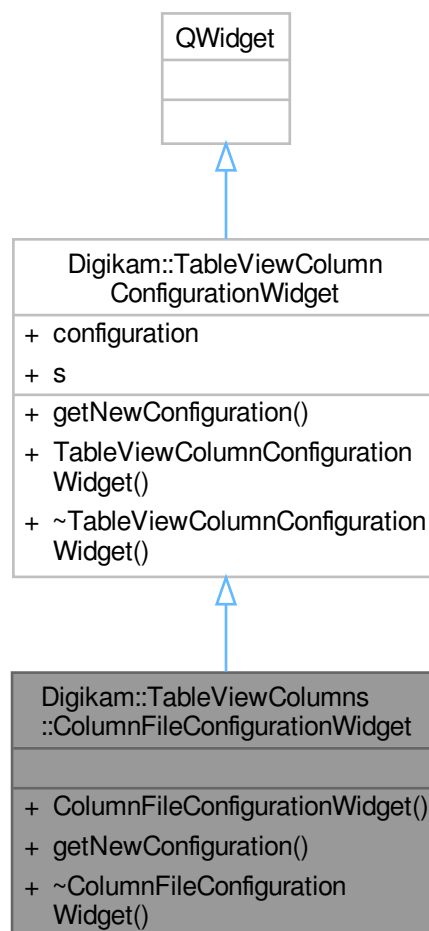
**6.1376.1.6 getTitle()**

```
QString Digikam::TableViewColumns::ColumnDigikamProperties::getTitle ( ) const [override],
[virtual]
```

Implements [Digikam::TableViewColumn](#).

## 6.1377 Digikam::TableViewColumns::ColumnFileConfigurationWidget Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnFileConfigurationWidget:



## Public Member Functions

- **ColumnFileConfigurationWidget** ([TableViewShared](#) \*const sharedObject, const [TableViewColumnConfiguration](#) &columnConfiguration, [QWidget](#) \*const parentWidget)
- [TableViewColumnConfiguration](#) `getNewConfiguration` () override

## Public Member Functions inherited from [Digikam::TableViewColumnConfigurationWidget](#)

- **TableViewColumnConfigurationWidget** ([TableViewShared](#) \*const sharedObject, const [TableViewColumnConfiguration](#) &currentConfiguration, [QWidget](#) \*const parent=nullptr)

## Additional Inherited Members

## Public Attributes inherited from [Digikam::TableViewColumnConfigurationWidget](#)

- [TableViewColumnConfiguration](#) `configuration`
- [TableViewShared](#) \*const `s` = nullptr

## 6.1377.1 Member Function Documentation

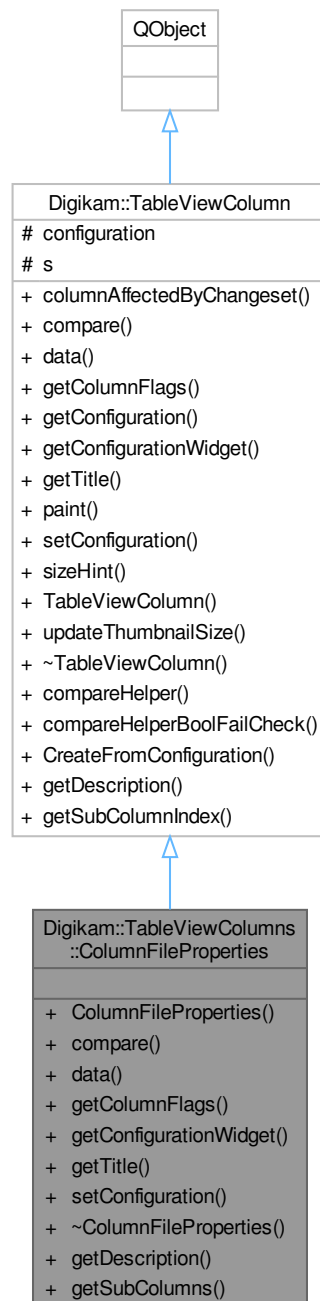
### 6.1377.1.1 `getNewConfiguration()`

[TableViewColumnConfiguration](#) `Digikam::TableViewColumns::ColumnFileConfigurationWidget::getNewConfiguration` ( ) [override], [virtual]

Implements [Digikam::TableViewColumnConfigurationWidget](#).

## 6.1378 Digikam::TableViewColumns::ColumnFileProperties Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnFileProperties:



### Public Types

- enum `SubColumn` { `SubColumnName` = 0 , `SubColumnFilePath` = 1 , `SubColumnSize` = 2 , `SubColumnLastModified` = 3 }

## Public Types inherited from [Digikam::TableViewColumn](#)

- enum **ColumnCompareResult** { **CmpEqual** = 0 , **CmpABiggerB** = 1 , **CmpALessB** = 2 }
- enum **ColumnFlag** { **ColumnNoFlags** = 0 , **ColumnCustomPainting** = 1 , **ColumnCustomSorting** = 2 , **ColumnHasConfigurationWidget** = 4 }

## Public Member Functions

- **ColumnFileProperties** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, const [SubColumn](#) pSubColumn, [QObject](#) \*const parent=nullptr)
- [ColumnCompareResult](#) **compare** ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const override
- [QVariant](#) **data** ([TableViewModel::Item](#) \*const item, const int role) const override
- [ColumnFlags](#) **getColumnFlags** () const override
- [TableViewColumnConfigurationWidget](#) \* **getConfigurationWidget** ([QWidget](#) \*const parentWidget) const override
- [QString](#) **getTitle** () const override
- void **setConfiguration** (const [TableViewColumnConfiguration](#) &newConfiguration) override

## Public Member Functions inherited from [Digikam::TableViewColumn](#)

- virtual bool **columnAffectedByChangeset** (const [ImageChangeset](#) &imageChangeset) const
- virtual [TableViewColumnConfiguration](#) **getConfiguration** () const
- virtual bool **paint** ([QPainter](#) \*const painter, const [QStyleOptionViewItem](#) &option, [TableViewModel::Item](#) \*const item) const
- virtual [QSize](#) **sizeHint** (const [QStyleOptionViewItem](#) &option, [TableViewModel::Item](#) \*const item) const
- **TableViewColumn** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [QObject](#) \*const parent=nullptr)
- virtual void **updateThumbnailSize** ()

## Static Public Member Functions

- static [TableViewColumnDescription](#) **getDescription** ()
- static [QStringList](#) **getSubColumns** ()

## Static Public Member Functions inherited from [Digikam::TableViewColumn](#)

- template<class MyType >  
static [ColumnCompareResult](#) **compareHelper** (const MyType &A, const MyType &B)
- static bool **compareHelperBoolFailCheck** (const bool okA, const bool okB, [ColumnCompareResult](#) \*const result)
- template<typename columnClass >  
static bool **CreateFromConfiguration** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, [QObject](#) \*const parent)
- static [TableViewColumnDescription](#) **getDescription** ()
- template<typename columnClass >  
static bool **getSubColumnIndex** (const [QString](#) &subColumnId, typename columnClass::SubColumn \*const subColumn)

## Additional Inherited Members

### Signals inherited from [Digikam::TableViewColumn](#)

- void **signalAllDataChanged** ()
- void **signalDataChanged** (const qlonglong imageld)

### Protected Attributes inherited from [Digikam::TableViewColumn](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1378.1 Member Function Documentation

### 6.1378.1.1 compare()

```
TableViewColumn::ColumnCompareResult Digikam::TableViewColumns::ColumnFileProperties::compare
(
    TableViewModel::Item *const itemA,
    TableViewModel::Item *const itemB ) const [override], [virtual]
```

This function should never be called, because subclasses have to do the comparison on their own. But it can not be pure, since then every subclass which does not do custom comparison would have to implement an empty stub.

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1378.1.2 data()

```
QVariant Digikam::TableViewColumns::ColumnFileProperties::data (
    TableViewModel::Item *const item,
    const int role ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1378.1.3 getColumnFlags()

```
TableViewColumn::ColumnFlags Digikam::TableViewColumns::ColumnFileProperties::getColumnFlags (
) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1378.1.4 getConfigurationWidget()

```
TableViewColumnConfigurationWidget * Digikam::TableViewColumns::ColumnFileProperties::get↔
ConfigurationWidget (
    QWidget *const parentWidget ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

**6.1378.1.5 getTitle()**

```
QString Digikam::TableViewColumns::ColumnFileProperties::getTitle ( ) const [override], [virtual]
```

Implements [Digikam::TableViewColumn](#).

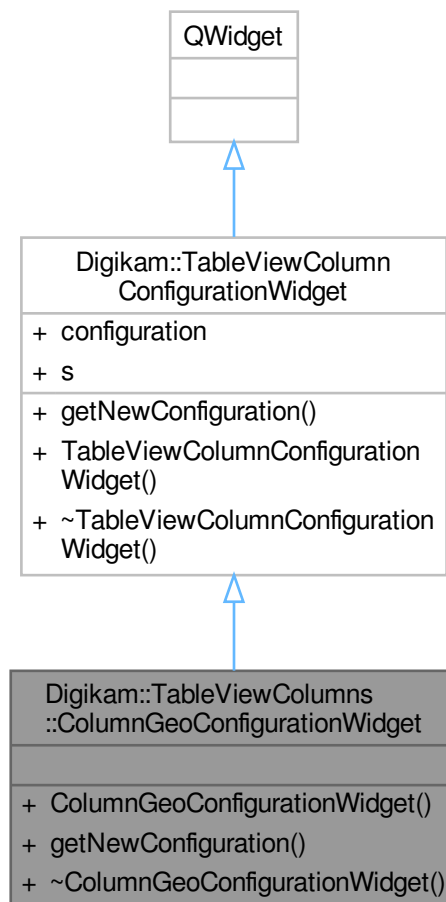
**6.1378.1.6 setConfiguration()**

```
void Digikam::TableViewColumns::ColumnFileProperties::setConfiguration (
    const TableViewColumnConfiguration & newConfiguration ) [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

## 6.1379 Digikam::TableViewColumns::ColumnGeoConfigurationWidget Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnGeoConfigurationWidget:



## Public Member Functions

- **ColumnGeoConfigurationWidget** ([TableViewShared](#) \*const sharedObject, const [TableViewColumnConfiguration](#) &columnConfiguration, [QWidget](#) \*const parentWidget)
- [TableViewColumnConfiguration](#) `getNewConfiguration` () override

## Public Member Functions inherited from [Digikam::TableViewColumnConfigurationWidget](#)

- **TableViewColumnConfigurationWidget** ([TableViewShared](#) \*const sharedObject, const [TableViewColumnConfiguration](#) &currentConfiguration, [QWidget](#) \*const parent=nullptr)

## Additional Inherited Members

## Public Attributes inherited from [Digikam::TableViewColumnConfigurationWidget](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1379.1 Member Function Documentation

### 6.1379.1.1 `getNewConfiguration()`

[TableViewColumnConfiguration](#) `Digikam::TableViewColumns::ColumnGeoConfigurationWidget::getNewConfiguration` ( ) [override], [virtual]

Implements [Digikam::TableViewColumnConfigurationWidget](#).



## 6.1380 Digikam::TableViewColumns::ColumnGeoProperties Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnGeoProperties:



### Public Types

- enum **SubColumn** { **SubColumnHasCoordinates** = 0 , **SubColumnCoordinates** = 1 , **SubColumnAltitude** = 2 }

## Public Types inherited from [Digikam::TableViewColumn](#)

- enum **ColumnCompareResult** { **CmpEqual** = 0 , **CmpABiggerB** = 1 , **CmpALessB** = 2 }
- enum **ColumnFlag** { **ColumnNoFlags** = 0 , **ColumnCustomPainting** = 1 , **ColumnCustomSorting** = 2 , **ColumnHasConfigurationWidget** = 4 }

## Public Member Functions

- **ColumnGeoProperties** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, const SubColumn pSubColumn, QObject \*const parent=nullptr)
- ColumnCompareResult **compare** ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const override
- QVariant **data** ([TableViewModel::Item](#) \*const item, const int role) const override
- ColumnFlags **getColumnFlags** () const override
- [TableViewColumnConfigurationWidget](#) \* **getConfigurationWidget** (QWidget \*const parentWidget) const override
- QString **getTitle** () const override
- void **setConfiguration** (const [TableViewColumnConfiguration](#) &newConfiguration) override

## Public Member Functions inherited from [Digikam::TableViewColumn](#)

- virtual bool **columnAffectedByChangeset** (const [ImageChangeset](#) &imageChangeset) const
- virtual [TableViewColumnConfiguration](#) **getConfiguration** () const
- virtual bool **paint** (QPainter \*const painter, const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- virtual QSize **sizeHint** (const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- **TableViewColumn** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, QObject \*const parent=nullptr)
- virtual void **updateThumbnailSize** ()

## Static Public Member Functions

- static [TableViewColumnDescription](#) **getDescription** ()
- static QStringList **getSubColumns** ()

## Static Public Member Functions inherited from [Digikam::TableViewColumn](#)

- template<class MyType >  
static ColumnCompareResult **compareHelper** (const MyType &A, const MyType &B)
- static bool **compareHelperBoolFailCheck** (const bool okA, const bool okB, ColumnCompareResult \*const result)
- template<typename columnClass >  
static bool **CreateFromConfiguration** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, QObject \*const parent)
- static [TableViewColumnDescription](#) **getDescription** ()
- template<typename columnClass >  
static bool **getSubColumnIndex** (const QString &subColumnId, typename columnClass::SubColumn \*const subColumn)

## Public Attributes

- enum [Digikam::TableViewColumns::ColumnGeoProperties::SubColumn](#) **subColumn**

## Additional Inherited Members

### Signals inherited from [Digikam::TableViewColumn](#)

- void **signalAllDataChanged** ()
- void **signalDataChanged** (const qlonglong imageId)

### Protected Attributes inherited from [Digikam::TableViewColumn](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1380.1 Member Function Documentation

### 6.1380.1.1 compare()

```
TableViewColumn::ColumnCompareResult Digikam::TableViewColumns::ColumnGeoProperties::compare (
    TableViewModel::Item *const itemA,
    TableViewModel::Item *const itemB ) const [override], [virtual]
```

This function should never be called, because subclasses have to do the comparison on their own. But it can not be pure, since then every subclass which does not do custom comparison would have to implement an empty stub.

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1380.1.2 data()

```
QVariant Digikam::TableViewColumns::ColumnGeoProperties::data (
    TableViewModel::Item *const item,
    const int role ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1380.1.3 getColumnFlags()

```
TableViewColumn::ColumnFlags Digikam::TableViewColumns::ColumnGeoProperties::getColumnFlags (
) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1380.1.4 getConfigurationWidget()

```
TableViewColumnConfigurationWidget * Digikam::TableViewColumns::ColumnGeoProperties::get←
ConfigurationWidget (
    QWidget *const parentWidget ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

**6.1380.1.5 getTitle()**

```
QString Digikam::TableViewColumns::ColumnGeoProperties::getTitle ( ) const [override], [virtual]
```

Implements [Digikam::TableViewColumn](#).

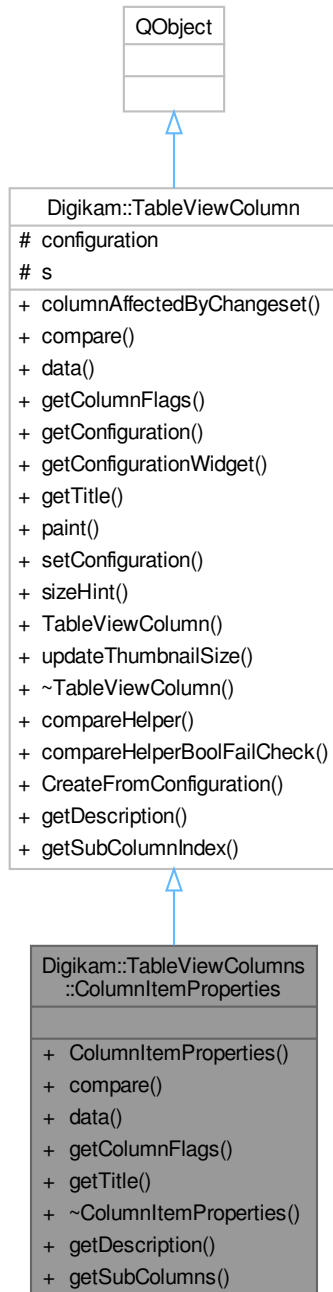
**6.1380.1.6 setConfiguration()**

```
void Digikam::TableViewColumns::ColumnGeoProperties::setConfiguration (
    const TableViewColumnConfiguration & newConfiguration ) [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

## 6.1381 Digikam::TableViewColumns::ColumnItemProperties Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnItemProperties:



### Public Types

- enum `SubColumn` {  
`SubColumnWidth = 0` , `SubColumnHeight = 1` , `SubColumnDimensions = 2` , `SubColumnPixelCount =`

```

3 ,
SubColumnBitDepth = 4 , SubColumnColorMode = 5 , SubColumnType = 6 , SubColumnCreationDate↵
DateTime = 7 ,
SubColumnDigitizationDateTime = 8 , SubColumnAspectRatio = 9 , SubColumnSimilarity = 10 }

```

## Public Types inherited from [Digikam::TableViewColumn](#)

- enum **ColumnCompareResult** { **CmpEqual** = 0 , **CmpABiggerB** = 1 , **CmpALessB** = 2 }
- enum **ColumnFlag** { **ColumnNoFlags** = 0 , **ColumnCustomPainting** = 1 , **ColumnCustomSorting** = 2 , **ColumnHasConfigurationWidget** = 4 }

## Public Member Functions

- **ColumnItemProperties** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, const SubColumn pSubColumn, QObject \*const parent=nullptr)
- ColumnCompareResult **compare** ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const override
- QVariant **data** ([TableViewModel::Item](#) \*const item, const int role) const override
- ColumnFlags **getColumnFlags** () const override
- QString **getTitle** () const override

## Public Member Functions inherited from [Digikam::TableViewColumn](#)

- virtual bool **columnAffectedByChangeset** (const [ImageChangeset](#) &imageChangeset) const
- virtual [TableViewColumnConfiguration](#) **getConfiguration** () const
- virtual [TableViewColumnConfigurationWidget](#) \* **getConfigurationWidget** (QWidget \*const parentWidget) const
- virtual bool **paint** (QPainter \*const painter, const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- virtual void **setConfiguration** (const [TableViewColumnConfiguration](#) &newConfiguration)
- virtual QSize **sizeHint** (const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- **TableViewColumn** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, QObject \*const parent=nullptr)
- virtual void **updateThumbnailSize** ()

## Static Public Member Functions

- static [TableViewColumnDescription](#) **getDescription** ()
- static QStringList **getSubColumns** ()

## Static Public Member Functions inherited from [Digikam::TableViewColumn](#)

- template<class MyType >  
static ColumnCompareResult **compareHelper** (const MyType &A, const MyType &B)
- static bool **compareHelperBoolFailCheck** (const bool okA, const bool okB, ColumnCompareResult \*const result)
- template<typename columnClass >  
static bool **CreateFromConfiguration** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, QObject \*const parent)
- static [TableViewColumnDescription](#) **getDescription** ()
- template<typename columnClass >  
static bool **getSubColumnIndex** (const QString &subColumnId, typename columnClass::SubColumn \*const subColumn)

## Additional Inherited Members

## Signals inherited from [Digikam::TableViewColumn](#)

- void **signalAllDataChanged** ()
- void **signalDataChanged** (const qlonglong imageId)

## Protected Attributes inherited from [Digikam::TableViewColumn](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1381.1 Member Function Documentation

### 6.1381.1.1 compare()

```
TableViewColumn::ColumnCompareResult Digikam::TableViewColumns::ColumnItemProperties::compare (
    TableViewModel::Item *const itemA,
    TableViewModel::Item *const itemB ) const [override], [virtual]
```

This function should never be called, because subclasses have to do the comparison on their own. But it can not be pure, since then every subclass which does not do custom comparison would have to implement an empty stub.

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1381.1.2 data()

```
QVariant Digikam::TableViewColumns::ColumnItemProperties::data (
    TableViewModel::Item *const item,
    const int role ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1381.1.3 getColumnFlags()

```
TableViewColumn::ColumnFlags Digikam::TableViewColumns::ColumnItemProperties::getColumnFlags (
) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

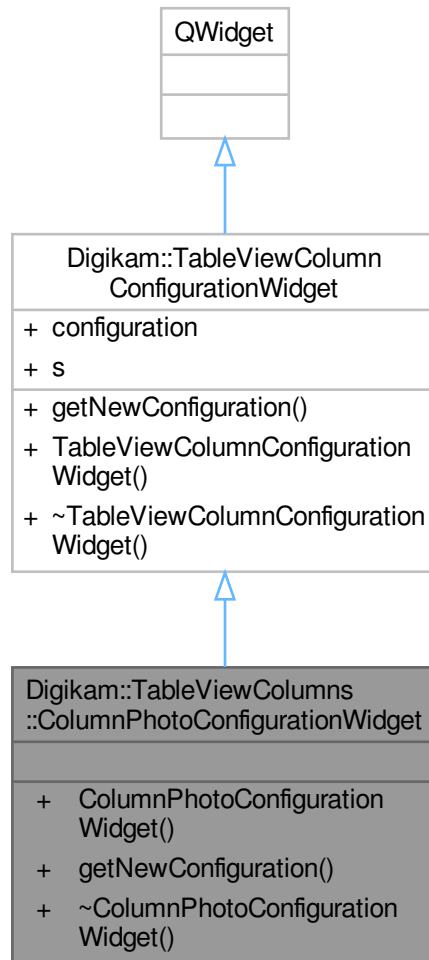
### 6.1381.1.4 getTitle()

```
QString Digikam::TableViewColumns::ColumnItemProperties::getTitle ( ) const [override], [virtual]
```

Implements [Digikam::TableViewColumn](#).

## 6.1382 Digikam::TableViewColumns::ColumnPhotoConfigurationWidget Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnPhotoConfigurationWidget:



### Public Member Functions

- **ColumnPhotoConfigurationWidget** (`TableViewShared *const sharedObject`, `const TableViewColumnConfiguration &columnConfiguration`, `QWidget *const parentWidget`)
- `TableViewColumnConfiguration` `getNewConfiguration ()` override

### Public Member Functions inherited from Digikam::TableViewColumnConfigurationWidget

- **TableViewColumnConfigurationWidget** (`TableViewShared *const sharedObject`, `const TableViewColumnConfiguration &currentConfiguration`, `QWidget *const parent=nullptr`)



## Additional Inherited Members

### Public Attributes inherited from [Digikam::TableViewColumnConfigurationWidget](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1382.1 Member Function Documentation

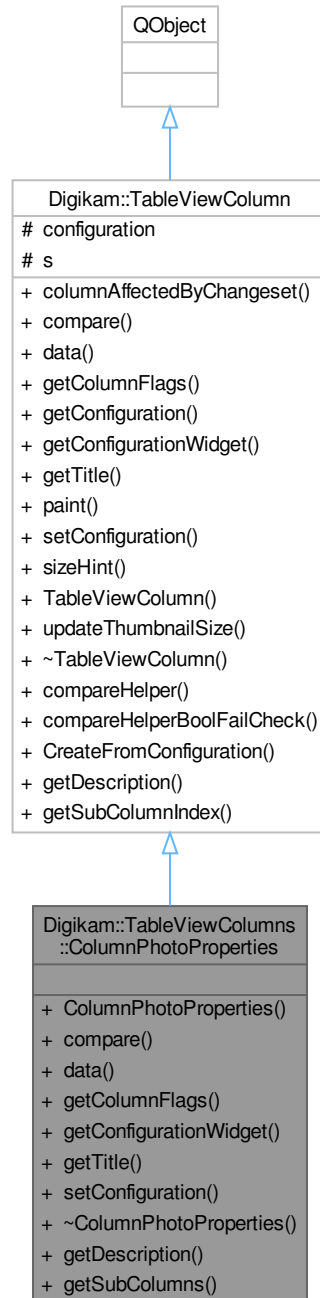
### 6.1382.1.1 [getNewConfiguration\(\)](#)

[TableViewColumnConfiguration](#) [Digikam::TableViewColumns::ColumnPhotoConfigurationWidget::getNewConfiguration](#) ( ) [override], [virtual]

Implements [Digikam::TableViewColumnConfigurationWidget](#).

## 6.1383 Digikam::TableViewColumns::ColumnPhotoProperties Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnPhotoProperties:



### Public Types

- enum **SubColumn** {  
**SubColumnCameraMaker** = 0 , **SubColumnCameraModel** = 1 , **SubColumnLens** = 2 , **SubColumnAperture** = 3 ,

**SubColumnFocal** = 4 , **SubColumnExposure** = 5 , **SubColumnSensitivity** = 6 , **SubColumnMode**↔  
**Program** = 7 ,  
**SubColumnFlash** = 8 , **SubColumnWhiteBalance** = 9 }

## Public Types inherited from [Digikam::TableViewColumn](#)

- enum **ColumnCompareResult** { **CmpEqual** = 0 , **CmpABiggerB** = 1 , **CmpALessB** = 2 }
- enum **ColumnFlag** { **ColumnNoFlags** = 0 , **ColumnCustomPainting** = 1 , **ColumnCustomSorting** = 2 , **ColumnHasConfigurationWidget** = 4 }

## Public Member Functions

- **ColumnPhotoProperties** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, const SubColumn pSubColumn, QObject \*const parent=nullptr)
- ColumnCompareResult **compare** ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const override
- QVariant **data** ([TableViewModel::Item](#) \*const item, const int role) const override
- ColumnFlags **getColumnFlags** () const override
- [TableViewColumnConfigurationWidget](#) \* **getConfigurationWidget** (QWidget \*const parentWidget) const override
- QString **getTitle** () const override
- void **setConfiguration** (const [TableViewColumnConfiguration](#) &newConfiguration) override

## Public Member Functions inherited from [Digikam::TableViewColumn](#)

- virtual bool **columnAffectedByChangeset** (const [ImageChangeset](#) &imageChangeset) const
- virtual [TableViewColumnConfiguration](#) **getConfiguration** () const
- virtual bool **paint** (QPainter \*const painter, const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- virtual QSize **sizeHint** (const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const
- **TableViewColumn** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &p↔  
Configuration, QObject \*const parent=nullptr)
- virtual void **updateThumbnailSize** ()

## Static Public Member Functions

- static [TableViewColumnDescription](#) **getDescription** ()
- static QStringList **getSubColumns** ()

## Static Public Member Functions inherited from [Digikam::TableViewColumn](#)

- template<class MyType >  
static ColumnCompareResult **compareHelper** (const MyType &A, const MyType &B)
- static bool **compareHelperBoolFailCheck** (const bool okA, const bool okB, ColumnCompareResult \*const result)
- template<typename columnClass >  
static bool **CreateFromConfiguration** ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, QObject \*const parent)
- static [TableViewColumnDescription](#) **getDescription** ()
- template<typename columnClass >  
static bool **getSubColumnIndex** (const QString &subColumnId, typename columnClass::SubColumn \*const subColumn)

## Additional Inherited Members

### Signals inherited from [Digikam::TableViewColumn](#)

- void **signalAllDataChanged** ()
- void **signalDataChanged** (const qlonglong imageld)

### Protected Attributes inherited from [Digikam::TableViewColumn](#)

- [TableViewColumnConfiguration](#) **configuration**
- [TableViewShared](#) \*const **s** = nullptr

## 6.1383.1 Member Function Documentation

### 6.1383.1.1 `compare()`

```
TableViewColumn::ColumnCompareResult Digikam::TableViewColumns::ColumnPhotoProperties::compare
(
    TableViewModel::Item *const itemA,
    TableViewModel::Item *const itemB ) const [override], [virtual]
```

This function should never be called, because subclasses have to do the comparison on their own. But it can not be pure, since then every subclass which does not do custom comparison would have to implement an empty stub.

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1383.1.2 `data()`

```
QVariant Digikam::TableViewColumns::ColumnPhotoProperties::data (
    TableViewModel::Item *const item,
    const int role ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1383.1.3 `getColumnFlags()`

```
TableViewColumn::ColumnFlags Digikam::TableViewColumns::ColumnPhotoProperties::getColumnFlags
( ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1383.1.4 `getConfigurationWidget()`

```
TableViewColumnConfigurationWidget * Digikam::TableViewColumns::ColumnPhotoProperties::get↔
ConfigurationWidget (
    QWidget *const parentWidget ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

### 6.1383.1.5 getTitle()

```
QString Digikam::TableViewColumns::ColumnPhotoProperties::getTitle ( ) const [override],  
[virtual]
```

Implements [Digikam::TableViewColumn](#).

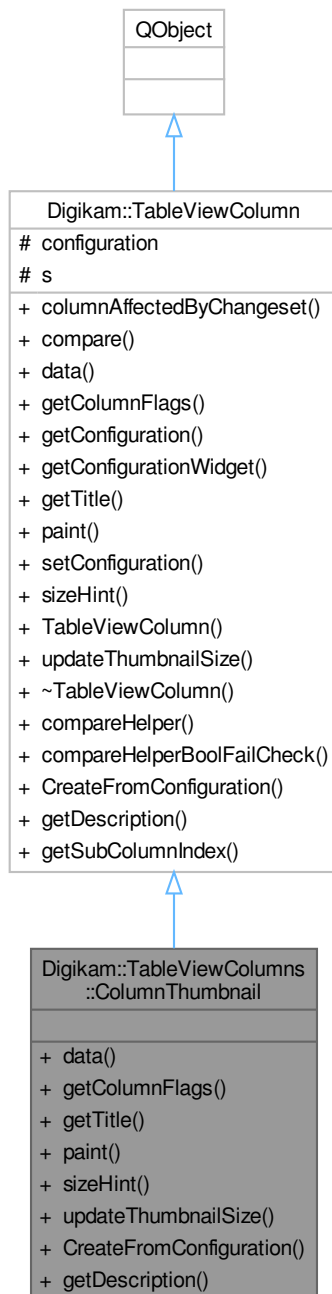
### 6.1383.1.6 setConfiguration()

```
void Digikam::TableViewColumns::ColumnPhotoProperties::setConfiguration (   
    const TableViewColumnConfiguration & newConfiguration ) [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

## 6.1384 Digikam::TableViewColumns::ColumnThumbnail Class Reference

Inheritance diagram for Digikam::TableViewColumns::ColumnThumbnail:



### Public Member Functions

- QVariant `data` (`TableViewModel::Item *const item`, `const int role`) `const` override
- ColumnFlags `getColumnFlags` () `const` override

- QString [getTitle](#) () const override
- bool [paint](#) (QPainter \*const painter, const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const override
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, [TableViewModel::Item](#) \*const item) const override
- void [updateThumbnailSize](#) () override

## Public Member Functions inherited from [Digikam::TableViewColumn](#)

- virtual bool [columnAffectedByChangeset](#) (const [ImageChangeset](#) &imageChangeset) const
- virtual ColumnCompareResult [compare](#) ([TableViewModel::Item](#) \*const itemA, [TableViewModel::Item](#) \*const itemB) const
- virtual [TableViewColumnConfiguration](#) [getConfiguration](#) () const
- virtual [TableViewColumnConfigurationWidget](#) \* [getConfigurationWidget](#) (QWidget \*const parentWidget) const
- virtual void [setConfiguration](#) (const [TableViewColumnConfiguration](#) &newConfiguration)
- [TableViewColumn](#) ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, QObject \*const parent=nullptr)

## Static Public Member Functions

- static bool [CreateFromConfiguration](#) ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, QWidget \*const parent)
- static [TableViewColumnDescription](#) [getDescription](#) ()

## Static Public Member Functions inherited from [Digikam::TableViewColumn](#)

- template<class MyType >  
static ColumnCompareResult [compareHelper](#) (const MyType &A, const MyType &B)
- static bool [compareHelperBoolFailCheck](#) (const bool okA, const bool okB, ColumnCompareResult \*const result)
- template<typename columnClass >  
static bool [CreateFromConfiguration](#) ([TableViewShared](#) \*const tableViewShared, const [TableViewColumnConfiguration](#) &pConfiguration, [TableViewColumn](#) \*\*const pNewColumn, QObject \*const parent)
- static [TableViewColumnDescription](#) [getDescription](#) ()
- template<typename columnClass >  
static bool [getSubColumnIndex](#) (const QString &subColumnId, typename columnClass::SubColumn \*const subColumn)

## Additional Inherited Members

## Public Types inherited from [Digikam::TableViewColumn](#)

- enum ColumnCompareResult { CmpEqual = 0 , CmpABiggerB = 1 , CmpALessB = 2 }
- enum ColumnFlag { ColumnNoFlags = 0 , ColumnCustomPainting = 1 , ColumnCustomSorting = 2 , ColumnHasConfigurationWidget = 4 }

## Signals inherited from [Digikam::TableViewColumn](#)

- void [signalAllDataChanged](#) ()
- void [signalDataChanged](#) (const qlonglong imageId)

## Protected Attributes inherited from [Digikam::TableViewColumn](#)

- [TableViewColumnConfiguration](#) configuration
- [TableViewShared](#) \*const **s** = nullptr

### 6.1384.1 Member Function Documentation

#### 6.1384.1.1 data()

```
QVariant Digikam::TableViewColumns::ColumnThumbnail::data (
    TableViewModel::Item *const item,
    const int role ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

#### 6.1384.1.2 getColumnFlags()

```
TableViewColumn::ColumnFlags Digikam::TableViewColumns::ColumnThumbnail::getColumnFlags ( )
const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

#### 6.1384.1.3 getTitle()

```
QString Digikam::TableViewColumns::ColumnThumbnail::getTitle ( ) const [override], [virtual]
```

Implements [Digikam::TableViewColumn](#).

#### 6.1384.1.4 paint()

```
bool Digikam::TableViewColumns::ColumnThumbnail::paint (
    QPainter *const painter,
    const QStyleOptionViewItem & option,
    TableViewModel::Item *const item ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

#### 6.1384.1.5 sizeHint()

```
QSize Digikam::TableViewColumns::ColumnThumbnail::sizeHint (
    const QStyleOptionViewItem & option,
    TableViewModel::Item *const item ) const [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).



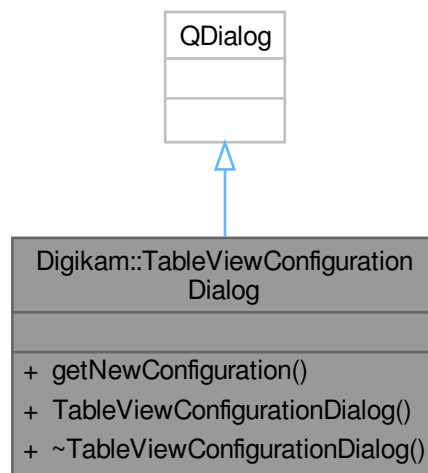
### 6.1384.1.6 updateThumbnailSize()

```
void Digikam::TableViewColumns::ColumnThumbnail::updateThumbnailSize ( ) [override], [virtual]
```

Reimplemented from [Digikam::TableViewColumn](#).

## 6.1385 Digikam::TableViewConfigurationDialog Class Reference

Inheritance diagram for Digikam::TableViewConfigurationDialog:

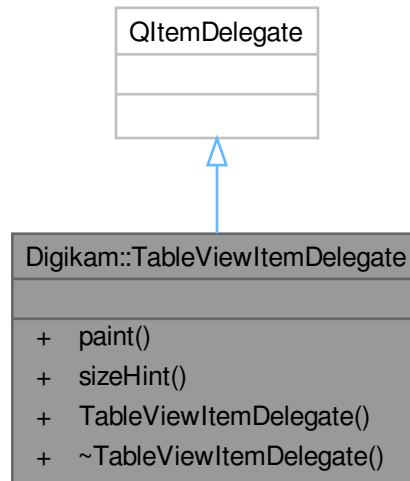


### Public Member Functions

- [TableViewColumnConfiguration](#) `getNewConfiguration` ( ) const
- `TableViewConfigurationDialog` ([TableViewShared](#) \*const sharedObject, const int columnIndex, QWidget \*const parentWidget)

## 6.1386 Digikam::TableViewItemDelegate Class Reference

Inheritance diagram for Digikam::TableViewItemDelegate:



### Public Member Functions

- void **paint** (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &tableViewModel←Index) const override
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &tableViewModelIndex) const override
- **TableViewItemDelegate** (TableViewShared \*const tableViewShared, QObject \*const parent=nullptr)

### 6.1386.1 Member Function Documentation

#### 6.1386.1.1 sizeHint()

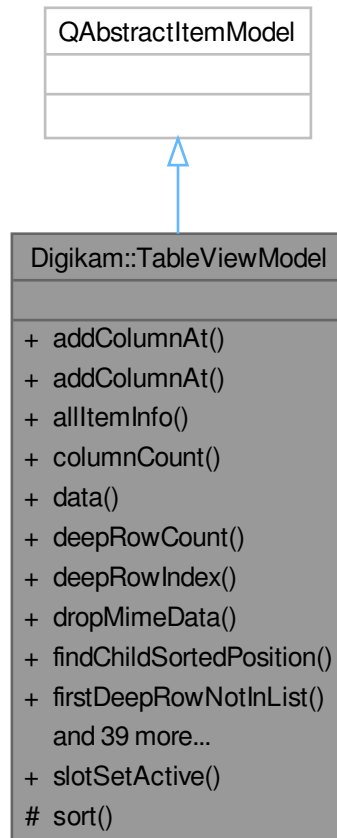
```

QSize Digikam::TableViewItemDelegate::sizeHint (
    const QStyleOptionViewItem & option,
    const QModelIndex & tableViewModelIndex ) const [override]
  
```

we have to take the maximum of all columns for the height

## 6.1387 Digikam::TableViewModel Class Reference

Inheritance diagram for Digikam::TableViewModel:



### Classes

- class [Item](#)

### Public Types

- typedef [DatabaseFields::Hash](#)< QVariant > **DatabaseFieldsHashRaw**
- enum **GroupingMode** { **GroupingHideGrouped** = 0 , **GroupingIgnoreGrouping** = 1 , **GroupingShow**↔  
**SubItems** = 2 }

### Public Slots

- void **slotSetActive** (const bool isActive)

## Signals

- void **signalGroupingModeChanged** ()

## Public Member Functions

- void **addColumnAt** (const [TableViewColumnConfiguration](#) &cpp, const int targetColumn=-1)
  - void **addColumnAt** (const [TableViewColumnDescription](#) &description, const int targetColumn=-1)
  - [QList](#)< [ItemInfo](#) > **allItemInfo** () const
  - int **columnCount** (const [QModelIndex](#) &i) const override
  - [QVariant](#) **data** (const [QModelIndex](#) &i, int role) const override
  - int **deepRowCount** () const
  - [QModelIndex](#) **deepRowIndex** (const int rowNumber) const
  - bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
  - int **findChildSortedPosition** ([Item](#) \*const parentItem, [Item](#) \*const childItem)
  - int **firstDeepRowNotInList** (const [QList](#)< [QModelIndex](#) > &needleList)
  - [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
  - [QModelIndex](#) **fromItemFilterModelIndex** (const [QModelIndex](#) &imageFilterModelIndex)
  - [QModelIndex](#) **fromItemModelIndex** (const [QModelIndex](#) &imageModelIndex)
  - [TableViewColumn](#) \* **getColumnObject** (const int columnIndex)
  - [QList](#)< [TableViewColumn](#) \* > **getColumnObjects** ()
  - [TableViewColumnProfile](#) **getColumnProfile** () const
  - [GroupingMode](#) **groupingMode** () const
  - bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
  - [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role) const override
  - [qulonglong](#) **imageId** (const [QModelIndex](#) &anIndex) const
  - [QList](#)< [qulonglong](#) > **imageIds** (const [QModelIndexList](#) &indexList) const
  - [ItemInfo](#) **imageInfo** (const [QModelIndex](#) &index) const
  - [QList](#)< [ItemInfo](#) > **imageInfos** (const [QModelIndexList](#) &indexList) const
  - [QModelIndex](#) **index** (int row, int column, const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
  - [QModelIndex](#) **indexFromImageId** (const [qulonglong](#) imageId, const int columnIndex) const
  - int **indexToDeepRowNumber** (const [QModelIndex](#) &index) const
  - [ItemInfo](#) **infoFromItem** ([Item](#) \*const item) const
  - [ItemInfoList](#) **infosFromItems** (const [QList](#)< [Item](#) \* > &items) const
  - [QVariant](#) **itemDatabaseFieldRaw** ([Item](#) \*const item, const [DatabaseFields::Set](#) &requestedField)
  - [DatabaseFieldsHashRaw](#) **itemDatabaseFieldsRaw** ([Item](#) \*const item, const [DatabaseFields::Set](#) &requestedSet)
  - [Item](#) \* **itemFromImageId** (const [qulonglong](#) imageId) const
  - [Item](#) \* **itemFromIndex** (const [QModelIndex](#) &i) const
  - [QModelIndex](#) **itemIndex** ([Item](#) \*const item) const
  - bool **lessThan** ([Item](#) \*const itemA, [Item](#) \*const itemB)
  - void **loadColumnProfile** (const [TableViewColumnProfile](#) &columnProfile)
  - [QMimeData](#) \* **mimeData** (const [QModelIndexList](#) &indexes) const override
  - [QStringList](#) **mimeTypes** () const override
  - [QModelIndex](#) **parent** (const [QModelIndex](#) &childIndex) const override
  - void **removeColumnAt** (const int columnIndex)
  - int **rowCount** (const [QModelIndex](#) &parent) const override
  - void **scheduleResort** ()
  - void **setGroupingMode** (const [GroupingMode](#) newGroupingMode)
  - [QList](#)< [Item](#) \* > **sortItems** (const [QList](#)< [Item](#) \* > &itemList)
  - [Qt::DropActions](#) **supportedDropActions** () const override
- drag-and-drop related functions*
- [TableViewModel](#) ([TableViewShared](#) \*const sharedObject, [QObject](#) \*const parent=nullptr)
  - [QModelIndex](#) **toColId** (const [QModelIndex](#) &anIndex) const
  - [QModelIndex](#) **toItemFilterModelIndex** (const [QModelIndex](#) &i) const
  - [QModelIndex](#) **toItemModelIndex** (const [QModelIndex](#) &i) const

## Protected Member Functions

- void `sort` (int `column`, Qt::SortOrder `order`=Qt::AscendingOrder) override

## 6.1387.1 Member Function Documentation

### 6.1387.1.1 addColumnAt()

```
void Digikam::TableViewModel::addColumnAt (
    const TableViewColumnDescription & description,
    const int targetColumn = -1 )
```

### 6.1387.1.2 flags()

```
Qt::ItemFlags Digikam::TableViewModel::flags (
    const QModelIndex & index ) const [override]
```

### 6.1387.1.3 indexFromImageId()

```
QModelIndex Digikam::TableViewModel::indexFromImageId (
    const qlonglong imageId,
    const int columnIndex ) const
```

### 6.1387.1.4 infoFromItem()

```
ItemInfo Digikam::TableViewModel::infoFromItem (
    TableViewModel::Item *const item ) const
```

### 6.1387.1.5 loadColumnProfile()

```
void Digikam::TableViewModel::loadColumnProfile (
    const TableViewColumnProfile & columnProfile )
```

### 6.1387.1.6 parent()

```
QModelIndex Digikam::TableViewModel::parent (
    const QModelIndex & childIndex ) const [override]
```

### 6.1387.1.7 sort()

```
void Digikam::TableViewModel::sort (
    int column,
    Qt::SortOrder order = Qt::AscendingOrder ) [override], [protected]
```

## 6.1388 Digikam::TableViewModel::Item Class Reference

### Public Member Functions

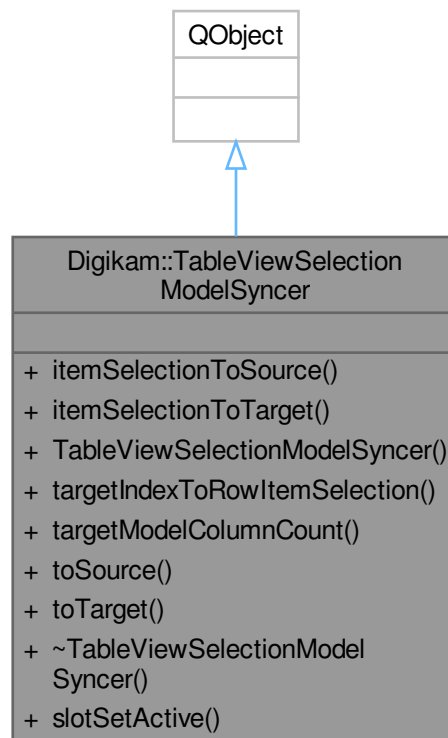
- void **addChild** ([Item](#) \*const newChild)
- [Item](#) \* **findChildWithImageId** (const qlonglong searchImageId)
- void **insertChild** (const int pos, [Item](#) \*const newChild)
- void **takeChild** ([Item](#) \*const oldChild)

### Public Attributes

- QList< [Item](#) \* > **children**
- qlonglong **imageId** = 0
- [Item](#) \* **parent** = nullptr

## 6.1389 Digikam::TableViewSelectionModeSyncer Class Reference

Inheritance diagram for Digikam::TableViewSelectionModeSyncer:



### Public Slots

- void **slotSetActive** (const bool isActive)

## Public Member Functions

- QItemSelection **itemSelectionToSource** (const QItemSelection &selection) const
- QItemSelection **itemSelectionToTarget** (const QItemSelection &selection) const
- [TableViewSelectionModeSyncer](#) ([TableViewShared](#) \*const sharedObject, QObject \*const parent=nullptr)
- QItemSelection **targetIndexToRowItemSelection** (const QModelIndex &targetIndex) const
- int **targetModelColumnCount** () const
- QModelIndex **toSource** (const QModelIndex &targetIndex) const
- QModelIndex **toTarget** (const QModelIndex &sourceIndex) const

## 6.1389.1 Constructor & Destructor Documentation

### 6.1389.1.1 TableViewSelectionModeSyncer()

```
Digikam::TableViewSelectionModeSyncer::TableViewSelectionModeSyncer (
    TableViewShared *const sharedObject,
    QObject *const parent = nullptr ) [explicit]
```

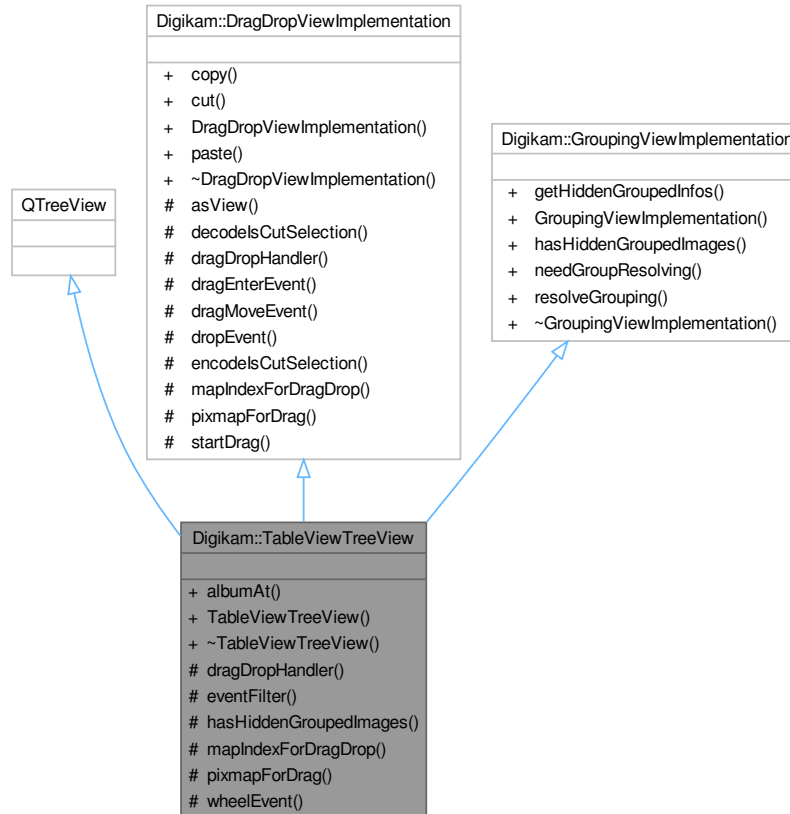
## 6.1390 Digikam::TableViewShared Class Reference

### Public Attributes

- [TableViewColumnFactory](#) \* **columnFactory** = nullptr
- [ItemFilterModel](#) \* **imageFilterModel** = nullptr
- QItemSelectionModel \* **imageFilterSelectionModel** = nullptr
- [ItemModel](#) \* **imageModel** = nullptr
- bool **isActive** = false
- [TableViewItemDelegate](#) \* **itemDelegate** = nullptr
- [TableView](#) \* **tableView** = nullptr
- [TableViewModel](#) \* **tableViewModel** = nullptr
- QItemSelectionModel \* **tableViewSelectionModel** = nullptr
- [TableViewSelectionModeSyncer](#) \* **tableViewSelectionModeSyncer** = nullptr
- [ThumbnailLoadThread](#) \* **thumbnailLoadThread** = nullptr
- [TableViewTreeView](#) \* **treeView** = nullptr

## 6.1391 Digikam::TableViewTreeView Class Reference

Inheritance diagram for Digikam::TableViewTreeView:



### Signals

- void **signalZoomInStep** ()
- void **signalZoomOutStep** ()

### Public Member Functions

- `Album * albumAt` (const QPoint &pos) const
- `TableViewTreeView` (`TableViewShared *const tableViewShared`, `QWidget *const parent=nullptr`)

### Public Member Functions inherited from Digikam::DragDropViewImplementation

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()



## Public Member Functions inherited from Digikam::GroupingViewImplementation

- [ItemInfoList](#) **getHiddenGroupedInfos** (const [ItemInfoList](#) &infos) const
- bool **needGroupResolving** ([OperationType](#) type, const [ItemInfoList](#) &infos) const
- [ItemInfoList](#) **resolveGrouping** (const [ItemInfoList](#) &infos) const

## Protected Member Functions

- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- bool **hasHiddenGroupedImages** (const [ItemInfo](#) &info) const override  
*must be implemented by parent view*
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **wheelEvent** (QWheelEvent \*event) override

## Protected Member Functions inherited from Digikam::DragDropViewImplementation

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

### 6.1391.1 Detailed Description

### 6.1391.2 Member Function Documentation

#### 6.1391.2.1 dragDropHandler()

```
AbstractItemDragDropHandler * Digikam::TableViewTreeView::dragDropHandler ( ) const [override],
[protected], [virtual]
```

You need to implement these three methods Returns the drag drop handler.

Implements [Digikam::DragDropViewImplementation](#).

#### 6.1391.2.2 hasHiddenGroupedImages()

```
bool Digikam::TableViewTreeView::hasHiddenGroupedImages (
    const ItemInfo & ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::GroupingViewImplementation](#).

### 6.1391.2.3 mapIndexForDragDrop()

```
QModelIndex Digikam::TableViewTreeView::mapIndexForDragDrop (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Maps the given index of the view's model to an index of the handler's model, which can be a source model of the view's model.

Implements [Digikam::DragDropViewImplementation](#).

### 6.1391.2.4 pixmapForDrag()

```
QPixmap Digikam::TableViewTreeView::pixmapForDrag (
    const QList< QModelIndex > & indexes ) const [override], [protected], [virtual]
```

Creates a pixmap for dragging the given indexes.

Implements [Digikam::DragDropViewImplementation](#).

## 6.1392 Digikam::TagChangeset Class Reference

### Public Types

- enum [Operation](#) {
 **Unknown** , **Added** , **Moved** , **Deleted** ,
 **Renamed** , **Reparented** , **IconChanged** , [PropertiesChanged](#) }

### Public Member Functions

- [Operation operation](#) () const
- [TagChangeset](#) (int tagId, [Operation operation](#))
- int [tagId](#) () const

### 6.1392.1 Member Enumeration Documentation

#### 6.1392.1.1 Operation

```
enum Digikam::TagChangeset::Operation
```

#### Enumerator

PropertiesChanged	ImageTagProperties Table.
-------------------	---------------------------



## Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) { [CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) , [AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots

- void **slotResetCheckState** ()  
*Resets the whole tag filter.*

## Public Slots inherited from [Digikam::TagFolderView](#)

- void **slotTagNewFromABCMenu** (const QString &personName)

## Public Slots inherited from [Digikam::TagTreeView](#)

- void **setCurrentAlbum** (int tagId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< [Album](#) \* > &tags, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)
- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)
- void **slotCollapseAllNodes** ()  
*slotCollapseAllNodes - collapse all nodes without root node*
- void **slotCollapseNode** ()  
*slotCollapseNode - collapse recursively selected nodes*
- void **slotExpandNode** ()  
*slotExpandNode - expands recursively selected nodes*

## Signals

- void [checkedTagsChanged](#) (const QList< [TAlbum](#) \* > &includedTags, const QList< [TAlbum](#) \* > &excludedTags)  
*Emitted if the checked tags have changed.*

## Signals inherited from [Digikam::TagFolderView](#)

- void **signalFindDuplicates** (const QList< [TAlbum](#) \* > &albums)

## Signals inherited from [Digikam::TagTreeView](#)

- void **assignTags** (int tagId, const QList< int > &imageIds)

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void **currentAlbumChanged** ([Album](#) \*currentAlbum)
- void **selectedAlbumsChanged** (const QList< [Album](#) \* > &selectedAlbums)

## Public Member Functions

- bool **checkNewTags** () const
- void **doLoadState** () override
- void **doSaveState** () override
- QList< [TAlbum](#) \* > **getCheckedTags** () const
- QList< [TAlbum](#) \* > **getPartiallyCheckedTags** () const
- ToggleAutoTags **getToggleAutoTags** () const
- void **setCheckNewTags** (bool checkNewTags)
- if.*
- void **setToggleAutoTags** (ToggleAutoTags toggle)
- **TagCheckView** (QWidget \*const parent, [TagModel](#) \*const tagModel)

## Public Member Functions inherited from [Digikam::TagFolderView](#)

- void **setShowDeleteFaceTagsAction** (bool show)
- void **setShowFindDuplicateAction** (bool show)
- [TagFolderView](#) (QWidget \*const parent, [TagModel](#) \*const model)
- void **tagPropsEdit** ()
- **~TagFolderView** () override

## Public Member Functions inherited from [Digikam::TagTreeView](#)

- [TAlbum](#) \* **albumForIndex** (const QModelIndex &index) const
- [TagModel](#) \* **albumModel** () const
- [TAlbum](#) \* **currentAlbum** () const
- currentAlbum Even if multiple selection is enabled current Album can be only one, the last clicked item if you need selected items, see selectedAlbums() It's NOT the same as AlbumManager::currentAlbums()*
- [TagPropertiesFilterModel](#) \* **filteredModel** () const
- QList< [TAlbum](#) \* > **selectedTagAlbums** ()
- QList< [Album](#) \* > **selectedTags** ()
- selectedTags - return a list of all selected items in tag model*
- void **setAlbumFilterModel** ([TagPropertiesFilterModel](#) \*const filteredModel, [CheckableAlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([TagModel](#) \*const model)
- [TagModificationHelper](#) \* **tagModificationHelper** () const
- **TagTreeView** (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- [CheckableAlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [albumModel](#) () const
- [CheckableAlbumFilterModel](#) \* [checkableAlbumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [checkableModel](#) () const
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void [addContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractSpecificAlbumModel](#) \* [albumModel](#) () const
- QList< [ContextMenuElement](#) \* > [contextMenuElements](#) () const
- template<class A >  
QList< A \* > [currentAlbums](#) ()
- bool [expandMatches](#) (const QModelIndex &index)
- QModelIndex [indexVisuallyAt](#) (const QPoint &p)
- void [removeContextMenuElement](#) ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > [selectedItems](#) ()  
*selectedItems()* -
- void [setAlbumManagerCurrentAlbum](#) (const bool setCurrentAlbum)
- void [setContextMenuIcon](#) (const QPixmap &pixmap)
- void [setContextMenuTitle](#) (const QString &title)
- void [setEnabledContextMenu](#) (const bool enable)
- void [setExpandNewCurrentItem](#) (const bool doThat)
- void [setExpandOnSingleClick](#) (const bool doThat)
- void [setSelectAlbumOnClick](#) (const bool selectOnClick)
- void [setSelectOnContextMenu](#) (const bool select)
- bool [viewportEvent](#) (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

**Protected Member Functions**

- void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override

**Protected Member Functions inherited from [Digikam::TagFolderView](#)**

- void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override
- void [contextMenuEvent](#) ([QContextMenuEvent](#) \*event) override
- [QString](#) [contextMenuTitle](#) () const override
- void [handleCustomContextMenuAction](#) ([QAction](#) \*action, const [AlbumPointer](#)< [Album](#) > &album) override
- void [keyPressEvent](#) ([QKeyEvent](#) \*event) override
- virtual void [setContextMenuItems](#) ([ContextMenuHelper](#) &cmh, const [QList](#)< [TAlbum](#) \* > &albums)

**Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)**

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

**Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)**

- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

**Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)**

- virtual [QPixmap](#) [contextMenuIcon](#) () const
  - void [dragEnterEvent](#) ([QDragEnterEvent](#) \*e) override
  - void [dragLeaveEvent](#) ([QDragLeaveEvent](#) \*e) override
  - void [dragMoveEvent](#) ([QDragMoveEvent](#) \*e) override
  - void [dropEvent](#) ([QDropEvent](#) \*e) override
  - void [mousePressEvent](#) ([QMouseEvent](#) \*e) override
- Other helper methods.*
- virtual [QPixmap](#) [pixmapForDrag](#) (const [QStyleOptionViewItem](#) &option, [QList](#)< [QModelIndex](#) > indexes)
  - void [rowsAboutToBeRemoved](#) (const [QModelIndex](#) &parent, int start, int end) override
  - void [rowsInserted](#) (const [QModelIndex](#) &index, int start, int end) override
  - void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
  - void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
  - virtual bool [showContextMenuAt](#) ([QContextMenuEvent](#) \*event, [Album](#) \*albumForEvent)
  - void [startDrag](#) ([Qt::DropActions](#) supportedActions) override

**Protected Member Functions inherited from [Digikam::StateSavingObject](#)**

- [QString](#) [entryName](#) (const [QString](#) &base) const
- [KConfigGroup](#) [getConfigGroup](#) () const

## Additional Inherited Members

### Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void `albumSettingsChanged` ()
- void `slotCurrentChanged` ()
- virtual void `slotRootAlbumAvailable` ()
- void `slotSearchTextSettingsAboutToChange` (bool searched, bool willSearch)
- void `slotSearchTextSettingsChanged` (bool wasSearching, bool searching)
- void `slotSelectionChanged` ()

### Protected Attributes inherited from [Digikam::TagTreeView](#)

- [TagPropertiesFilterModel](#) \* `m_filteredModel` = nullptr
- [TagModificationHelper](#) \* `m_modificationHelper` = nullptr

### Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* `m_albumFilterModel` = nullptr
- [AbstractSpecificAlbumModel](#) \* `m_albumModel` = nullptr
- bool `m_checkOnMiddleClick` = false
- [AlbumModelDragDropHandler](#) \* `m_dragDropHandler` = nullptr
- Flags `m_flags` = DefaultFlags
- int `m_lastScrollBarValue` = 0
- bool `m_restoreCheckState` = false

## 6.1393.1 Member Function Documentation

### 6.1393.1.1 `addCustomContextMenuActions()`

```
void Digikam::TagCheckView::addCustomContextMenuActions (
    ContextMenuHelper & cmh,
    Album * album ) [override], [protected], [virtual]
```

Hook method to add custom actions to the generated context menu.

#### Parameters

<i>cmh</i>	helper object to create the context menu
<i>album</i>	tag on which the context menu will be created. May be null if it is requested on no tag entry

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

Reimplemented in [Digikam::TagFilterView](#).

### 6.1393.1.2 `checkedTagsChanged`

```
void Digikam::TagCheckView::checkedTagsChanged (
    const QList< TAlbum * > & includedTags,
    const QList< TAlbum * > & excludedTags ) [signal]
```



## Parameters

<i>includedTags</i>	a list of selected tag ids processed.
<i>excludedTags</i>	a list of tag ids ignored.

**6.1393.1.3 doLoadState()**

```
void Digikam::TagCheckView::doLoadState ( ) [override], [virtual]
```

Implements state loading for the album tree view in a somewhat clumsy procedure because the model may not be fully loaded when this method is called. Therefore the config is first parsed into `d->statesByAlbumId` which holds the state of all tree view entries indexed by album id. Afterwards an initial sync run is done restoring the state of all model entries that are already present at this time. Every processed entry is removed from `d->statesByAlbumId`. If there are still entries left in this map we assume that the model is not fully loaded at the moment. Therefore the `rowsInserted` signal is connected to a slot that restores the state of new rows based on the remaining entries in `d->statesByAlbumId`.

Reimplemented from [Digikam::AbstractCheckableAlbumTreeView](#).

**6.1393.1.4 doSaveState()**

```
void Digikam::TagCheckView::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Reimplemented from [Digikam::AbstractCheckableAlbumTreeView](#).

**6.1393.1.5 setCheckNewTags()**

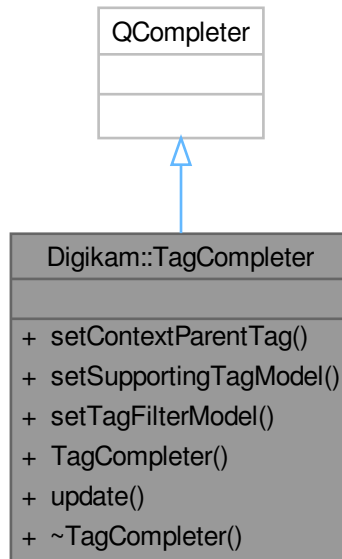
```
void Digikam::TagCheckView::setCheckNewTags (
    bool checkNewTags )
```

## Parameters

<i>checkNewTags</i>	is switched on, a tag that is created from <i>within</i> this view, typically via the context menu, will automatically be set checked.
---------------------	--

## 6.1394 Digikam::TagCompleter Class Reference

Inheritance diagram for Digikam::TagCompleter:



### Signals

- void **signalActivated** (const [TaggingAction](#) &action)
- void **signalHighlighted** (const [TaggingAction](#) &action)

### Public Member Functions

- void [setContextParentTag](#) (int parentTagId)
- void [setSupportingTagModel](#) ([TagModel](#) \*const supportingModel)
- void **setTagFilterModel** ([AlbumFilterModel](#) \*const supportingModel)
- [TagCompleter](#) (QObject \*const parent=nullptr)
- void [update](#) (const QString &fragment)

### 6.1394.1 Constructor & Destructor Documentation

#### 6.1394.1.1 TagCompleter()

```

Digikam::TagCompleter::TagCompleter (
    QObject *const parent = nullptr ) [explicit]
  
```

A completion object operating on a [TagModel](#)

## 6.1394.2 Member Function Documentation

### 6.1394.2.1 setContextParentTag()

```
void Digikam::TagCompleter::setContextParentTag (
    int parentTagId )
```

Set a parent tag which may by the user be considered as a parent for a new tag during completion

### 6.1394.2.2 setSupportingTagModel()

```
void Digikam::TagCompleter::setSupportingTagModel (
    TagModel *const supportingModel )
```

Set a supporting model from which the completer may get data for its display. Optional.

### 6.1394.2.3 update()

```
void Digikam::TagCompleter::update (
    const QString & fragment )
```

Update the completer for the given fragment

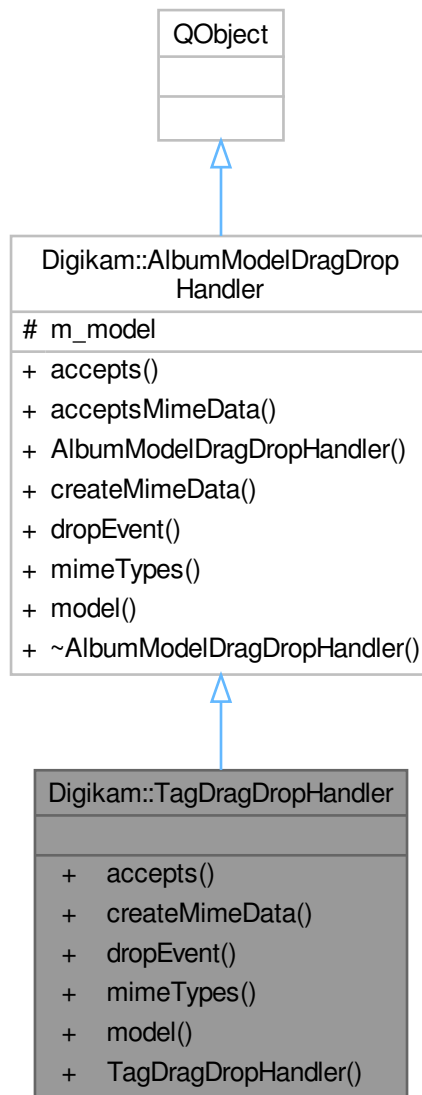
## 6.1395 Digikam::TagData Struct Reference

### Public Attributes

- QString **tagName**
- Type **tagType** = TypeChild
- QString **tipName**

## 6.1396 Digikam::TagDragDropHandler Class Reference

Inheritance diagram for Digikam::TagDragDropHandler:



### Signals

- void **assignTags** (const QList< qlonglong > &imageIDs, const QList< int > &tagIDs)

### Public Member Functions

- Qt::DropAction **accepts** (const QDropEvent \*e, const QModelIndex &dropIndex) override
- QMimeData \* **createMimeData** (const QList< Album \* > &) override

- bool [dropEvent](#) (QAbstractItemView \*view, const QDropEvent \*e, const QModelIndex &droppedOn) override
- QStringList [mimeTypes](#) () const override
- [TagModel](#) \* **model** () const
- [TagDragDropHandler](#) ([TagModel](#) \*const model)

## Public Member Functions inherited from [Digikam::AlbumModelDragDropHandler](#)

- virtual bool [acceptsMimeData](#) (const QMimeData \*data)
- [AlbumModelDragDropHandler](#) ([AbstractAlbumModel](#) \*model)
- [AbstractAlbumModel](#) \* **model** () const

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::AlbumModelDragDropHandler](#)

- [AbstractAlbumModel](#) \* **m\_model** = nullptr

## 6.1396.1 Member Function Documentation

### 6.1396.1.1 [accepts\(\)](#)

```
Qt::DropAction Digikam::TagDragDropHandler::accepts (
    const QDropEvent * e,
    const QModelIndex & dropIndex ) [override], [virtual]
```

Returns if the given mime data is accepted for drop on dropIndex. Returns the proposed action, or Qt::IgnoreAction if not accepted.

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

### 6.1396.1.2 [createMimeData\(\)](#)

```
QMimeData * Digikam::TagDragDropHandler::createMimeData (
    const QList< Album * > & ) [override], [virtual]
```

Create a mime data object for starting a drag from the given Albums

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

### 6.1396.1.3 [dropEvent\(\)](#)

```
bool Digikam::TagDragDropHandler::dropEvent (
    QAbstractItemView * view,
    const QDropEvent * e,
    const QModelIndex & droppedOn ) [override], [virtual]
```

Gives the view and the occurring drop event. The index is the index where the drop was dropped on. It may be invalid (dropped on decoration, viewport) Returns true if the event is to be accepted.

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

### 6.1396.1.4 mimeTypees()

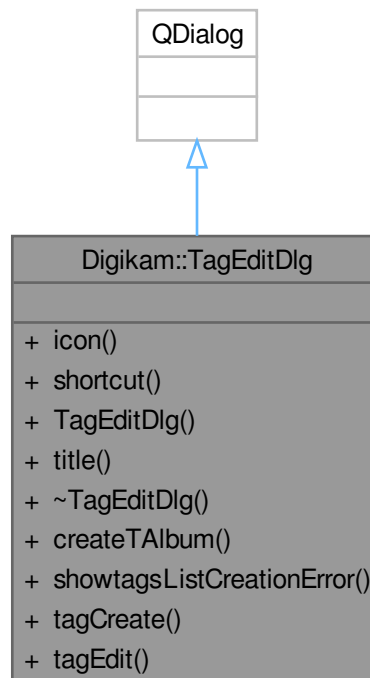
```
QStringList Digikam::TagDragDropHandler::mimeTypees ( ) const [override], [virtual]
```

Returns the supported mime types. Called by the default implementation of model's [mimeTypees\(\)](#).

Reimplemented from [Digikam::AlbumModelDragDropHandler](#).

## 6.1397 Digikam::TagEditDlg Class Reference

Inheritance diagram for Digikam::TagEditDlg:



### Public Member Functions

- QString **icon** () const
- QKeySequence **shortcut** () const
- **TagEditDlg** (QWidget \*const parent, [TAlbum](#) \*const album, bool create=false)
- QString **title** () const

## Static Public Member Functions

- static AlbumList [createTAlbum](#) (TAlbum \*const mainRootAlbum, const QString &tagStr, const QString &icon, const QKeySequence &ks, QMap< QString, QString > &errMap)
- static void **showtagsListCreationError** (QWidget \*const parent, const QMap< QString, QString > &errMap)
- static bool **tagCreate** (QWidget \*const parent, TAlbum \*const album, QString &title, QString &icon, QKeySequence &ks)
- static bool **tagEdit** (QWidget \*const parent, TAlbum \*const album, QString &title, QString &icon, QKeySequence &ks)

## 6.1397.1 Member Function Documentation

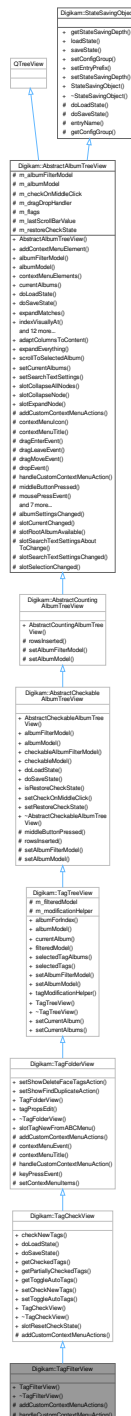
### 6.1397.1.1 createTAlbum()

```
AlbumList Digikam::TagEditDlg::createTAlbum (  
    TAlbum *const mainRootAlbum,  
    const QString & tagStr,  
    const QString & icon,  
    const QKeySequence & ks,  
    QMap< QString, QString > & errMap ) [static]
```

Create a list of new Tag album using a list of tags hierarchies separated by ",". A hierarchy of tags is a string path of tags name separated by "/". If a hierarchy start by "/" or if mainRootAlbum is null, it will be created from root tag album, else it will be created from mainRootAlbum as parent album. 'errMap' is Map of TAlbum path and error message if tag creation failed. Return the list of created Albums.

## 6.1398 Digikam::TagFilterView Class Reference

Inheritance diagram for Digikam::TagFilterView:



### Public Member Functions

- [TagFilterView](#) (QWidget \*const parent, [TagModel](#) \*const tagFilterModel)
- [~TagFilterView](#) () override



## Public Member Functions inherited from Digikam::TagCheckView

- bool **checkNewTags** () const
- void **doLoadState** () override
- void **doSaveState** () override
- QList< TAlbum \* > **getCheckedTags** () const
- QList< TAlbum \* > **getPartiallyCheckedTags** () const
- ToggleAutoTags **getToggleAutoTags** () const
- void **setCheckNewTags** (bool checkNewTags)
- *If.*
- void **setToggleAutoTags** (ToggleAutoTags toggle)
- **TagCheckView** (QWidget \*const parent, TagModel \*const tagModel)

## Public Member Functions inherited from Digikam::TagFolderView

- void **setShowDeleteFaceTagsAction** (bool show)
- void **setShowFindDuplicateAction** (bool show)
- TagFolderView (QWidget \*const parent, TagModel \*const model)
- void **tagPropsEdit** ()
- ~TagFolderView () override

## Public Member Functions inherited from Digikam::TagTreeView

- TAlbum \* **albumForIndex** (const QModelIndex &index) const
- TagModel \* **albumModel** () const
- TAlbum \* **currentAlbum** () const
- *currentAlbum Even if multiple selection is enabled current Album can be only one, the last clicked item if you need selected items, see selectedAlbums() It's NOT the same as AlbumManager::currentAlbums()*
- TagPropertiesFilterModel \* **filteredModel** () const
- QList< TAlbum \* > **selectedTagAlbums** ()
- QList< Album \* > **selectedTags** ()
- *selectedTags - return a list of all selected items in tag model*
- void **setAlbumFilterModel** (TagPropertiesFilterModel \*const filteredModel, CheckableAlbumFilterModel \*const filterModel)
- void **setAlbumModel** (TagModel \*const model)
- TagModificationHelper \* **tagModificationHelper** () const
- TagTreeView (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)

## Public Member Functions inherited from Digikam::AbstractCheckableAlbumTreeView

- AbstractCheckableAlbumTreeView (QWidget \*const parent, Flags flags)
- CheckableAlbumFilterModel \* **albumFilterModel** () const
- AbstractCheckableAlbumModel \* **albumModel** () const
- CheckableAlbumFilterModel \* **checkableAlbumFilterModel** () const
- AbstractCheckableAlbumModel \* **checkableModel** () const
- bool **isRestoreCheckState** () const
- void **setCheckOnMiddleClick** (bool doThat)
- void **setRestoreCheckState** (bool restore)

## Public Member Functions inherited from Digikam::AbstractCountingAlbumTreeView

- AbstractCountingAlbumTreeView (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Protected Member Functions

- void **addCustomContextMenuActions** ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override
- void **handleCustomContextMenuAction** (QAction \*action, const [AlbumPointer](#)< [Album](#) > &album) override

## Protected Member Functions inherited from [Digikam::TagFolderView](#)

- void **addCustomContextMenuActions** ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override
- void **contextMenuEvent** (QContextMenuEvent \*event) override
- QString **contextMenuTitle** () const override
- void **handleCustomContextMenuAction** (QAction \*action, const [AlbumPointer](#)< [Album](#) > &album) override
- void **keyPressEvent** (QKeyEvent \*event) override
- virtual void **setContextMenuItems** ([ContextMenuHelper](#) &cmh, const QList< [TAlbum](#) \* > &albums)

### Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual [QPixmap](#) [contextMenuIcon](#) () const
  - void [dragEnterEvent](#) ([QDragEnterEvent](#) \*e) override
  - void [dragLeaveEvent](#) ([QDragLeaveEvent](#) \*e) override
  - void [dragMoveEvent](#) ([QDragMoveEvent](#) \*e) override
  - void [dropEvent](#) ([QDropEvent](#) \*e) override
  - void [mousePressEvent](#) ([QMouseEvent](#) \*e) override
- Other helper methods.*
- virtual [QPixmap](#) [pixmapForDrag](#) (const [QStyleOptionViewItem](#) &option, [QList](#)< [QModelIndex](#) > indexes)
  - void [rowsAboutToBeRemoved](#) (const [QModelIndex](#) &parent, int start, int end) override
  - void [rowsInserted](#) (const [QModelIndex](#) &index, int start, int end) override
  - void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
  - void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
  - virtual bool [showContextMenuAt](#) ([QContextMenuEvent](#) \*event, [Album](#) \*albumForEvent)
  - void [startDrag](#) ([Qt::DropActions](#) supportedActions) override

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- [QString](#) [entryName](#) (const [QString](#) &base) const
- [KConfigGroup](#) [getConfigGroup](#) () const

### Additional Inherited Members

### Public Types inherited from [Digikam::TagCheckView](#)

- enum [ToggleAutoTags](#) { [NoToggleAuto](#) = 0 , [Children](#) , [Parents](#) , [ChildrenAndParents](#) }

### Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) { [CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) , [AlwaysShowInclusiveCounts](#) , [DefaultFlags](#) = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots inherited from [Digikam::TagCheckView](#)

- void [slotResetCheckState](#) ()  
*Resets the whole tag filter.*

## Public Slots inherited from [Digikam::TagFolderView](#)

- void [slotTagNewFromABCMenu](#) (const QString &personName)

## Public Slots inherited from [Digikam::TagTreeView](#)

- void [setCurrentAlbum](#) (int tagId, bool selectInAlbumManager=true)
- void [setCurrentAlbums](#) (const QList< [Album](#) \* > &tags, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()  
*slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()  
*slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()  
*slotExpandNode - expands recursively selected nodes*

## Signals inherited from [Digikam::TagCheckView](#)

- void [checkedTagsChanged](#) (const QList< [TAlbum](#) \* > &includedTags, const QList< [TAlbum](#) \* > &excludedTags)  
*Emitted if the checked tags have changed.*

## Signals inherited from [Digikam::TagFolderView](#)

- void [signalFindDuplicates](#) (const QList< [TAlbum](#) \* > &albums)

## Signals inherited from [Digikam::TagTreeView](#)

- void [assignTags](#) (int tagId, const QList< int > &imageIds)

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const [QList](#)< [Album](#) \* > &selectedAlbums)

## Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Attributes inherited from [Digikam::TagTreeView](#)

- [TagPropertiesFilterModel](#) \* [m\\_filteredModel](#) = nullptr
- [TagModificationHelper](#) \* [m\\_modificationHelper](#) = nullptr

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* [m\\_albumFilterModel](#) = nullptr
- [AbstractSpecificAlbumModel](#) \* [m\\_albumModel](#) = nullptr
- bool [m\\_checkOnMiddleClick](#) = false
- [AlbumModelDragDropHandler](#) \* [m\\_dragDropHandler](#) = nullptr
- Flags [m\\_flags](#) = DefaultFlags
- int [m\\_lastScrollBarValue](#) = 0
- bool [m\\_restoreCheckState](#) = false

### 6.1398.1 Detailed Description

A view to filter the currently displayed album by tags.

Author

jwienke

### 6.1398.2 Constructor & Destructor Documentation

#### 6.1398.2.1 TagFilterView()

```
Digikam::TagFilterView::TagFilterView (  
    QWidget *const parent,  
    TagModel *const tagFilterModel ) [explicit]
```

Constructor.

## Parameters

<i>parent</i>	the parent for qt parent child mechanism
<i>tagFilterModel</i>	tag model to work on

**6.1398.2.2 ~TagFilterView()**

```
Digikam::TagFilterView::~TagFilterView ( ) [override]
```

Destructor.

**6.1398.3 Member Function Documentation****6.1398.3.1 addCustomContextMenuActions()**

```
void Digikam::TagFilterView::addCustomContextMenuActions (
    ContextMenuHelper & cmh,
    Album * album ) [override], [protected], [virtual]
```

Hook method to add custom actions to the generated context menu.

## Parameters

<i>cmh</i>	helper object to create the context menu
<i>album</i>	tag on which the context menu will be created. May be null if it is requested on no tag entry

Reimplemented from [Digikam::TagCheckView](#).

**6.1398.3.2 handleCustomContextMenuAction()**

```
void Digikam::TagFilterView::handleCustomContextMenuAction (
    QAction * action,
    const AlbumPointer< Album > & album ) [override], [protected], [virtual]
```

Hook method to handle the custom context menu actions that were added with addCustomContextMenuActions.

## Parameters

<i>action</i>	the action that was chosen by the user, may be null if none of the custom actions were selected
<i>album</i>	the tag on which the context menu was requested. May be null if there was no

Reimplemented from [Digikam::AbstractAlbumTreeView](#).



## Public Slots inherited from [Digikam::TagTreeView](#)

- void **setCurrentAlbum** (int tagId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< [Album](#) \* > &tags, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void **adaptColumnsToContent** ()
- void **expandEverything** (const QModelIndex &index)
- void **scrollToSelectedAlbum** ()
- void **setCurrentAlbums** (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)
- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)
- void **slotCollapseAllNodes** ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void **slotCollapseNode** ()
  - slotCollapseNode - collapse recursively selected nodes*
- void **slotExpandNode** ()
  - slotExpandNode - expands recursively selected nodes*

## Signals

- void **signalFindDuplicates** (const QList< [TAlbum](#) \* > &albums)

## Signals inherited from [Digikam::TagTreeView](#)

- void **assignTags** (int tagId, const QList< int > &imageIds)

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void **currentAlbumChanged** ([Album](#) \*currentAlbum)
- void **selectedAlbumsChanged** (const QList< [Album](#) \* > &selectedAlbums)

## Public Member Functions

- void **setShowDeleteFaceTagsAction** (bool show)
- void **setShowFindDuplicateAction** (bool show)
- [TagFolderView](#) (QWidget \*const parent, [TagModel](#) \*const model)
- void **tagPropsEdit** ()
- [~TagFolderView](#) () override

## Public Member Functions inherited from [Digikam::TagTreeView](#)

- [TAlbum](#) \* **albumForIndex** (const QModelIndex &index) const
- [TagModel](#) \* **albumModel** () const
- [TAlbum](#) \* **currentAlbum** () const
  - currentAlbum Even if multiple selection is enabled current Album can be only one, the last clicked item if you need selected items, see selectedAlbums() It's NOT the same as AlbumManager::currentAlbums()*
- [TagPropertiesFilterModel](#) \* **filteredModel** () const
- QList< [TAlbum](#) \* > **selectedTagAlbums** ()
- QList< [Album](#) \* > **selectedTags** ()
  - selectedTags - return a list of all selected items in tag model*
- void **setAlbumFilterModel** ([TagPropertiesFilterModel](#) \*const filteredModel, [CheckableAlbumFilterModel](#) \*const filterModel)
- void **setAlbumModel** ([TagModel](#) \*const model)
- [TagModificationHelper](#) \* **tagModificationHelper** () const
- **TagTreeView** (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)



## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- [CheckableAlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractCheckableAlbumModel](#) \* **albumModel** () const
- [CheckableAlbumFilterModel](#) \* **checkableAlbumFilterModel** () const
- [AbstractCheckableAlbumModel](#) \* **checkableModel** () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool [expandMatches](#) (const QModelIndex &index)
- QModelIndex [indexVisuallyAt](#) (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void [setAlbumManagerCurrentAlbum](#) (const bool setCurrentAlbum)
- void [setContextMenuIcon](#) (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void [setEnabledContextMenu](#) (const bool enable)
- void [setExpandNewCurrentItem](#) (const bool doThat)
- void [setExpandOnSingleClick](#) (const bool doThat)
- void [setSelectAlbumOnClick](#) (const bool selectOnClick)
- void [setSelectOnContextMenu](#) (const bool select)
- bool [viewportEvent](#) (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

**Protected Member Functions**

- void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override
- void [contextMenuEvent](#) ([QContextMenuEvent](#) \*event) override
- [QString](#) [contextMenuTitle](#) () const override
- void [handleCustomContextMenuAction](#) ([QAction](#) \*action, const [AlbumPointer](#)< [Album](#) > &album) override
- void [keyPressEvent](#) ([QKeyEvent](#) \*event) override
- virtual void [setContextMenuItems](#) ([ContextMenuHelper](#) &cmh, const [QList](#)< [TAlbum](#) \* > &albums)

**Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)**

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

**Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)**

- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

**Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)**

- virtual [QPixmap](#) [contextMenuIcon](#) () const
  - void [dragEnterEvent](#) ([QDragEnterEvent](#) \*e) override
  - void [dragLeaveEvent](#) ([QDragLeaveEvent](#) \*e) override
  - void [dragMoveEvent](#) ([QDragMoveEvent](#) \*e) override
  - void [dropEvent](#) ([QDropEvent](#) \*e) override
  - void [mousePressEvent](#) ([QMouseEvent](#) \*e) override
- Other helper methods.*
- virtual [QPixmap](#) [pixmapForDrag](#) (const [QStyleOptionViewItem](#) &option, [QList](#)< [QModelIndex](#) > indexes)
  - void [rowsAboutToBeRemoved](#) (const [QModelIndex](#) &parent, int start, int end) override
  - void [rowsInserted](#) (const [QModelIndex](#) &index, int start, int end) override
  - void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
  - void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
  - virtual bool [showContextMenuAt](#) ([QContextMenuEvent](#) \*event, [Album](#) \*albumForEvent)
  - void [startDrag](#) ([Qt::DropActions](#) supportedActions) override

**Protected Member Functions inherited from [Digikam::StateSavingObject](#)**

- [QString](#) [entryName](#) (const [QString](#) &base) const
- [KConfigGroup](#) [getConfigGroup](#) () const

**Additional Inherited Members****Public Types inherited from [Digikam::AbstractAlbumTreeView](#)**

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **[DefaultFlags](#)** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from Digikam::StateSavingObject

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Slots inherited from Digikam::AbstractAlbumTreeView

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

## Protected Attributes inherited from Digikam::TagTreeView

- [TagPropertiesFilterModel](#) \* [m\\_filteredModel](#) = nullptr
- [TagModificationHelper](#) \* [m\\_modificationHelper](#) = nullptr

## Protected Attributes inherited from Digikam::AbstractAlbumTreeView

- [AlbumFilterModel](#) \* [m\\_albumFilterModel](#) = nullptr
- [AbstractSpecificAlbumModel](#) \* [m\\_albumModel](#) = nullptr
- bool [m\\_checkOnMiddleClick](#) = false
- [AlbumModelDragDropHandler](#) \* [m\\_dragDropHandler](#) = nullptr
- Flags [m\\_flags](#) = DefaultFlags
- int [m\\_lastScrollBarValue](#) = 0
- bool [m\\_restoreCheckState](#) = false

## 6.1399.1 Constructor & Destructor Documentation

### 6.1399.1.1 TagFolderView()

```
Digikam::TagFolderView::TagFolderView (
    QWidget *const parent,
    TagModel *const model )
```

Constructor.

Parameters

<i>parent</i>	the parent for Qt's parent child mechanism
<i>model</i>	tag model to display

This ensures that the View appears sorted

### 6.1399.1.2 ~TagFolderView()

```
Digikam::TagFolderView::~~TagFolderView ( ) [override]
```

Destructor.

## 6.1399.2 Member Function Documentation

### 6.1399.2.1 addCustomContextMenuActions()

```
void Digikam::TagFolderView::addCustomContextMenuActions (
    ContextMenuHelper & cmh,
    Album * album ) [override], [protected], [virtual]
```

Hook method to add custom actions to the generated context menu.

The default implementation adds actions to reset the tag icon and to find duplicates in a tag album. If you want to use these actions, remember to call this class' implementation of this method and the `handleCustomContextMenuAction` in your derived class.

#### Parameters

<i>cmh</i>	helper object to create the context menu
<i>album</i>	tag on which the context menu will be created. May be null if it is requested on no tag entry

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

### 6.1399.2.2 contextMenuEvent()

```
void Digikam::TagFolderView::contextMenuEvent (
    QContextMenuEvent * event ) [override], [protected]
```

Reimplement `contextMenuEvent` from `AbstractAlbumTree` to support multiple selection

#### Parameters

<i>event</i>	context menu event triggered by right click
--------------	---

If no item is selected append root tag

### 6.1399.2.3 contextMenuTitle()

```
QString Digikam::TagFolderView::contextMenuTitle ( ) const [override], [protected], [virtual]
```

Hook method to implement that returns the title for the context menu.

#### Returns

title for the context menu

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

**6.1399.2.4 handleCustomContextMenuAction()**

```
void Digikam::TagFolderView::handleCustomContextMenuAction (
    QAction * action,
    const AlbumPointer< Album > & album ) [override], [protected], [virtual]
```

Hook method to handle the custom context menu actions that were added with addCustomContextMenuActions.

**Parameters**

<i>action</i>	the action that was chosen by the user, may be null if none of the custom actions were selected
<i>album</i>	the tag on which the context menu was requested. May be null if there was no

Reimplemented from [Digikam::AbstractAlbumTreeView](#).

**6.1399.2.5 setContextMenuItems()**

```
void Digikam::TagFolderView::setContextMenuItems (
    ContextMenuHelper & cmh,
    const QList< TAlbum * > & albums ) [protected], [virtual]
```

Implementation of AddCustomContextMenuActions(see above) that handle multiple selection. If only one element is selected, only AddCustomContextMenuActions is called

**Parameters**

<i>cmh</i>	- helper object to create context menu
<i>albums</i>	- vector of selected albums to be used on menu actions

Reimplemented in [Digikam::TagMngrTreeView](#).

**6.1399.2.6 setShowDeleteFaceTagsAction()**

```
void Digikam::TagFolderView::setShowDeleteFaceTagsAction (
    bool show )
```

Define whether to show the "Delete People Tags" action in context menus or not.

**Parameters**

<i>show</i>	if <code>true</code> the action to delete people tags in the tag album is displayed
-------------	---

**6.1399.2.7 setShowFindDuplicateAction()**

```
void Digikam::TagFolderView::setShowFindDuplicateAction (
    bool show )
```

Define whether to show the "find duplicate" action in context menus or not.

## Parameters

<code>show</code>	if <code>true</code> the action to find duplicate images in the tag album is displayed
-------------------	--

**6.1399.2.8 tagPropsEdit()**

```
void Digikam::TagFolderView::tagPropsEdit ( )
```

Open tag for editing.

**6.1400 Digikam::TaggingAction Class Reference****Public Types**

- enum [Type](#) { [NoAction](#) , [AssignTag](#) , [CreateNewTag](#) }

**Public Member Functions**

- bool **isValid** () const
- QString **newTagName** () const
  - If shallCreateNewTag(), returns the tag name and the parent tag id, 0 for toplevel tag.*
- bool **operator==** (const [TaggingAction](#) &other) const
- int **parentTagId** () const
- bool **shallAssignTag** () const
- bool **shallCreateNewTag** () const
- [TaggingAction](#) ()=default
- [TaggingAction](#) (const QString &name, int parentTagId)
- [TaggingAction](#) (int tagId)
- int **tagId** () const
  - If shallAssignTag(), returns the tag id.*
- [Type](#) **type** () const

**Protected Attributes**

- int **m\_tagId** = -1
- QString **m\_tagName**
- [Type](#) **m\_type** = NoAction

**6.1400.1 Member Enumeration Documentation****6.1400.1.1 Type**

```
enum Digikam::TaggingAction::Type
```

Describes two possible actions: Assigning an existing tag, known by tag id, or creation of a new tag, with a given tag name and a parent tag.

## 6.1400.2 Constructor & Destructor Documentation

### 6.1400.2.1 TaggingAction() [1/3]

```
Digikam::TaggingAction::TaggingAction ( ) [default]
```

Create a NoAction

### 6.1400.2.2 TaggingAction() [2/3]

```
Digikam::TaggingAction::TaggingAction (
    int tagId ) [explicit]
```

Assign the existing tag with given id

### 6.1400.2.3 TaggingAction() [3/3]

```
Digikam::TaggingAction::TaggingAction (
    const QString & name,
    int parentTagId )
```

Create a new tag with the given name. The parent shall be the tag with the given id, or 0 for a toplevel tag.

## 6.1401 Digikam::TaggingActionFactory Class Reference

### Classes

- class [ConstraintInterface](#)

### Public Types

- enum [NameMatchMode](#) { [MatchStartingWithFragment](#) , [MatchContainingFragment](#) }

### Public Member Functions

- [QList< TaggingAction > actions](#) () const  
*Returns the sorted list of suggested tagging actions, based on the above settings.*
- [ConstraintInterface \\* constraintInterface](#) () const
- [TaggingAction defaultTaggingAction](#) () const  
*Returns one single action, which is decided to be the presumably best action based on the settings.*
- [QString fragment](#) () const
- [int indexOfDefaultAction](#) () const  
*Returns the index of the default action in the list returned by generate()*
- [int indexOfLastRecentAction](#) () const  
*Returns the index of the last recent action in the list returned by actions()*
- [NameMatchMode nameMatchMode](#) () const
- [int parentTagId](#) () const

- void **reset** ()  
*reset all settings to the default (no fragment, no actions)*
- void **setConstraintInterface** ([ConstraintInterface](#) \*const iface)
- void **setFragment** (const QString &fragment)  
*Set a fragment of a tag name to generate possible tags, as known from completers.*
- void **setNameMatchMode** ([NameMatchMode](#) mode)  
*Set the matching mode for the tag name.*
- void **setParentTag** (int parentTagId)  
*Set a tag which may by the user be intended to be the parent of a newly created tag.*
- QString **suggestedUIString** (const [TaggingAction](#) &action) const  
*Returns a string to be used in the UI for the given [TaggingAction](#), interpreted in the context of the current settings.*

### Static Public Member Functions

- static [TaggingAction](#) **defaultTaggingAction** (const QString &tagName, int parentTagId=0)

## 6.1401.1 Member Enumeration Documentation

### 6.1401.1.1 NameMatchMode

enum [Digikam::TaggingActionFactory::NameMatchMode](#)

#### Enumerator

MatchStartingWithFragment	Default: use the "startingWith" method.
MatchContainingFragment	use the "contains" method

## 6.1401.2 Member Function Documentation

### 6.1401.2.1 setConstraintInterface()

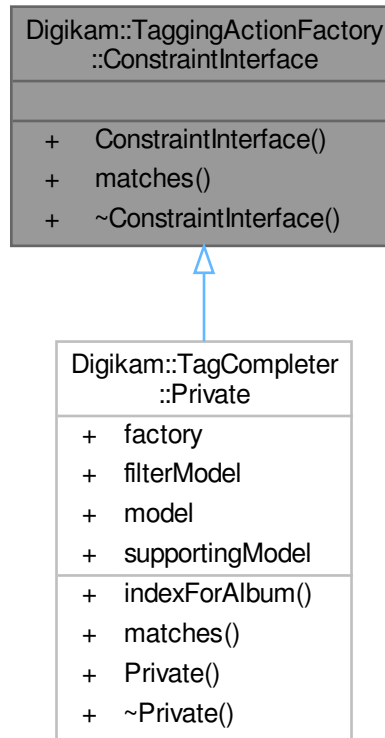
```
void Digikam::TaggingActionFactory::setConstraintInterface (
    ConstraintInterface *const iface )
```

Allows to filter the scope of suggested tags. Pass an implementation of [ConstraintInterface](#) (reamins in your ownership). [actions\(\)](#) will then only suggest to assign tags for which [matches\(\)](#) is true



## 6.1402 Digikam::TaggingActionFactory::ConstraintInterface Class Reference

Inheritance diagram for Digikam::TaggingActionFactory::ConstraintInterface:



### Public Member Functions

- virtual bool **matches** (int tagId)=0

## 6.1403 Digikam::TagInfo Class Reference

### Public Types

- typedef QList< [TagInfo](#) > **List**

### Public Member Functions

- bool **isNull** () const
- bool **operator**< (const [TagInfo](#) &info) const

## Public Attributes

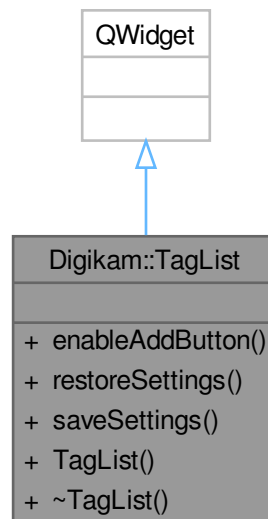
- QString **icon**
- qlonglong **iconId** = 0
- int **id** = 0
- QString **name**
- int **pid** = 0

### 6.1403.1 Detailed Description

A container class for transporting tag information from the database to [AlbumManager](#)

## 6.1404 Digikam::TagList Class Reference

Inheritance diagram for Digikam::TagList:



## Public Member Functions

- void **enableAddButton** (bool value)  
*enableAddButton* - disable Add Button when selection is empty or only root tag is selected
- void **restoreSettings** ()  
*restoreSettings* - read settings from `digikam_tagsmanagerrc` config and populate model with data
- void **saveSettings** ()  
*saveSettings* - save settings to `digiKam_tagsmanagerrc KConfig`
- **TagList** ([TagMgrTreeView](#) \*const treeView, QWidget \*const parent)

## 6.1404.1 Member Function Documentation

### 6.1404.1.1 restoreSettings()

```
void Digikam::TagList::restoreSettings ( )
```

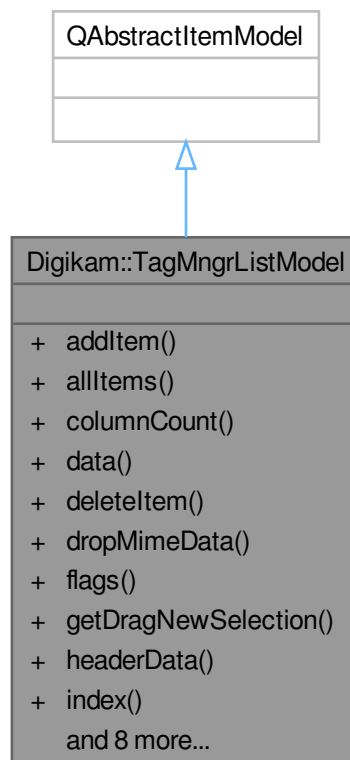
If config is empty add generic All Tags

Use this map to find all List Items that contain specific tag usually to remove deleted tag

"All Tags" item should be selected

## 6.1405 Digikam::TagMgrListModel Class Reference

Inheritance diagram for Digikam::TagMgrListModel:



## Public Member Functions

- `ListItem * addItem (QList< QVariant > values)`  
*addItem* - add new item to list
- `QList< ListItem * > allItems () const`  
*allItems* - return all items from List, usually to be saved in KConfig
- `int columnCount (const QModelIndex &parent=QModelIndex()) const override`
- `QVariant data (const QModelIndex &index, int role) const override`
- `void deleteItem (ListItem *const item)`
- `bool dropMimeData (const QMimeData *data, Qt::DropAction action, int row, int column, const QModelIndex &parent) override`
- `Qt::ItemFlags flags (const QModelIndex &index) const override`
- `QList< int > getDragNewSelection () const`
- `QVariant headerData (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override`
- `QModelIndex index (int row, int column, const QModelIndex &parent=QModelIndex()) const override`
- `QMimeData * mimeData (const QModelIndexList &indexes) const override`
- `QStringList mimeTypes () const override`
- `QModelIndex parent (const QModelIndex &index) const override`
- `int rowCount (const QModelIndex &parent=QModelIndex()) const override`
- `bool setData (const QModelIndex &index, const QVariant &value, int role) override`
- `Qt::DropActions supportedDropActions () const override`
- `TagMngrListModel (QObject *const parent=nullptr)`

## 6.1405.1 Member Function Documentation

### 6.1405.1.1 addItem()

```
ListItem * Digikam::TagMngrListModel::addItem (
    QList< QVariant > values )
```

#### Parameters

<i>values</i>	- A list of data for item: Name as QString, QBrush as background and qlonglong as id
---------------	--

#### Returns

- pointer to newly created listitem

containsItem will return a valid pointer if item with the same values is already added to it's children list.

### 6.1405.1.2 data()

```
QVariant Digikam::TagMngrListModel::data (
    const QModelIndex & index,
    int role ) const [override]
```

Standard methods to be implemented when subclassing QAbstractListModel

### 6.1405.1.3 dropMimeData()

```
bool Digikam::TagMngrListModel::dropMimeData (
    const QMimeData * data,
    Qt::DropAction action,
    int row,
    int column,
    const QModelIndex & parent ) [override]
```

After drag-n-drop selection is messed up, store the interval were new items are and TagsMngrListView will update selection

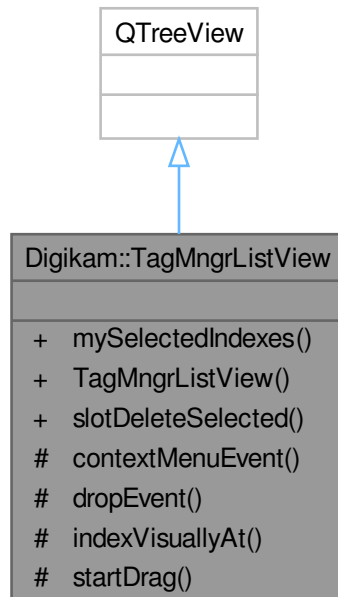
### 6.1405.1.4 supportedDropActions()

```
Qt::DropActions Digikam::TagMngrListModel::supportedDropActions ( ) const [override]
```

Reimplemented methods for handling drag-n-drop, encoding and decoding mime types

## 6.1406 Digikam::TagMngrListView Class Reference

Inheritance diagram for Digikam::TagMngrListView:



### Public Slots

- void **slotDeleteSelected** ()  
*slotDeleteSelected* - delete selected item from Quick Access List

## Public Member Functions

- QModelIndexList **mySelectedIndexes** ()
- **TagMngrListView** (QWidget \*const parent=nullptr)

## Protected Member Functions

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*contextMenuEvent - reimplemented method from QListView to handle custom context menu*
- void **dropEvent** (QDropEvent \*e) override
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **startDrag** (Qt::DropActions supportedActions) override

## 6.1406.1 Member Function Documentation

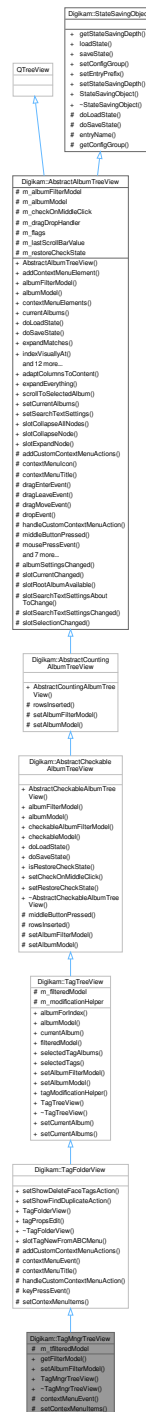
### 6.1406.1.1 startDrag()

```
void Digikam::TagMngrListView::startDrag (  
    Qt::DropActions supportedActions ) [override], [protected]
```

Reimplemented methods to enable custom drag-n-drop in QListView

## 6.1407 Digikam::TagMgrTreeView Class Reference

Inheritance diagram for Digikam::TagMgrTreeView:



### Public Member Functions

- [TagsManagerFilterModel](#) \* **getFilterModel** () const
- void **setAlbumFilterModel** ([TagsManagerFilterModel](#) \*const [filteredModel](#), [CheckableAlbumFilterModel](#) \*const [filterModel](#))

*setAlbumFilterModel* reimplement from *AbstractAlbumTree*

- **TagMngrTreeView** ([TagsManager](#) \*const parent, [TagModel](#) \*const model)

## Public Member Functions inherited from [Digikam::TagFolderView](#)

- void [setShowDeleteFaceTagsAction](#) (bool show)
- void [setShowFindDuplicateAction](#) (bool show)
- [TagFolderView](#) (QWidget \*const parent, [TagModel](#) \*const model)
- void [tagPropsEdit](#) ()
- [~TagFolderView](#) () override

## Public Member Functions inherited from [Digikam::TagTreeView](#)

- [TAlbum](#) \* **albumForIndex** (const QModelIndex &index) const
- [TagModel](#) \* **albumModel** () const
- [TAlbum](#) \* **currentAlbum** () const  
*currentAlbum* Even if multiple selection is enabled current *Album* can be only one, the last clicked item if you need selected items, see *selectedAlbums()* It's NOT the same as *AlbumManager::currentAlbums()*
- [TagPropertiesFilterModel](#) \* **filteredModel** () const
- QList< [TAlbum](#) \* > **selectedTagAlbums** ()
- QList< [Album](#) \* > **selectedTags** ()  
*selectedTags* - return a list of all selected items in tag model
- void **setAlbumFilterModel** ([TagPropertiesFilterModel](#) \*const *filteredModel*, [CheckableAlbumFilterModel](#) \*const *filterModel*)
- void **setAlbumModel** ([TagModel](#) \*const model)
- [TagModificationHelper](#) \* **tagModificationHelper** () const
- **TagTreeView** (QWidget \*const parent=nullptr, Flags flags=DefaultFlags)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- [CheckableAlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractCheckableAlbumModel](#) \* **albumModel** () const
- [CheckableAlbumFilterModel](#) \* **checkableAlbumFilterModel** () const
- [AbstractCheckableAlbumModel](#) \* **checkableModel** () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool doThat)
- void [setRestoreCheckState](#) (bool restore)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- **AbstractCountingAlbumTreeView** (QWidget \*const parent, Flags flags)



## Public Member Functions inherited from Digikam::AbstractAlbumTreeView

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from Digikam::StateSavingObject

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Protected Member Functions

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*contextMenuEvent Reimplement contextMenuEvent from AbstractAlbumTree to support multiple selection*
- void **setContextMenuItems** ([ContextMenuHelper](#) &cmh, const QList< [TAlbum](#) \* > &albums) override  
*setContextMenuItems Reimplemented method from TagsFolderView. Will set custom actions for Tags Manager. Some actions are also available in toolbar*

## Protected Member Functions inherited from Digikam::TagFolderView

- void **addCustomContextMenuActions** ([ContextMenuHelper](#) &cmh, [Album](#) \*album) override
- void **contextMenuEvent** (QContextMenuEvent \*event) override
- QString **contextMenuTitle** () const override
- void **handleCustomContextMenuAction** (QAction \*action, const [AlbumPointer](#)< [Album](#) > &album) override
- void **keyPressEvent** (QKeyEvent \*event) override

### Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual [QPixmap](#) [contextMenuIcon](#) () const
  - void [dragEnterEvent](#) ([QDragEnterEvent](#) \*e) override
  - void [dragLeaveEvent](#) ([QDragLeaveEvent](#) \*e) override
  - void [dragMoveEvent](#) ([QDragMoveEvent](#) \*e) override
  - void [dropEvent](#) ([QDropEvent](#) \*e) override
  - void [mousePressEvent](#) ([QMouseEvent](#) \*e) override
- Other helper methods.*
- virtual [QPixmap](#) [pixmapForDrag](#) (const [QStyleOptionViewItem](#) &option, [QList](#)< [QModelIndex](#) > indexes)
  - void [rowsAboutToBeRemoved](#) (const [QModelIndex](#) &parent, int start, int end) override
  - void [rowsInserted](#) (const [QModelIndex](#) &index, int start, int end) override
  - void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
  - void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
  - virtual bool [showContextMenuAt](#) ([QContextMenuEvent](#) \*event, [Album](#) \*albumForEvent)
  - void [startDrag](#) ([Qt::DropActions](#) supportedActions) override

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- [QString](#) [entryName](#) (const [QString](#) &base) const
- [KConfigGroup](#) [getConfigGroup](#) () const

### Protected Attributes

- [TagsManagerFilterModel](#) \* [m\\_filteredModel](#) = nullptr

### Protected Attributes inherited from [Digikam::TagTreeView](#)

- [TagPropertiesFilterModel](#) \* [m\\_filteredModel](#) = nullptr
- [TagModificationHelper](#) \* [m\\_modificationHelper](#) = nullptr

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* **m\_albumFilterModel** = nullptr
- [AbstractSpecificAlbumModel](#) \* **m\_albumModel** = nullptr
- bool **m\_checkOnMiddleClick** = false
- [AlbumModelDragDropHandler](#) \* **m\_dragDropHandler** = nullptr
- Flags **m\_flags** = DefaultFlags
- int **m\_lastScrollBarValue** = 0
- bool **m\_restoreCheckState** = false

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) {  
[CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) ,  
[AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Public Slots inherited from [Digikam::TagFolderView](#)

- void **slotTagNewFromABCMenu** (const QString &personName)

## Public Slots inherited from [Digikam::TagTreeView](#)

- void **setCurrentAlbum** (int tagId, bool selectInAlbumManager=true)
- void **setCurrentAlbums** (const QList< [Album](#) \* > &tags, bool selectInAlbumManager=true)

## Public Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void **adaptColumnsToContent** ()
- void **expandEverything** (const QModelIndex &index)
- void **scrollToSelectedAlbum** ()
- void **setCurrentAlbums** (const QList< [Album](#) \* > &albums, bool selectInAlbumManager=true)
- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)
- void **slotCollapseAllNodes** ()  
*slotCollapseAllNodes - collapse all nodes without root node*
- void **slotCollapseNode** ()  
*slotCollapseNode - collapse recursively selected nodes*
- void **slotExpandNode** ()  
*slotExpandNode - expands recursively selected nodes*

## Signals inherited from [Digikam::TagFolderView](#)

- void **signalFindDuplicates** (const QList< [TAlbum](#) \* > &albums)

## Signals inherited from [Digikam::TagTreeView](#)

- void **assignTags** (int tagId, const QList< int > &imageIDs)

## Signals inherited from [Digikam::AbstractAlbumTreeView](#)

- void [currentAlbumChanged](#) ([Album](#) \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< [Album](#) \* > &selectedAlbums)

## Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void **albumSettingsChanged** ()
- void **slotCurrentChanged** ()
- virtual void [slotRootAlbumAvailable](#) ()
- void **slotSearchTextSettingsAboutToChange** (bool searched, bool willSearch)
- void **slotSearchTextSettingsChanged** (bool wasSearching, bool searching)
- void **slotSelectionChanged** ()

## 6.1407.1 Member Function Documentation

### 6.1407.1.1 contextMenuEvent()

```
void Digikam::TagMngrTreeView::contextMenuEvent (
    QContextMenuEvent * event ) [override], [protected]
```

#### Parameters

<i>event</i>	context menu event triggered by right click
--------------	---

Append root tag if no nodes are selected

### 6.1407.1.2 setContextMenuItems()

```
void Digikam::TagMngrTreeView::setContextMenuItems (
    ContextMenuHelper & cmh,
    const QList< TAlbum * > & albums ) [override], [protected], [virtual]
```

#### Parameters

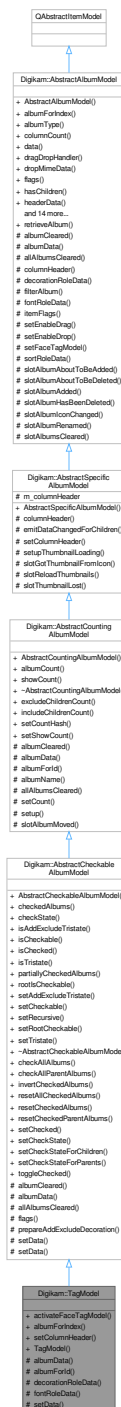
<i>cmh</i>	<a href="#">ContextMenuHelper</a> class to help setting some basic actions
<i>albums</i>	List of currently selected albums

This is a dummy action, delete is disable for root tag

Reimplemented from [Digikam::TagFolderView](#).

## 6.1408 Digikam::TagModel Class Reference

Inheritance diagram for Digikam::TagModel:



### Public Member Functions

- void **activateFaceTagModel** ()
- **Album** \* **albumForIndex** (const QModelIndex &index) const
- void **setColumnHeader** (const QString &header)
- **TagModel** (**RootAlbumBehavior** rootBehavior=**IncludeRootAlbum**, QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::AbstractCheckableAlbumModel](#)

- [AbstractCheckableAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [QList](#)< [Album](#) \* > **checkedAlbums** () const  
*Returns a list of album with check state Checked.*
- [Qt::CheckState](#) **checkState** ([Album](#) \*album) const  
*Returns the check state of the album.*
- bool **isAddExcludeTristate** () const
- bool **isCheckable** () const
- bool **isChecked** ([Album](#) \*album) const  
*Returns if the given album has the check state Checked.*
- bool **isTristate** () const
- [QList](#)< [Album](#) \* > **partiallyCheckedAlbums** () const  
*Returns a list of album with partially check state Checked.*
- bool **rootsCheckable** () const
- void **setAddExcludeTristate** (bool b)
- void **setCheckable** (bool isCheckable)  
*Triggers if the albums in this model are checkable.*
- void **setRecursive** (bool recursive)
- void **setRootCheckable** (bool rootsCheckable)
- void **setTristate** (bool isTristate)

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumModel](#)

- [AbstractCountingAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Supports displaying a count alongside the album name in DisplayRole.*
- virtual int **albumCount** ([Album](#) \*album) const
- bool **showCount** () const

## Public Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- [AbstractSpecificAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)  
*Abstract base class, do not instantiate.*

## Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- [AbstractAlbumModel](#) ([Album::Type](#) albumType, [Album](#) \*const rootAlbum, [RootAlbumBehavior](#) rootBehavior=[IncludeRootAlbum](#), [QObject](#) \*const parent=nullptr)
- [Album](#) \* **albumForIndex** (const [QModelIndex](#) &index) const
- [Album::Type](#) **albumType** () const
- int **columnCount** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **data** (const [QModelIndex](#) &index, int role=[Qt::DisplayRole](#)) const override
- [AlbumModelDragDropHandler](#) \* **dragDropHandler** () const
- bool **dropMimeData** (const [QMimeData](#) \*data, [Qt::DropAction](#) action, int row, int column, const [QModelIndex](#) &parent) override
- [Qt::ItemFlags](#) **flags** (const [QModelIndex](#) &index) const override
- bool **hasChildren** (const [QModelIndex](#) &parent=[QModelIndex](#)()) const override
- [QVariant](#) **headerData** (int section, [Qt::Orientation](#) orientation, int role=[Qt::DisplayRole](#)) const override

- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex **indexForAlbum** (Album \*album) const
- bool **isFaceTagModel** () const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const override
- QStringList **mimeTypes** () const override
- QModelIndex **parent** (const QModelIndex &index) const override
- Album \* **rootAlbum** () const
- RootAlbumBehavior **rootAlbumBehavior** () const
- QModelIndex **rootAlbumIndex** () const
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- void **setDragDropHandler** (AlbumModelDragDropHandler \*handler)
- void **setDropIndex** (const QModelIndex &index)
- Qt::DropActions **supportedDropActions** () const override

### Protected Member Functions

- QVariant **albumData** (Album \*a, int role) const override  
*For subclassing convenience: A part of the implementation of data()*
- Album \* **albumForId** (int id) const override  
*need to implement in subclass*
- QVariant **decorationRoleData** (Album \*a) const override  
*For subclassing convenience: A part of the implementation of data()*
- QVariant **fontRoleData** (Album \*a) const override  
*For subclassing convenience: A part of the implementation of data()*
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override

### Protected Member Functions inherited from Digikam::AbstractCheckableAlbumModel

- void **albumCleared** (Album \*album) override  
*Notification when an entry is removed.*
- void **allAlbumsCleared** () override  
*Notification when all entries are removed.*
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- void **prepareAddExcludeDecoration** (Album \*a, QPixmap &icon) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role, bool recursive)
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override

### Protected Member Functions inherited from Digikam::AbstractCountingAlbumModel

- void **albumCleared** (Album \*album) override  
*Notification when an entry is removed.*
- virtual QString **albumName** (Album \*a) const  
*Can reimplement in subclass.*
- void **allAlbumsCleared** () override  
*Notification when all entries are removed.*
- void **setCount** (Album \*album, int count)  
*If you do not use setCountHash, excludeChildrenCount and includeChildrenCount, you can set a count here.*
- void **setup** ()

## Protected Member Functions inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString [columnHeader](#) () const override  
*For subclassing convenience: A part of the implementation of [headerData\(\)](#)*
- void **emitDataChangedForChildren** ([Album](#) \*album)
- void **setColumnHeader** (const QString &header)
- void **setupThumbnailLoading** ()  
*You need to call this from your constructor if you intend to load the thumbnail facilities of this class.*

## Protected Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- virtual bool [filterAlbum](#) ([Album](#) \*album) const
- virtual Qt::ItemFlags **itemFlags** ([Album](#) \*album) const  
*For subclassing convenience: A part of the implementation of [itemFlags\(\)](#)*
- void [setEnabledDrag](#) (bool enable)
- void **setEnabledDrop** (bool enable)
- void **setFaceTagModel** (bool enable)
- virtual QVariant [sortRoleData](#) ([Album](#) \*a) const  
*For subclassing convenience: A part of the implementation of [data\(\)](#)*

## Additional Inherited Members

## Public Types inherited from [Digikam::AbstractAlbumModel](#)

- enum [AlbumDataRole](#) {  
[AlbumTitleRole](#) = Qt::UserRole , [AlbumTypeRole](#) = Qt::UserRole + 1 , [AlbumPointerRole](#) = Qt::UserRole + 2  
, [AlbumIdRole](#) = Qt::UserRole + 3 ,  
[AlbumGlobalIdRole](#) = Qt::UserRole + 4 , [AlbumSortRole](#) = Qt::UserRole + 5 }
- enum [RootAlbumBehavior](#) { [IncludeRootAlbum](#) , [IgnoreRootAlbum](#) }

## Public Slots inherited from [Digikam::AbstractCheckableAlbumModel](#)

- void **checkAllAlbums** (const QModelIndex &parent=QModelIndex())  
*Checks all albums beneath the given parent.*
- void **checkAllParentAlbums** (const QModelIndex &child)  
*Checks all parent albums starting at the child, including it.*
- void **invertCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Inverts the checked state of all albums under the given parent.*
- void **resetAllCheckedAlbums** ()  
*Resets the checked state of all albums to Qt::Unchecked.*
- void **resetCheckedAlbums** (const QModelIndex &parent=QModelIndex())  
*Resets the checked state of all albums under the given parent.*
- void **resetCheckedParentAlbums** (const QModelIndex &child)  
*Resets the checked state of all parents of the child including it.*
- void **setChecked** ([Album](#) \*album, bool [isChecked](#))  
*Sets the check state of album to Checked or Unchecked.*
- void **setCheckState** ([Album](#) \*album, Qt::CheckState state)  
*Sets the check state of the album.*
- void **setCheckStateForChildren** ([Album](#) \*album, Qt::CheckState state)  
*Sets the checked state recursively for all children of but not for the given album.*
- void **setCheckStateForParents** ([Album](#) \*album, Qt::CheckState state)  
*Sets the checked state recursively for all parents of but not for the given album.*
- void **toggleChecked** ([Album](#) \*album)  
*Toggles the check state of album between Checked or Unchecked.*



### Public Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [excludeChildrenCount](#) (const QModelIndex &index)
- void [includeChildrenCount](#) (const QModelIndex &index)
- void [setCountHash](#) (const QHash< int, int > &idCountHash)
- void [setShowCount](#) (bool show)

*Call to enable or disable showing the count. Default is false.*

### Signals inherited from [Digikam::AbstractCheckableAlbumModel](#)

- void [checkStateChanged](#) (Album \*album, Qt::CheckState checkState)

### Signals inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [signalUpdateAlbumCount](#) (Album \*album)

### Signals inherited from [Digikam::AbstractAlbumModel](#)

- void [rootAlbumAvailable](#) ()

### Static Public Member Functions inherited from [Digikam::AbstractAlbumModel](#)

- static Album \* [retrieveAlbum](#) (const QModelIndex &index)

### Protected Slots inherited from [Digikam::AbstractCountingAlbumModel](#)

- void [slotAlbumMoved](#) (Album \*album)

### Protected Slots inherited from [Digikam::AbstractSpecificAlbumModel](#)

- void [slotGotThumbnailFromIcon](#) (Album \*album, const QPixmap &thumbnail)
- void [slotReloadThumbnails](#) ()
- void [slotThumbnailLost](#) (Album \*album)

### Protected Slots inherited from [Digikam::AbstractAlbumModel](#)

- void [slotAlbumAboutToBeAdded](#) (Album \*album, Album \*parent, Album \*prev)
- void [slotAlbumAboutToBeDeleted](#) (Album \*album)
- void [slotAlbumAdded](#) (Album \*)
- void [slotAlbumHasBeenDeleted](#) (Album \*album)
- void [slotAlbumIconChanged](#) (Album \*album)
- void [slotAlbumRenamed](#) (Album \*album)
- void [slotAlbumsCleared](#) ()

### Protected Attributes inherited from [Digikam::AbstractSpecificAlbumModel](#)

- QString [m\\_columnHeader](#)

## 6.1408.1 Constructor & Destructor Documentation

### 6.1408.1.1 TagModel()

```
Digikam::TagModel::TagModel (
    RootAlbumBehavior rootBehavior = IncludeRootAlbum,
    QObject *const parent = nullptr ) [explicit]
```

Create a model containing all tags

## 6.1408.2 Member Function Documentation

### 6.1408.2.1 albumData()

```
QVariant Digikam::TagModel::albumData (
    Album * a,
    int role ) const [override], [protected], [virtual]
```

#### Note

these can be reimplemented in a subclass

Reimplemented from [Digikam::AbstractCheckableAlbumModel](#).

### 6.1408.2.2 albumForId()

```
Album * Digikam::TagModel::albumForId (
    int id ) const [override], [protected], [virtual]
```

Implements [Digikam::AbstractCountingAlbumModel](#).

### 6.1408.2.3 decorationRoleData()

```
QVariant Digikam::TagModel::decorationRoleData (
    Album * a ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).

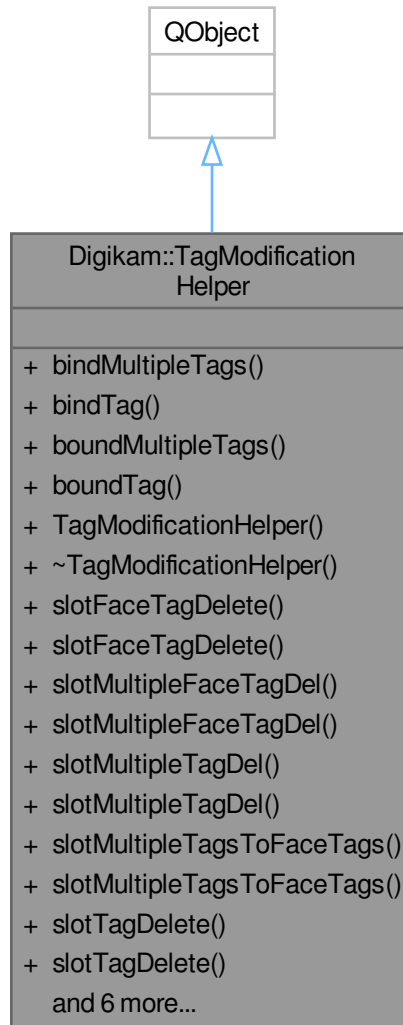
### 6.1408.2.4 fontRoleData()

```
QVariant Digikam::TagModel::fontRoleData (
    Album * a ) const [override], [protected], [virtual]
```

Reimplemented from [Digikam::AbstractAlbumModel](#).

## 6.1409 Digikam::TagModificationHelper Class Reference

Inheritance diagram for Digikam::TagModificationHelper:



### Public Slots

- void `slotFaceTagDelete ()`
- void `slotFaceTagDelete (TAIbum *tag)`
- void `slotMultipleFaceTagDel ()`
- void `slotMultipleFaceTagDel (const QList< TAIbum * > &tags)`
- void `slotMultipleTagDel ()`
- void `slotMultipleTagDel (const QList< TAIbum * > &tags)`  
*must use bindTag and a QAction*
- void `slotMultipleTagsToFaceTags ()`
- void `slotMultipleTagsToFaceTags (const QList< TAIbum * > &tags)`

- void `slotTagDelete` ()
- void `slotTagDelete` (TAIbum \*tag)
- void `slotTagEdit` ()
- void `slotTagEdit` (TAIbum \*tag)
- TAIbum \* `slotTagNew` ()
- TAIbum \* `slotTagNew` (TAIbum \*parent, const QString &title=QString(), const QString &iconName=QString())
- void `slotTagToFaceTag` ()
- void `slotTagToFaceTag` (TAIbum \*tag)

## Signals

- void `aboutToDeleteTag` (TAIbum \*tag)
- void `tagCreated` (TAIbum \*tag)
- void `tagEdited` (TAIbum \*tag)

## Public Member Functions

- void `bindMultipleTags` (QAction \*action, const QList< TAIbum \* > &tags)
- void `bindTag` (QAction \*action, TAIbum \*parent) const
- QList< TAIbum \* > `boundMultipleTags` (QObject \*sender)
- TAIbum \* `boundTag` (QObject \*action) const
- `TagModificationHelper` (QObject \*const parent, QWidget \*const dialogParent)
- `~TagModificationHelper` () override

### 6.1409.1 Detailed Description

Utility class providing methods to modify tag albums (TAIbum) in a way useful to implement views.

This class can do background processing for batch tag operations. So be sure that the signals indicating the progress of these operations are used.

#### Author

jwienke

### 6.1409.2 Constructor & Destructor Documentation

#### 6.1409.2.1 TagModificationHelper()

```
Digikam::TagModificationHelper::TagModificationHelper (
    QObject *const parent,
    QWidget *const dialogParent ) [explicit]
```

Constructor.

#### Parameters

<i>parent</i>	the parent for qt parent child mechanism
<i>dialogParent</i>	parent widget for dialogs displayed by this object

### 6.1409.2.2 ~TagModificationHelper()

```
Digikam::TagModificationHelper::~TagModificationHelper ( ) [override]
```

Destructor.

## 6.1409.3 Member Function Documentation

### 6.1409.3.1 bindMultipleTags()

```
void Digikam::TagModificationHelper::bindMultipleTags (
    QAction * action,
    const QList< TAlbum * > & tags )
```

Set QVector's pointer into action's data. Make sure that QVector is not a local object and it's not destroyed before boundMultipleTags are called

#### Parameters

<i>action</i>	- action to store pointer
<i>tags</i>	- QVector pointer to be stored

### 6.1409.3.2 bindTag()

```
void Digikam::TagModificationHelper::bindTag (
    QAction * action,
    TAlbum * parent ) const
```

Sets the tag that the given action operates on. You must call bindTag and then connect the action's triggered to the desired slot, [slotTagNew\(\)](#), [slotTagEdit\(\)](#) or [slotTagDelete\(\)](#). Note: Changes the Action's user data.

### 6.1409.3.3 boundMultipleTags()

```
QList< TAlbum * > Digikam::TagModificationHelper::boundMultipleTags (
    QObject * sender )
```

Return QVector pointer bound with bindMultipleTags. Use when context menu should delete more than one item: multiple-selection.

### 6.1409.3.4 boundTag()

```
TAlbum * Digikam::TagModificationHelper::boundTag (
    QObject * action ) const
```

Returns the tag bound with bindTag. The given QObject shall be a QAction, but for convenience the given object will be checked with `qobject_cast` first, so you can pass `QObject::sender()`.

### 6.1409.3.5 slotFaceTagDelete [1/2]

```
void Digikam::TagModificationHelper::slotFaceTagDelete ( ) [slot]
```

must use bindTag and a QAction

### 6.1409.3.6 slotFaceTagDelete [2/2]

```
void Digikam::TagModificationHelper::slotFaceTagDelete (
    TAlbum * tag ) [slot]
```

Deletes the given face tag and after prompting the user for this. The tag itself is not deleted. Only its property as face tag.

#### Parameters

<i>tag</i>	the face tag to delete
------------	------------------------

### 6.1409.3.7 slotMultipleFaceTagDel [1/2]

```
void Digikam::TagModificationHelper::slotMultipleFaceTagDel ( ) [slot]
```

must use bindMultipleTags and a QAction

### 6.1409.3.8 slotMultipleFaceTagDel [2/2]

```
void Digikam::TagModificationHelper::slotMultipleFaceTagDel (
    const QList< TAlbum * > & tags ) [slot]
```

Delete multiple face tags and prompt user only once for all The tags itself are not deleted. Only their properties as face tags.

#### Parameters

<i>tags</i>	face tags to be deleted.
-------------	--------------------------

### 6.1409.3.9 slotMultipleTagDel [1/2]

```
void Digikam::TagModificationHelper::slotMultipleTagDel ( ) [slot]
```

must use bindMultipleTags and a QAction

### 6.1409.3.10 slotMultipleTagDel [2/2]

```
void Digikam::TagModificationHelper::slotMultipleTagDel (
    const QList< TAlbum * > & tags ) [slot]
```

Delete multiple tags and prompt user only once for all

**Parameters**

<i>tags</i>	the tags to be deleted, without root tag
-------------	--

Tags must be deleted from children to parents, if we don't want to step on invalid index. Use QMap to order them by distance to root tag

QMultimap doesn't provide reverse iterator, use QList.

**6.1409.3.11 slotMultipleTagsToFaceTags [1/2]**

```
void Digikam::TagModificationHelper::slotMultipleTagsToFaceTags ( ) [slot]
```

must use bindMultipleTags and a QAction

**6.1409.3.12 slotMultipleTagsToFaceTags [2/2]**

```
void Digikam::TagModificationHelper::slotMultipleTagsToFaceTags (
    const QList< TAlbum * > & tags ) [slot]
```

Marks the tags as face tags if they are not already.

**Parameters**

<i>tags</i>	the tags to mark.
-------------	-------------------

**6.1409.3.13 slotTagDelete [1/2]**

```
void Digikam::TagModificationHelper::slotTagDelete ( ) [slot]
```

must use bindTag and a QAction

**6.1409.3.14 slotTagDelete [2/2]**

```
void Digikam::TagModificationHelper::slotTagDelete (
    TAlbum * tag ) [slot]
```

Deletes the given tag and after prompting the user for this.

**Parameters**

<i>tag</i>	the tag to delete, must not be the root tag album
------------	---

**6.1409.3.15 slotTagEdit [1/2]**

```
void Digikam::TagModificationHelper::slotTagEdit ( ) [slot]
```

must use bindTag and a QAction

**6.1409.3.16 slotTagEdit [2/2]**

```
void Digikam::TagModificationHelper::slotTagEdit (
    TAlbum * tag ) [slot]
```

Edits the given tag via a user dialog.

**Parameters**

<i>tag</i>	the tag to change
------------	-------------------

**6.1409.3.17 slotTagNew [1/2]**

```
TAlbum * Digikam::TagModificationHelper::slotTagNew ( ) [slot]
```

Same as above, but this slot can be triggered from a QAction if a parent tag is bound to this action, see below. Without this mechanism, will add a toplevel tag.

**Returns**

new tag created or 0 if no tag was created

**6.1409.3.18 slotTagNew [2/2]**

```
TAlbum * Digikam::TagModificationHelper::slotTagNew (
    TAlbum * parent,
    const QString & title = QString(),
    const QString & iconName = QString() ) [slot]
```

Creates one ore more new tags under the given parent. If only the parent is given, then a dialog is shown to create new tags. Else, if also a title and optionally an icon are given, then these values will be used directly to create the tag.

**Parameters**

<i>parent</i>	the parent tag album under which to create the new tags. May be 0 to use the root album
<i>title</i>	if this isn't an empty string, then this tag name is suggested
<i>iconName</i>	an optional name for the icon to suggest for the new tag



**Returns**

new tag album or 0 if not created

**6.1409.3.19 slotTagToFaceTag [1/2]**

```
void Digikam::TagModificationHelper::slotTagToFaceTag ( ) [slot]
```

must use bindTag and a QAction

**6.1409.3.20 slotTagToFaceTag [2/2]**

```
void Digikam::TagModificationHelper::slotTagToFaceTag (
    TAlbum * tag ) [slot]
```

Marks the tag as face tag if it is not already.

**Parameters**

<i>tag</i>	the tag to mark
------------	-----------------

**6.1410 Digikam::TagProperties Class Reference****Public Member Functions**

- void [addProperty](#) (const QString &key, const QString &value)
- bool [hasProperty](#) (const QString &key) const
- bool [hasProperty](#) (const QString &key, const QString &value) const
- bool [isNull](#) () const
- [TagProperties](#) & [operator=](#) (const [TagProperties](#) &other)
- QMap< QString, QString > [properties](#) () const  
*Returns a map of all key->value pairs.*
- QStringList [propertyKeys](#) () const  
*Returns all set property keys.*
- void [removeProperties](#) (const QString &key)  
*Remove all occurrences of the property.*
- void [removeProperty](#) (const QString &key, const QString &value)  
*Remove the given property/value.*
- void [setProperty](#) (const QString &key, const QString &value)  
*Set the given property. Replaces all previous occurrences of this property.*
- int [tagId](#) () const
- [TagProperties](#) ()
- [TagProperties](#) (const [TagProperties](#) &other)
- [TagProperties](#) (int tagId)
- [TagProperties](#) (const QString &key, const QString &value) const

## Static Public Member Functions

- static [TagProperties](#) `getOrCreate` (const QString &tagPath)

## 6.1410.1 Constructor & Destructor Documentation

### 6.1410.1.1 TagProperties() [1/2]

```
Digikam::TagProperties::TagProperties ( )
```

This class provides a wrapper over the Database methods to access the properties of a tag. It is meant to be a short-lived object, it does not listen to external database changes.

### 6.1410.1.2 TagProperties() [2/2]

```
Digikam::TagProperties::TagProperties (
    int tagId ) [explicit]
```

Access the properties of the given tag.

## 6.1410.2 Member Function Documentation

### 6.1410.2.1 addProperty()

```
void Digikam::TagProperties::addProperty (
    const QString & key,
    const QString & value )
```

Adds the given property. Does not change any previous occurrences of this property, allowing multiple properties with the same key.

### 6.1410.2.2 getOrCreate()

```
TagProperties Digikam::TagProperties::getOrCreate (
    const QString & tagPath ) [static]
```

Finds the tag for the given tag path or creates a new tag. Then returns the tag properties for this tag.

### 6.1410.2.3 hasProperty() [1/2]

```
bool Digikam::TagProperties::hasProperty (
    const QString & key ) const
```

Returns true if the property is set.

#### 6.1410.2.4 hasProperty() [2/2]

```
bool Digikam::TagProperties::hasProperty (
    const QString & key,
    const QString & value ) const
```

Returns true if the property is set, with exactly the given value.

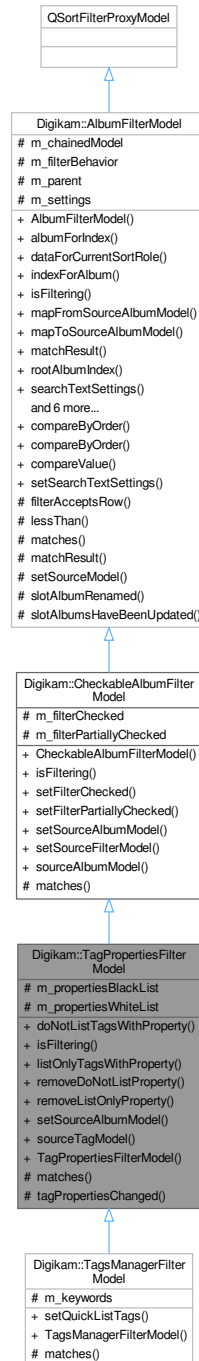
#### 6.1410.2.5 value()

```
QString Digikam::TagProperties::value (
    const QString & key ) const
```

Returns the value of the given property. If the property is not set, a null string is returned. But a null string is also returned if the property is set, but without a value. Use hasProperty to check that case.

## 6.1411 Digikam::TagPropertiesFilterModel Class Reference

Inheritance diagram for Digikam::TagPropertiesFilterModel:



### Public Member Functions

- void **doNotListTagsWithProperty** (const QString &property)
- bool **isFiltering** () const override

- void **listOnlyTagsWithProperty** (const QString &property)
- void **removeDoNotListProperty** (const QString &property)
- void **removeListOnlyProperty** (const QString &property)
- void **setSourceAlbumModel** ([TagModel](#) \*const source)
- [TagModel](#) \* **sourceTagModel** () const
- **TagPropertiesFilterModel** (QObject \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::CheckableAlbumFilterModel](#)

- **CheckableAlbumFilterModel** (QObject \*const parent=nullptr)
- void **setFilterChecked** (bool filter)
- void **setFilterPartiallyChecked** (bool filter)
- void **setSourceAlbumModel** ([AbstractCheckableAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([CheckableAlbumFilterModel](#) \*const source)
- [AbstractCheckableAlbumModel](#) \* **sourceAlbumModel** () const

### Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- **AlbumFilterModel** (QObject \*const parent=nullptr)
- [Album](#) \* **albumForIndex** (const QModelIndex &index) const  
*Convenience methods.*
- QVariant **dataForCurrentSortRole** ([Album](#) \*album) const
- QModelIndex **indexForAlbum** ([Album](#) \*album) const
- QModelIndex **mapFromSourceAlbumModel** (const QModelIndex &index) const
- QModelIndex **mapToSourceAlbumModel** (const QModelIndex &index) const
- [MatchResult](#) **matchResult** (const QModelIndex &index) const
- QModelIndex **rootAlbumIndex** () const
- [SearchTextSettings](#) **searchTextSettings** () const
- void **setFilterBehavior** ([FilterBehavior](#) behavior)
- void **setSourceAlbumModel** ([AbstractAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([AlbumFilterModel](#) \*const source)
- [AbstractAlbumModel](#) \* **sourceAlbumModel** () const
- [AlbumFilterModel](#) \* **sourceFilterModel** () const
- void **updateFilter** ()

### Protected Slots

- void **tagPropertiesChanged** ([TAlbum](#) \*)

### Protected Slots inherited from [Digikam::AlbumFilterModel](#)

- void **slotAlbumRenamed** ([Album](#) \*album)
- void **slotAlbumsHaveBeenUpdated** (int type)

### Protected Member Functions

- bool **matches** ([Album](#) \*album) const override

## Protected Member Functions inherited from [Digikam::AlbumFilterModel](#)

- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- [MatchResult](#) **matchResult** ([Album](#) \*album) const
- void **setSourceModel** ([QAbstractItemModel](#) \*const model) override

## Protected Attributes

- [QSet](#)< [QString](#) > **m\_propertiesBlackList**
- [QSet](#)< [QString](#) > **m\_propertiesWhiteList**

## Protected Attributes inherited from [Digikam::CheckableAlbumFilterModel](#)

- bool **m\_filterChecked** = false
- bool **m\_filterPartiallyChecked** = false

## Protected Attributes inherited from [Digikam::AlbumFilterModel](#)

- [QPointer](#)< [AlbumFilterModel](#) > **m\_chainedModel** = nullptr
- [FilterBehavior](#) **m\_filterBehavior** = [FullFiltering](#)
- [QObject](#) \* **m\_parent** = nullptr
- [SearchTextSettings](#) **m\_settings**

## Additional Inherited Members

## Public Types inherited from [Digikam::AlbumFilterModel](#)

- enum [FilterBehavior](#) { [SimpleFiltering](#) , [FullFiltering](#) , [StrictFiltering](#) }
- enum [MatchResult](#) { [NoMatch](#) = 0 , [DirectMatch](#) , [ParentMatch](#) , [ChildMatch](#) , [SpecialMatch](#) }

## Public Slots inherited from [Digikam::AlbumFilterModel](#)

- void **setSearchTextSettings** (const [SearchTextSettings](#) &settings)

## Signals inherited from [Digikam::AlbumFilterModel](#)

- void **hasSearchResult** (bool hasResult)
- void **searchTextSettingsAboutToChange** (bool searched, bool willSearch)
- void **searchTextSettingsChanged** (bool wasSearching, bool searched)
- void **signalFilterChanged** ()

## Static Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- `template<typename T >`  
static int **compareByOrder** (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int [compareByOrder](#) (int compareResult, Qt::SortOrder sortOrder)
- `template<typename T >`  
static int [compareValue](#) (const T &a, const T &b)

### 6.1411.1 Detailed Description

[Filter](#) model for tags that can filter by tag property

### 6.1411.2 Member Function Documentation

#### 6.1411.2.1 [isFiltering\(\)](#)

```
bool Digikam::TagPropertiesFilterModel::isFiltering ( ) const [override], [virtual]
```

Returns if the currently applied filters will result in any filtering.

#### Returns

`true` if the current selected filter could result in any filtering without checking if this really happens.

Reimplemented from [Digikam::CheckableAlbumFilterModel](#).

#### 6.1411.2.2 [matches\(\)](#)

```
bool Digikam::TagPropertiesFilterModel::matches (
    Album * album ) const [override], [protected], [virtual]
```

This method provides the basic match checking algorithm. Return true if this single album matches the current criteria. This method can be overridden to provide custom filtering.

#### Parameters

<i>album</i>	the album to tell if it matches the filter criteria or not.
--------------	---

Reimplemented from [Digikam::CheckableAlbumFilterModel](#).

Reimplemented in [Digikam::TagsManagerFilterModel](#).

## 6.1412 Digikam::TagProperty Class Reference

### Public Member Functions

- bool **isNull** () const

**Public Attributes**

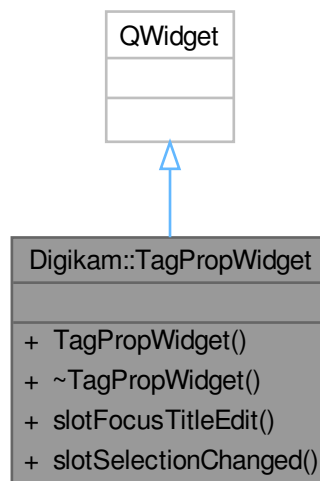
- QString **property**
- int **tagId** = -1
- QString **value**

**6.1413 Digikam::TagPropertyName Class Reference****Static Public Member Functions**

- static QLatin1String **faceEngineName** ()
- static QLatin1String **faceEngineUuid** ()
- static QLatin1String **ignoredPerson** ()
- static QLatin1String **person** ()
- static QLatin1String **tagKeyboardShortcut** ()
- static QLatin1String **unconfirmedPerson** ()
- static QLatin1String **unknownPerson** ()

**6.1414 Digikam::TagPropWidget Class Reference**

Inheritance diagram for Digikam::TagPropWidget:

**Public Types**

- enum **ItemsEnable** { **DisabledAll** , **EnabledAll** , **IconOnly** }



**Public Slots**

- void **slotFocusTitleEdit** ()
- void **slotSelectionChanged** (const QList< Album \* > &albums)

**Signals**

- void **signalTitleEditReady** ()

**Public Member Functions**

- **TagPropWidget** (QWidget \*const parent)

## 6.1415 Digikam::TagRegion Class Reference

**Public Types**

- enum **Type** { **Invalid** , **Rect** }

**Public Member Functions**

- bool **intersects** (const TagRegion &other, double fraction=0)
- bool **isValid** () const
- bool **operator!=** (const TagRegion &other) const
- bool **operator==** (const TagRegion &other) const
- TagRegion ()=default
- TagRegion (const QRect &rect)
- TagRegion (const QString &descriptor)
- QRect **toRect** () const
- QVariant **toVariant** () const
- QString **toXml** () const
- Type **type** () const

**Static Public Member Functions**

- static QRectF **absoluteToRelative** (const QRect &region, const QSize &fullSize)
- static QSize **adjustToOrientation** (QRect &region, int orientation, const QSize &fullSize)
- static TagRegion **fromVariant** (const QVariant &var)
- static QRect **mapFromOriginalSize** (const DImg &reducedSizeImage, const QRect &fullSizeDetail)
- static QRect **mapFromOriginalSize** (const QSize &fullImageSize, const QSize &reducedImageSize, const QRect &fullSizeDetail)
- static QRect **mapToOriginalSize** (const DImg &reducedSizeImage, const QRect &reducedSizeDetail)
- static QRect **mapToOriginalSize** (const QSize &fullImageSize, const QSize &reducedImageSize, const QRect &reducedSizeDetail)
- static QRect **relativeToAbsolute** (const QRectF &region, const DImg &reducedSizeImage)
- static QRect **relativeToAbsolute** (const QRectF &region, const QSize &fullSize)
- static void **reverseToOrientation** (QRect &region, int orientation, const QSize &fullSize)

## Protected Attributes

- Type `m_type` = Invalid
- QVariant `m_value`

## 6.1415.1 Constructor & Destructor Documentation

### 6.1415.1.1 TagRegion() [1/3]

```
Digikam::TagRegion::TagRegion ( ) [default]
```

Use this small class to convert between the formatted textual representation of a tag region in the database and the corresponding object. Construct an invalid region.

### 6.1415.1.2 TagRegion() [2/3]

```
Digikam::TagRegion::TagRegion (
    const QString & descriptor ) [explicit]
```

Construct with the textual descriptor.

### 6.1415.1.3 TagRegion() [3/3]

```
Digikam::TagRegion::TagRegion (
    const QRect & rect ) [explicit]
```

Construct with the region.

## 6.1415.2 Member Function Documentation

### 6.1415.2.1 absoluteToRelative()

```
QRectF Digikam::TagRegion::absoluteToRelative (
    const QRect & region,
    const QSize & fullSize ) [static]
```

Takes absolute region and full size to return the original relative region. Used to write back rectangles into image's XMP. see [MetadataHub::write](#).

### 6.1415.2.2 adjustToOrientation()

```
QSize Digikam::TagRegion::adjustToOrientation (
    QRect & region,
    int orientation,
    const QSize & fullSize ) [static]
```

Rotate and flip region to [MetaEngine::ImageOrientation](#). The value region are calculated for the new image orientation.

### 6.1415.2.3 intersects()

```
bool Digikam::TagRegion::intersects (
    const TagRegion & other,
    double fraction = 0 )
```

Returns true if this and the other region intersect. *fraction* describes the relative overlap area needed to return true: If *fraction* is 0, returns true if the regions intersect at all. If *fraction* is 1, returns true only if other is completely contained in this region. If *fraction* is *x*,  $0 < x < 1$ , returns true if the area of this region covered by the other is greater than *x*. Invalid areas never intersect.

### 6.1415.2.4 mapToOriginalSize() [1/2]

```
QRect Digikam::TagRegion::mapToOriginalSize (
    const DImg & reducedSizeImage,
    const QRect & reducedSizeDetail ) [static]
```

Takes the original and reduced size from the [DImg](#).

### 6.1415.2.5 mapToOriginalSize() [2/2]

```
QRect Digikam::TagRegion::mapToOriginalSize (
    const QSize & fullImageSize,
    const QSize & reducedImageSize,
    const QRect & reducedSizeDetail ) [static]
```

Converts detail rectangles taken from a reduced size image to the original size, and vice versa.

### 6.1415.2.6 relativeToAbsolute() [1/2]

```
QRect Digikam::TagRegion::relativeToAbsolute (
    const QRectF & region,
    const DImg & reducedSizeImage ) [static]
```

Takes the original and reduced size from the [DImg](#), maps to original size.

### 6.1415.2.7 relativeToAbsolute() [2/2]

```
QRect Digikam::TagRegion::relativeToAbsolute (
    const QRectF & region,
    const QSize & fullSize ) [static]
```

Takes a relative region and a full size and returns the absolute region.

### 6.1415.2.8 reverseToOrientation()

```
void Digikam::TagRegion::reverseToOrientation (
    QRect & region,
    int orientation,
    const QSize & fullSize ) [static]
```

Reverse rotate and flip region to [MetaEngine::ImageOrientation](#). The value region are calculated for the new image orientation.

**6.1415.2.9 toRect()**

```
QRect Digikam::TagRegion::toRect ( ) const
```

If type is Rect, returns the contained rectangle.

**6.1415.2.10 toVariant()**

```
QVariant Digikam::TagRegion::toVariant ( ) const
```

Stores in / loads from a variant. Will only use native QVariant types.

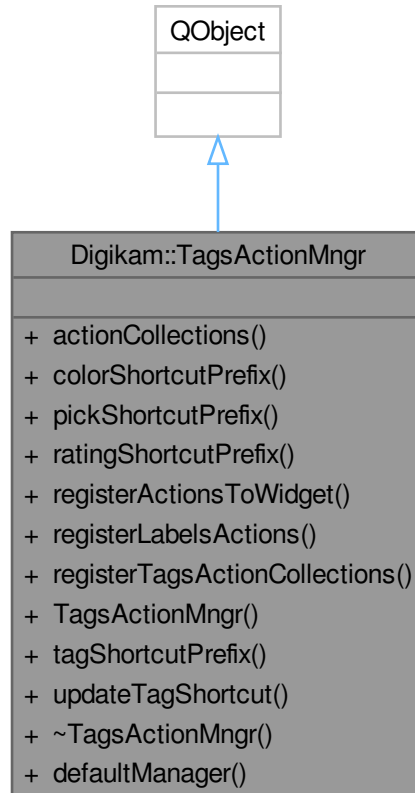
**6.1415.2.11 toXml()**

```
QString Digikam::TagRegion::toXml ( ) const
```

Returns an XML textual representation of this region.

**6.1416 Digikam::TagsActionMngr Class Reference**

Inheritance diagram for Digikam::TagsActionMngr:



## Signals

- void **signalShortcutPressed** (const QString &shortcut, int val)

## Public Member Functions

- QList< KActionCollection \* > [actionCollections](#) () const
- QString **colorShortcutPrefix** () const
- QString **pickShortcutPrefix** () const
- QString **ratingShortcutPrefix** () const
- void **registerActionsToWidget** (QWidget \*const wdg)
- void [registerLabelsActions](#) (KActionCollection \*const ac)
- void [registerTagsActionCollections](#) ()
- **TagsActionMngr** (QWidget \*const parent)
- QString **tagShortcutPrefix** () const
- void [updateTagShortcut](#) (int tagId, const QKeySequence &ks, bool delAction=true)

## Static Public Member Functions

- static [TagsActionMngr](#) \* **defaultManager** ()

## 6.1416.1 Member Function Documentation

### 6.1416.1.1 actionCollections()

```
QList< KActionCollection * > Digikam::TagsActionMngr::actionCollections ( ) const
```

Return the list of whole action collections managed.

### 6.1416.1.2 registerLabelsActions()

```
void Digikam::TagsActionMngr::registerLabelsActions (
    KActionCollection *const ac )
```

Register all labels actions to collections managed with keyboard shortcuts. Unlike tags actions, labels shortcuts are stored in XML GUI file of each root windows, to be able to customize it through KDE keyboards shortcuts config panel. This method must be called before to `DXmlGuiWindow::createGUI()`, typically when window actions are registered to ActionCollection instance.

### 6.1416.1.3 registerTagsActionCollections()

```
void Digikam::TagsActionMngr::registerTagsActionCollections ( )
```

Register all tag actions to collections managed with keyboard shortcuts. Because Tags shortcuts are stored in database this method must be called after database initialization and after that all root window instances have been created.

### 6.1416.1.4 updateTagShortcut()

```
void Digikam::TagsActionMngr::updateTagShortcut (
    int tagId,
    const QKeySequence & ks,
    bool delAction = true )
```

Updates the shortcut action for a tag. Call this when a shortcut was added, removed or changed.

## 6.1417 Digikam::TagsCache Class Reference

Inheritance diagram for Digikam::TagsCache:



### Public Types

- enum `HiddenTagsPolicy` { `NoHiddenTags` , `IncludeHiddenTags` }
- enum `LeadingSlashPolicy` { `NoLeadingSlash` , `IncludeLeadingSlash` }

## Signals

- void **tagAboutToBeDeleted** (QString name)
- void **tagAdded** (int tagId)
- void **tagDeleted** (int tagId)

## Public Member Functions

- bool **canBeWrittenToMetadata** (int tagId) const
- int **colorLabelForTag** (int tagId)
- int **colorLabelFromTags** (const QList< int > &tagIds)
- QVector< int > **colorLabelTags** ()
- bool **containsPublicTags** (const QList< int > &tagIds) const
- int **createTag** (const QString &tagPathToCreate)
- QList< int > **createTags** (const QStringList &tagPaths)
- int **getOrCreateInternalTag** (const QString &tagName)
- int **getOrCreateTag** (const QString &tagPath)
- QList< int > **getOrCreateTags** (const QStringList &tagPaths)
- int **getOrCreateTagWithProperty** (const QString &tagPath, const QString &property, const QString &value=QString())
- bool **hasProperty** (int tagId, const QString &property, const QString &value=QString()) const
- bool **hasTag** (int id) const
- bool **isInternalTag** (int tagId) const
- int **parentTag** (int id) const
- QList< int > **parentTags** (int id) const
- int **pickLabelForTag** (int tagId)
- int **pickLabelFromTags** (const QList< int > &tagIds)
- QVector< int > **pickLabelTags** ()
- QMap< QString, QString > **properties** (int tagId) const
- QString **propertyValue** (int tagId, const QString &property) const
- QStringList **propertyValues** (int tagId, const QString &property) const
- QList< int > **publicTags** (const QList< int > &tagIds) const
- QStringList **shortenedTagPaths** (const QList< int > &ids, [LeadingSlashPolicy](#) slashPolicy=[IncludeLeadingSlash](#), [HiddenTagsPolicy](#) hiddenTagsPolicy=[IncludeHiddenTags](#)) const
- QStringList **shortenedTagPaths** (const QList< int > &ids, QList< int > \*sortedIds, [LeadingSlashPolicy](#) slashPolicy=[IncludeLeadingSlash](#), [HiddenTagsPolicy](#) hiddenTagsPolicy=[IncludeHiddenTags](#)) const
- int **tagForColorLabel** (int label)
- int **tagForName** (const QString &tagName, int parentId=0) const
- int **tagForPath** (const QString &path) const
- int **tagForPickLabel** (int label)
- QString **tagName** (int id) const
- QStringList **tagNames** (const QList< int > &ids, [HiddenTagsPolicy](#) hiddenTagsPolicy=[IncludeHiddenTags](#)) const
- QString **tagPath** (int id, [LeadingSlashPolicy](#) slashPolicy=[IncludeLeadingSlash](#)) const
- QStringList **tagPaths** (const QList< int > &ids, [LeadingSlashPolicy](#) slashPolicy=[IncludeLeadingSlash](#), [HiddenTagsPolicy](#) hiddenTagsPolicy=[IncludeHiddenTags](#)) const
- QList< int > **tagsContaining** (const QString &fragment, Qt::CaseSensitivity caseSensitivity=Qt::Case↔Insensitive, [HiddenTagsPolicy](#) hiddenTagsPolicy=[NoHiddenTags](#))
- QList< int > **tagsForName** (const QString &tagName, [HiddenTagsPolicy](#) hiddenTagsPolicy=[NoHiddenTags](#)) const
- QList< int > **tagsForPaths** (const QStringList &tagPaths) const
- QList< int > **tagsStartingWith** (const QString &begin, Qt::CaseSensitivity caseSensitivity=Qt::Case↔Insensitive, [HiddenTagsPolicy](#) hiddenTagsPolicy=[NoHiddenTags](#))
- QList< int > **tagsWithProperty** (const QString &property, const QString &value=QString()) const
- QList< int > **tagsWithPropertyCached** (const QString &property) const

## Static Public Member Functions

- static [TagsCache](#) \* **instance** ()
- static QLatin1String **propertyNameDigikamInternalTag** ()
- static QLatin1String **propertyNameExcludedFromWriting** ()
- static QLatin1String **tagPathOfDigikamInternalTags** ([LeadingSlashPolicy](#) slashPolicy=[IncludeLeadingSlash](#))

## Friends

- class **ChangingDB**
- class **CoreDbAccess**
- class **TagsCacheCreator**

## 6.1417.1 Member Enumeration Documentation

### 6.1417.1.1 LeadingSlashPolicy

```
enum Digikam::TagsCache::LeadingSlashPolicy
```

#### Enumerator

NoLeadingSlash	Ex: "Places/Cities/Paris".
IncludeLeadingSlash	Ex: "/Places/Cities/Paris".

## 6.1417.2 Member Function Documentation

### 6.1417.2.1 canBeWrittenToMetadata()

```
bool Digikam::TagsCache::canBeWrittenToMetadata (
    int tagId ) const
```

Returns if a tag shall be written to the metadata of a file. Always returns false if the tag is a program-internal tag.

### 6.1417.2.2 colorLabelForTag()

```
int Digikam::TagsCache::colorLabelForTag (
    int tagId )
```

Return color label id corresponding of internal tags ID. see ColorLabel values from globals.h. Return -1 if not it's found.

### 6.1417.2.3 colorLabelFromTags()

```
int Digikam::TagsCache::colorLabelFromTags (
    const QList< int > & tagIds )
```

From the given list of tags, returns the color label corresponding to the first encountered tag which is a color label tag. Returns -1 if no tag in the list is a color label tag.



#### 6.1417.2.4 colorLabelTags()

```
QVector< int > Digikam::TagsCache::colorLabelTags ( )
```

Returns all color label tags, where index is the label id and value the tag id.

#### 6.1417.2.5 containsPublicTags()

```
bool Digikam::TagsCache::containsPublicTags (
    const QList< int > & tagIds ) const
```

Returns true if the given list of tag ids contains at least one non-internal tag.

#### 6.1417.2.6 createTag()

```
int Digikam::TagsCache::createTag (
    const QString & tagPathToCreate )
```

Add the tag described by the given tag path, and all missing parent tags, to the database. Returns the tag id. Use this if you know that tag path does not exist. If you are unsure, use `getOrCreateTag`.

#### 6.1417.2.7 getOrCreateInternalTag()

```
int Digikam::TagsCache::getOrCreateInternalTag (
    const QString & tagName )
```

For the given tag name (not path!), find the existing tag or creates a new internal tags under the usual tag path used for internal tags.

#### 6.1417.2.8 getOrCreateTag()

```
int Digikam::TagsCache::getOrCreateTag (
    const QString & tagPath )
```

A combination of `tagForPath` and `createTag`: Finds ids for the given tagPaths. If a tag does not exist yet and `create` is true, it will be created. Otherwise the id 0 is returned for this path.

#### 6.1417.2.9 getOrCreateTagWithProperty()

```
int Digikam::TagsCache::getOrCreateTagWithProperty (
    const QString & tagPath,
    const QString & property,
    const QString & value = QString() )
```

Calls `getOrCreateTag` for the given path, and ensures that the tag has assigned the given property. If you pass a null string as value, then the value is not checked and not changed.

### 6.1417.2.10 hasProperty()

```
bool Digikam::TagsCache::hasProperty (
    int tagId,
    const QString & property,
    const QString & value = QString() ) const
```

Tests if the tag has the given property: a) just has the property. b) has the property with the given value (value not null).

### 6.1417.2.11 hasTag()

```
bool Digikam::TagsCache::hasTag (
    int id ) const
```

Returns true if the tag for the given id exists.

### 6.1417.2.12 isInternalTag()

```
bool Digikam::TagsCache::isInternalTag (
    int tagId ) const
```

Returns if a tag is to be regarded program-internal, that is, a technical implementation detail not visible to the user at any time.

### 6.1417.2.13 parentTag()

```
int Digikam::TagsCache::parentTag (
    int id ) const
```

Returns the parent tag id, or 0 if a toplevel tag or tag does not exist.

### 6.1417.2.14 parentTags()

```
QList< int > Digikam::TagsCache::parentTags (
    int id ) const
```

Returns the parent tag ids of the given tag, starting with the toplevel tag, ending with the direct parent tag. If the tag is a toplevel tag or does not exist, an empty list is returned.

### 6.1417.2.15 pickLabelForTag()

```
int Digikam::TagsCache::pickLabelForTag (
    int tagId )
```

Return pick label id corresponding of internal tags ID. see PickLabel values from globals.h. Return -1 if not it's found.

### 6.1417.2.16 pickLabelFromTags()

```
int Digikam::TagsCache::pickLabelFromTags (
    const QList< int > & tagIds )
```

From the given list of tags, returns the pick label corresponding to the first encountered tag which is a pick label tag. Returns -1 if no tag in the list is a pick label tag.

### 6.1417.2.17 pickLabelTags()

```
QVector< int > Digikam::TagsCache::pickLabelTags ( )
```

Returns all pick label tags, where index is the label id and value the tag id.

### 6.1417.2.18 properties()

```
QMap< QString, QString > Digikam::TagsCache::properties (
    int tagId ) const
```

Returns a list or a map of the properties of the tag. Note: The list and map may be constructed for each call. Prefer [hasProperty\(\)](#) and [property\(\)](#).

### 6.1417.2.19 propertyValue()

```
QString Digikam::TagsCache::propertyValue (
    int tagId,
    const QString & property ) const
```

Returns the value of the property. Returning a null string cannot distinguish between the property set with a null value, or the property not set. The first method returns any property, if multiple are set with the same key.

### 6.1417.2.20 publicTags()

```
QList< int > Digikam::TagsCache::publicTags (
    const QList< int > & tagIds ) const
```

From the given list of tag ids, filter out any internal tags and return only public tags.

### 6.1417.2.21 shortenedTagPaths()

```
QStringList Digikam::TagsCache::shortenedTagPaths (
    const QList< int > & ids,
    LeadingSlashPolicy slashPolicy = IncludeLeadingSlash,
    HiddenTagsPolicy hiddenTagsPolicy = IncludeHiddenTags ) const
```

Utility method. Orders the given tag paths. If tags begin with the same path (parent tags), the relevant part is cut off in the second line. The second variant allows you to pass a list as return parameter. This list will contain, upon return, the tag id corresponding to each tag in the returned, sorted list of shortened paths.

### 6.1417.2.22 tagAdded

```
void Digikam::TagsCache::tagAdded (
    int tagId ) [signal]
```

These signals are provided for convenience; for finer grained information use [CoreDbWatch](#). Use a queued connection if you carry out longer operations from slots connected to these signals.

### 6.1417.2.23 tagForColorLabel()

```
int Digikam::TagsCache::tagForColorLabel (
    int label )
```

Return internal tags ID corresponding of color label id. see ColorLabel values from globals.h. Return 0 if not it's found.

### 6.1417.2.24 tagForName()

```
int Digikam::TagsCache::tagForName (
    const QString & tagName,
    int parentId = 0 ) const
```

Returns the id of the tag with the given name and parent tag. If parentId is 0, the tag is a toplevel tag. Returns 0 if there is no such tag.

### 6.1417.2.25 tagForPath()

```
int Digikam::TagsCache::tagForPath (
    const QString & path ) const
```

Returns the tag matched exactly by the given path. The path can be given with or without leading slash. Returns 0 if there is no such tag, or if path is empty. If you want to create the tag if it does not yet exist, use `getOrCreateTag`.

### 6.1417.2.26 tagForPickLabel()

```
int Digikam::TagsCache::tagForPickLabel (
    int label )
```

Return internal tags ID corresponding of pick label id. see PickLabel values from globals.h. Return 0 if not it's found.

### 6.1417.2.27 tagName()

```
QString Digikam::TagsCache::tagName (
    int id ) const
```

Returns the name of the tag with the given id. For the tag Places/Cities/Paris, this is Paris. If there is no tag for the given id a null string is returned.

### 6.1417.2.28 tagPath()

```
QString Digikam::TagsCache::tagPath (
    int id,
    LeadingSlashPolicy slashPolicy = IncludeLeadingSlash ) const
```

Returns the path of the tag with the given id. For the tag Places/Cities/Paris, this is Places/Cities/Paris. If there is no tag for the given id a null string is returned.

### 6.1417.2.29 tagsContaining()

```
QList< int > Digikam::TagsCache::tagsContaining (
    const QString & fragment,
    Qt::CaseSensitivity caseSensitivity = Qt::CaseInsensitive,
    HiddenTagsPolicy hiddenTagsPolicy = NoHiddenTags )
```

Returns a list of tag ids whose tag name (not path) starts with / contains the given fragment

### 6.1417.2.30 tagsForName()

```
QList< int > Digikam::TagsCache::tagsForName (
    const QString & tagName,
    HiddenTagsPolicy hiddenTagsPolicy = NoHiddenTags ) const
```

Finds all tags with the given name. For "Paris", this may give "Places/Cities/Paris" and "Places/USA/Texas/Paris". If there is no tag with the given name at all, returns an empty list.

### 6.1417.2.31 tagsWithProperty()

```
QList< int > Digikam::TagsCache::tagsWithProperty (
    const QString & property,
    const QString & value = QString() ) const
```

Finds all tags with the given property. The tag: a) just has the property. b) has the property with the given value (value not null). Note: The returned list is sorted.

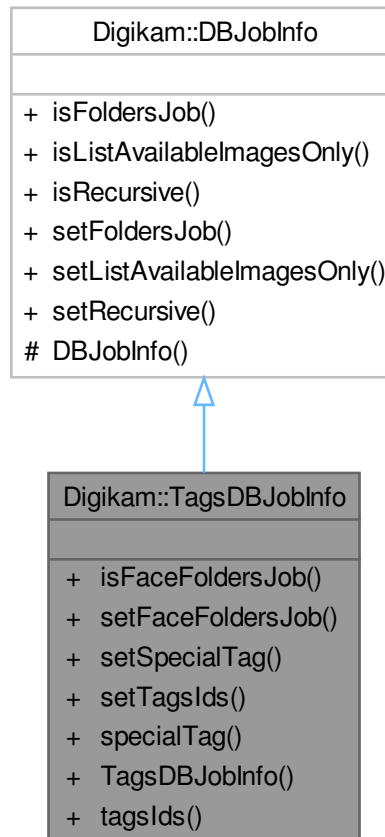
### 6.1417.2.32 tagsWithPropertyCached()

```
QList< int > Digikam::TagsCache::tagsWithPropertyCached (
    const QString & property ) const
```

This method is equivalent to calling tagsWithProperty(property), but the immediate result will be cached for subsequent calls. Use it for queries for which you know that they will be issued very often, so that it's worth caching the result of the already pretty fast tagsWithProperty().

## 6.1418 Digikam::TagsDBJobInfo Class Reference

Inheritance diagram for Digikam::TagsDBJobInfo:



### Public Member Functions

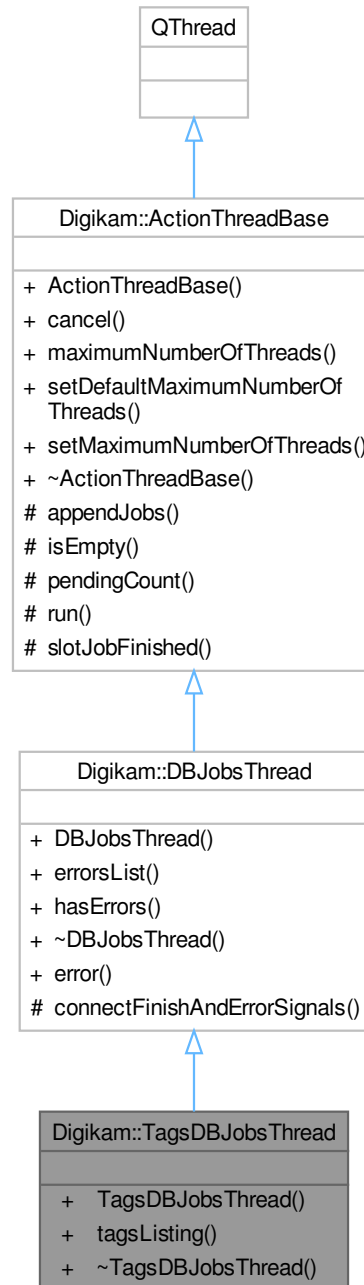
- bool **isFaceFoldersJob** () const
- void **setFaceFoldersJob** ()
- void **setSpecialTag** (const QString &tag)
- void **setTagsIds** (const QList< int > &tagsIds)
- QString **specialTag** () const
- QList< int > **tagsIds** () const

### Public Member Functions inherited from [Digikam::DBJobInfo](#)

- bool **isFoldersJob** () const
- bool **isListAvailableImagesOnly** () const
- bool **isRecursive** () const
- void **setFoldersJob** ()
- void **setListAvailableImagesOnly** ()
- void **setRecursive** ()

## 6.1419 Digikam::TagsDBJobsThread Class Reference

Inheritance diagram for Digikam::TagsDBJobsThread:



### Signals

- void **faceFoldersData** (const QMap< QString, QHash< int, int > > &)
- void **foldersData** (const QHash< int, int > &)

## Signals inherited from [Digikam::DBJobsThread](#)

- void **data** (const QList< [ItemLISTERRecord](#) > &records)
- void **finished** ()

## Public Member Functions

- **TagsDBJobsThread** (QObject \*const parent)
- void **tagsListing** (const [TagsDBJobInfo](#) &info)  
*Starts tags listing and scanning job(s)*

## Public Member Functions inherited from [Digikam::DBJobsThread](#)

- **DBJobsThread** (QObject \*const parent)
- QList< QString > & **errorsList** ()  
*A method to get all errors reported from jobs.*
- bool **hasErrors** ()  
*hasErrors: a method to check for jobs errors*

## Public Member Functions inherited from [Digikam::ActionThreadBase](#)

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

## Additional Inherited Members

## Public Slots inherited from [Digikam::DBJobsThread](#)

- void **error** (const QString &errString)  
*Appends the error string to m\_errorsList.*

## Protected Slots inherited from [Digikam::ActionThreadBase](#)

- void **slotJobFinished** ()

## Protected Member Functions inherited from [Digikam::DBJobsThread](#)

- void **connectFinishAndErrorSignals** (DBJob \*const j)  
*Connects the signals of job to the signals of the thread.*

## Protected Member Functions inherited from [Digikam::ActionThreadBase](#)

- void **appendJobs** (const [ActionJobCollection](#) &jobs)
- bool **isEmpty** () const
- int **pendingCount** () const
- void **run** () override

## 6.1419.1 Member Function Documentation

### 6.1419.1.1 tagsListing()

```
void Digikam::TagsDBJobsThread::tagsListing (
    const TagsDBJobInfo & info )
```

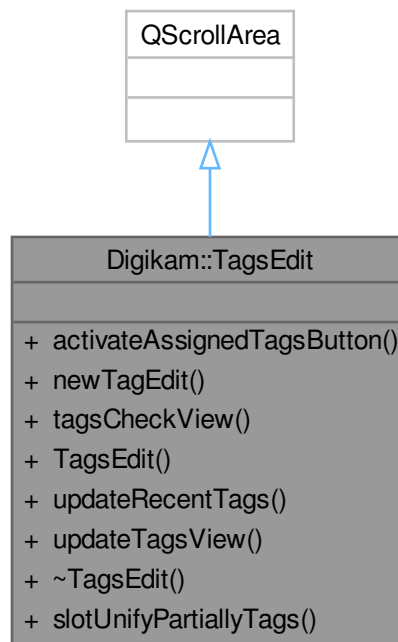


## Parameters

<i>info</i>	represents the tags job info
-------------	------------------------------

## 6.1420 Digikam::TagsEdit Class Reference

Inheritance diagram for Digikam::TagsEdit:



### Public Slots

- void **slotUnifyPartiallyTags** ()

### Signals

- void **signalChanged** ()
- void **signalImageTagsChanged** (qlonglong imageld)

### Public Member Functions

- void **activateAssignedTagsButton** ()
- [AddTagsLineEdit](#) \* **newTagEdit** () const
- [TagCheckView](#) \* **tagsCheckView** () const
- **TagsEdit** ([DisjointMetadata](#) \*const hub, QWidget \*const parent)
- void **updateRecentTags** ()
- void **updateTagsView** ()

## 6.1421 Digikam::TagShortInfo Class Reference

### Public Member Functions

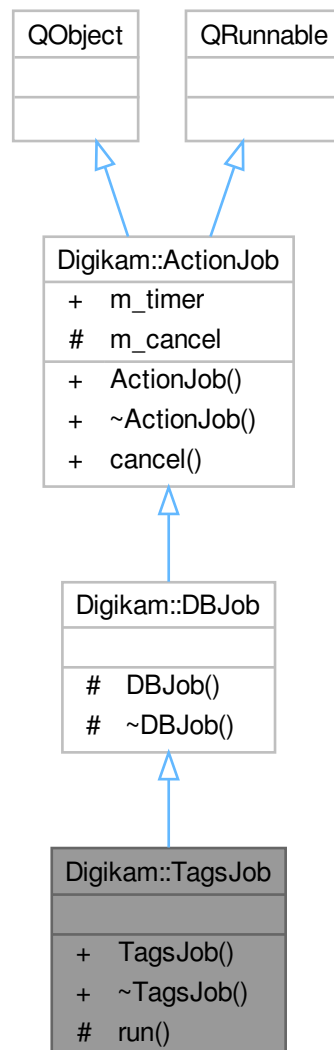
- bool **isNull** () const

### Public Attributes

- int **id** = 0
- QString **name**
- int **pid** = 0

## 6.1422 Digikam::TagsJob Class Reference

Inheritance diagram for Digikam::TagsJob:



## Signals

- void **faceFoldersData** (const QMap< QString, QHash< int, int > > &data)
- void **foldersData** (const QHash< int, int > &data)

## Signals inherited from [Digikam::DBJob](#)

- void **data** (const QList< [ItemListerRecord](#) > &records)
- void **error** (const QString &err)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Member Functions

- **TagsJob** (const [TagsDBJobInfo](#) &jobInfo)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void [run](#) () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.1423 Digikam::TagsLineEditOverlay Class Reference

Inheritance diagram for Digikam::TagsLineEditOverlay:



### Signals

- void **tagEdited** (const QModelIndex &index, const QString &)
- void **tagEdited** (const QModelIndex &index, int rating)

## Signals inherited from [Digikam::ItemDelegateOverlay](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)
- void **update** (const QModelIndex &index)

## Public Member Functions

- [AddTagsLineEdit](#) \* **addTagsLineEdit** () const
- [TagsLineEditOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- [ItemDelegateOverlay](#) (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

## Protected Slots

- void **slotDataChanged** (const QModelIndex &, const QModelIndex &)
- void **slotTagChanged** (const QString &)
- void **slotTagChanged** (int)

## Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual void **slotLayoutChanged** ()
- virtual void **slotReset** ()
- virtual void **slotRowsRemoved** (const QModelIndex &parent, int start, int end)
- virtual void **slotViewportEntered** ()

## Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

## Protected Member Functions

- QWidget \* **createWidget** () override
- void **hide** () override
- void **setActive** (bool) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **updateTag** ()
- void **visualChange** () override

## Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- virtual bool [checkIndex](#) (const QModelIndex &index) const
- bool [checkIndexOnEnter](#) (const QModelIndex &index) const
- bool [eventFilter](#) (QObject \*obj, QEvent \*event) override
- virtual QString [notifyMultipleMessage](#) (const QModelIndex &, int number)
- QWidget \* [parentWidget](#) () const
- virtual void [viewportLeaveEvent](#) (QObject \*obj, QEvent \*event)
- virtual void [widgetEnterEvent](#) ()
- void [widgetEnterNotifyMultiple](#) (const QModelIndex &index)
- virtual void [widgetLeaveEvent](#) ()
- void [widgetLeaveNotifyMultiple](#) ()

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > [affectedIndexes](#) (const QModelIndex &index) const
- bool [affectsMultiple](#) (const QModelIndex &index) const
- int [numberOfAffectedIndexes](#) (const QModelIndex &index) const
- bool [viewHasMultiSelection](#) () const

## Protected Attributes

- QPersistentModelIndex [m\\_index](#)

## Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool [m\\_mouseButtonPressedOnWidget](#) = false
- QWidget \* [m\\_widget](#) = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- QAbstractItemDelegate \* [m\\_delegate](#) = nullptr
- QAbstractItemView \* [m\\_view](#) = nullptr

## 6.1423.1 Member Function Documentation

### 6.1423.1.1 createWidget()

```
QWidget * Digikam::TagsLineEditOverlay::createWidget ( ) [override], [protected], [virtual]
```

Create your widget here. When creating the object, pass [parentWidget\(\)](#) as parent widget. Ownership of the object is passed. It will be deleted in [setActive\(false\)](#).

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1423.1.2 hide()

```
void Digikam::TagsLineEditOverlay::hide ( ) [override], [protected], [virtual]
```

Called when the widget shall be hidden (mouse cursor left index, viewport, uninstalled etc.). Default implementation [hide\(\)](#)s `m_widget`.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1423.1.3 setActive()

```
void Digikam::TagsLineEditOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1423.1.4 slotEntered()

```
void Digikam::TagsLineEditOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and `checkIndex` returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1423.1.5 visualChange()

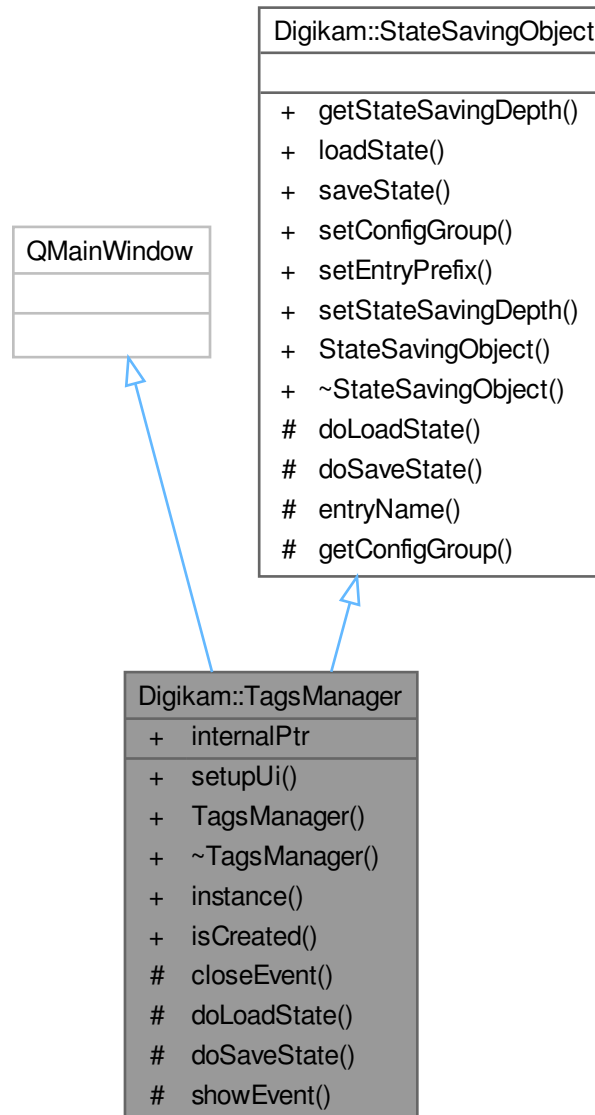
```
void Digikam::TagsLineEditOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

## 6.1424 Digikam::TagsManager Class Reference

Inheritance diagram for Digikam::TagsManager:



### Signals

- void **signalSelectionChanged** ([TAlbum](#) \*album)

### Public Member Functions

- void **setupUi** ()  
*setupUi setup all gui elements for Tag Manager*



## Public Member Functions inherited from Digikam::StateSavingObject

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Static Public Member Functions

- static [TagsManager](#) \* [instance](#) ()
- static bool [isCreated](#) ()

## Static Public Attributes

- static QPointer< [TagsManager](#) > [internalPtr](#) = QPointer<[TagsManager](#)>()

## Protected Member Functions

- void [closeEvent](#) (QCloseEvent \*event) override
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- void [showEvent](#) (QShowEvent \*event) override

## Protected Member Functions inherited from Digikam::StateSavingObject

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## Additional Inherited Members

## Public Types inherited from Digikam::StateSavingObject

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### 6.1424.1 Member Function Documentation

#### 6.1424.1.1 doLoadState()

```
void Digikam::TagsManager::doLoadState ( ) [override], [protected], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

**6.1424.1.2 doSaveState()**

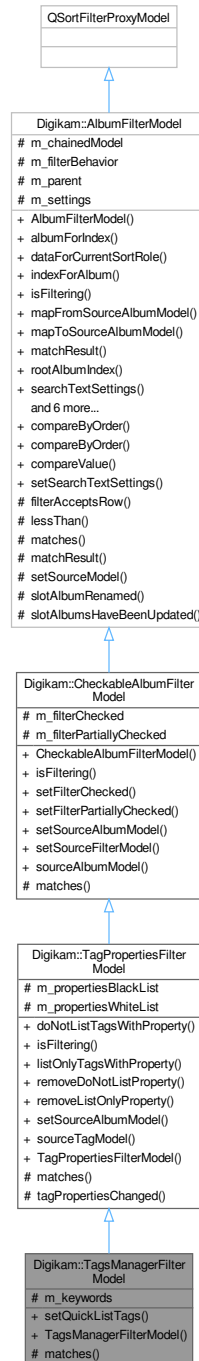
```
void Digikam::TagsManager::doSaveState ( ) [override], [protected], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

## 6.1425 Digikam::TagsManagerFilterModel Class Reference

Inheritance diagram for Digikam::TagsManagerFilterModel:



### Public Member Functions

- void **setQuickListTags** (const QList< int > &tags)
- **TagsManagerFilterModel** (QObject \*const data=nullptr)

## Public Member Functions inherited from [Digikam::TagPropertiesFilterModel](#)

- void **doNotListTagsWithProperty** (const QString &property)
- bool **isFiltering** () const override
- void **listOnlyTagsWithProperty** (const QString &property)
- void **removeDoNotListProperty** (const QString &property)
- void **removeListOnlyProperty** (const QString &property)
- void **setSourceAlbumModel** ([TagModel](#) \*const source)
- [TagModel](#) \* **sourceTagModel** () const
- [TagPropertiesFilterModel](#) (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::CheckableAlbumFilterModel](#)

- [CheckableAlbumFilterModel](#) (QObject \*const parent=nullptr)
- void **setFilterChecked** (bool filter)
- void **setFilterPartiallyChecked** (bool filter)
- void **setSourceAlbumModel** ([AbstractCheckableAlbumModel](#) \*const source)
- void **setSourceFilterModel** ([CheckableAlbumFilterModel](#) \*const source)
- [AbstractCheckableAlbumModel](#) \* **sourceAlbumModel** () const

## Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- [AlbumFilterModel](#) (QObject \*const parent=nullptr)
  - [Album](#) \* **albumForIndex** (const QModelIndex &index) const
- Convenience methods.*
- QVariant **dataForCurrentSortRole** ([Album](#) \*album) const
  - QModelIndex **indexForAlbum** ([Album](#) \*album) const
  - QModelIndex **mapFromSourceAlbumModel** (const QModelIndex &index) const
  - QModelIndex **mapToSourceAlbumModel** (const QModelIndex &index) const
  - [MatchResult](#) **matchResult** (const QModelIndex &index) const
  - QModelIndex **rootAlbumIndex** () const
  - [SearchTextSettings](#) **searchTextSettings** () const
  - void **setFilterBehavior** ([FilterBehavior](#) behavior)
  - void **setSourceAlbumModel** ([AbstractAlbumModel](#) \*const source)
  - void **setSourceFilterModel** ([AlbumFilterModel](#) \*const source)
  - [AbstractAlbumModel](#) \* **sourceAlbumModel** () const
  - [AlbumFilterModel](#) \* **sourceFilterModel** () const
  - void **updateFilter** ()

## Protected Member Functions

- bool **matches** ([Album](#) \*album) const override

## Protected Member Functions inherited from [Digikam::AlbumFilterModel](#)

- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- [MatchResult](#) **matchResult** ([Album](#) \*album) const
- void **setSourceModel** (QAbstractItemModel \*const model) override

### Protected Attributes

- QSet< int > **m\_keywords**

### Protected Attributes inherited from [Digikam::TagPropertiesFilterModel](#)

- QSet< QString > **m\_propertiesBlackList**
- QSet< QString > **m\_propertiesWhiteList**

### Protected Attributes inherited from [Digikam::CheckableAlbumFilterModel](#)

- bool **m\_filterChecked** = false
- bool **m\_filterPartiallyChecked** = false

### Protected Attributes inherited from [Digikam::AlbumFilterModel](#)

- QPointer< [AlbumFilterModel](#) > **m\_chainedModel** = nullptr
- [FilterBehavior](#) **m\_filterBehavior** = [FullFiltering](#)
- QObject \* **m\_parent** = nullptr
- [SearchTextSettings](#) **m\_settings**

### Additional Inherited Members

### Public Types inherited from [Digikam::AlbumFilterModel](#)

- enum [FilterBehavior](#) { [SimpleFiltering](#) , [FullFiltering](#) , [StrictFiltering](#) }
- enum [MatchResult](#) {  
    [NoMatch](#) = 0 , [DirectMatch](#) , [ParentMatch](#) , [ChildMatch](#) ,  
    [SpecialMatch](#) }

### Public Slots inherited from [Digikam::AlbumFilterModel](#)

- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)

### Signals inherited from [Digikam::AlbumFilterModel](#)

- void [hasSearchResult](#) (bool hasResult)
- void [searchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [searchTextSettingsChanged](#) (bool wasSearching, bool searched)
- void [signalFilterChanged](#) ()

### Static Public Member Functions inherited from [Digikam::AlbumFilterModel](#)

- template<typename T >  
    static int [compareByOrder](#) (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int [compareByOrder](#) (int compareResult, Qt::SortOrder sortOrder)
- template<typename T >  
    static int [compareValue](#) (const T &a, const T &b)

## Protected Slots inherited from [Digikam::TagPropertiesFilterModel](#)

- void `tagPropertiesChanged` ([TAlbum](#) \*)

## Protected Slots inherited from [Digikam::AlbumFilterModel](#)

- void `slotAlbumRenamed` ([Album](#) \*album)
- void `slotAlbumsHaveBeenUpdated` (int type)

### 6.1425.1 Member Function Documentation

#### 6.1425.1.1 matches()

```
bool Digikam::TagsManagerFilterModel::matches (
    Album * album ) const [override], [protected], [virtual]
```

This method provides the basic match checking algorithm. Return true if this single album matches the current criteria. This method can be overridden to provide custom filtering.

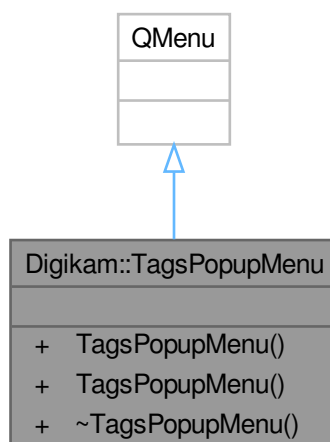
#### Parameters

<i>album</i>	the album to tell if it matches the filter criteria or not.
--------------	---

Reimplemented from [Digikam::TagPropertiesFilterModel](#).

### 6.1426 Digikam::TagsPopupMenu Class Reference

Inheritance diagram for `Digikam::TagsPopupMenu`:



## Public Types

- enum `Mode` { `ASSIGN = 0` , `REMOVE` , `DISPLAY` , `RECENTLYASSIGNED` }

## Signals

- void `signalPopupMenuView` ()
- void `signalTagActivated` (int id)

## Public Member Functions

- `TagsPopupMenu` (const QList< qlonglong > &selectedImageIDs, `Mode` mode, QWidget \*const parent=nullptr)
- `TagsPopupMenu` (qlonglong selectedImageId, `Mode` mode, QWidget \*const parent=nullptr)

## 6.1426.1 Member Enumeration Documentation

### 6.1426.1.1 Mode

enum `Digikam::TagsPopupMenu::Mode`

#### Enumerator

DISPLAY	Used by "GoTo Tag" feature.
---------	-----------------------------





## Public Slots inherited from Digikam::AbstractAlbumTreeView

- void [adaptColumnsToContent](#) ()
- void [expandEverything](#) (const QModelIndex &index)
- void [scrollToSelectedAlbum](#) ()
- void [setCurrentAlbums](#) (const QList< Album \* > &albums, bool selectInAlbumManager=true)
- void [setSearchTextSettings](#) (const [SearchTextSettings](#) &settings)
- void [slotCollapseAllNodes](#) ()
  - slotCollapseAllNodes - collapse all nodes without root node*
- void [slotCollapseNode](#) ()
  - slotCollapseNode - collapse recursively selected nodes*
- void [slotExpandNode](#) ()
  - slotExpandNode - expands recursively selected nodes*

## Signals

- void [assignTags](#) (int tagId, const QList< int > &imageIds)

## Signals inherited from Digikam::AbstractAlbumTreeView

- void [currentAlbumChanged](#) (Album \*currentAlbum)
- void [selectedAlbumsChanged](#) (const QList< Album \* > &selectedAlbums)

## Public Member Functions

- [TAlbum](#) \* [albumForIndex](#) (const QModelIndex &index) const
- [TagModel](#) \* [albumModel](#) () const
- [TAlbum](#) \* [currentAlbum](#) () const
  - currentAlbum Even if multiple selection is enabled current Album can be only one, the last clicked item if you need selected items, see selectedAlbums() It's NOT the same as AlbumManager::currentAlbums()*
- [TagPropertiesFilterModel](#) \* [filteredModel](#) () const
- QList< [TAlbum](#) \* > [selectedTagAlbums](#) ()
- QList< [Album](#) \* > [selectedTags](#) ()
  - selectedTags - return a list of all selected items in tag model*
- void [setAlbumFilterModel](#) ([TagPropertiesFilterModel](#) \*const [filteredModel](#), [CheckableAlbumFilterModel](#) \*const [filterModel](#))
- void [setAlbumModel](#) ([TagModel](#) \*const [model](#))
- [TagModificationHelper](#) \* [tagModificationHelper](#) () const
- [TagTreeView](#) (QWidget \*const [parent](#)=nullptr, [Flags](#) [flags](#)=DefaultFlags)

## Public Member Functions inherited from Digikam::AbstractCheckableAlbumTreeView

- [AbstractCheckableAlbumTreeView](#) (QWidget \*const [parent](#), [Flags](#) [flags](#))
- [CheckableAlbumFilterModel](#) \* [albumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [albumModel](#) () const
- [CheckableAlbumFilterModel](#) \* [checkableAlbumFilterModel](#) () const
- [AbstractCheckableAlbumModel](#) \* [checkableModel](#) () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- bool [isRestoreCheckState](#) () const
- void [setCheckOnMiddleClick](#) (bool [doThat](#))
- void [setRestoreCheckState](#) (bool [restore](#))

## Public Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- [AbstractCountingAlbumTreeView](#) (QWidget \*const parent, Flags flags)

## Public Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- [AbstractAlbumTreeView](#) (QWidget \*const parent, Flags flags)
- void **addContextMenuElement** ([ContextMenuElement](#) \*const element)
- [AlbumFilterModel](#) \* **albumFilterModel** () const
- [AbstractSpecificAlbumModel](#) \* **albumModel** () const
- QList< [ContextMenuElement](#) \* > **contextMenuElements** () const
- template<class A >  
QList< A \* > **currentAlbums** ()
- bool **expandMatches** (const QModelIndex &index)
- QModelIndex **indexVisuallyAt** (const QPoint &p)
- void **removeContextMenuElement** ([ContextMenuElement](#) \*const element)
- QList< [Album](#) \* > **selectedItems** ()  
*selectedItems()* -
- void **setAlbumManagerCurrentAlbum** (const bool setCurrentAlbum)
- void **setContextMenuIcon** (const QPixmap &pixmap)
- void **setContextMenuTitle** (const QString &title)
- void **setEnabledContextMenu** (const bool enable)
- void **setExpandNewCurrentItem** (const bool doThat)
- void **setExpandOnSingleClick** (const bool doThat)
- void **setSelectAlbumOnClick** (const bool selectOnClick)
- void **setSelectOnContextMenu** (const bool select)
- bool **viewportEvent** (QEvent \*event) override

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual **~StateSavingObject** ()

## Protected Attributes

- [TagPropertiesFilterModel](#) \* **m\_filteredModel** = nullptr
- [TagModificationHelper](#) \* **m\_modificationHelper** = nullptr

## Protected Attributes inherited from [Digikam::AbstractAlbumTreeView](#)

- [AlbumFilterModel](#) \* **m\_albumFilterModel** = nullptr
- [AbstractSpecificAlbumModel](#) \* **m\_albumModel** = nullptr
- bool **m\_checkOnMiddleClick** = false
- [AlbumModelDragDropHandler](#) \* **m\_dragDropHandler** = nullptr
- Flags **m\_flags** = DefaultFlags
- int **m\_lastScrollBarValue** = 0
- bool **m\_restoreCheckState** = false

## Additional Inherited Members

### Public Types inherited from [Digikam::AbstractAlbumTreeView](#)

- enum [Flag](#) { [CreateDefaultModel](#) , [CreateDefaultFilterModel](#) , [CreateDefaultDelegate](#) , [ShowCountAccordingToSettings](#) , [AlwaysShowInclusiveCounts](#) , **DefaultFlags** = [CreateDefaultFilterModel](#) | [CreateDefaultDelegate](#) | [ShowCountAccordingToSettings](#) }

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Protected Slots inherited from [Digikam::AbstractAlbumTreeView](#)

- void [albumSettingsChanged](#) ()
- void [slotCurrentChanged](#) ()
- virtual void [slotRootAlbumAvailable](#) ()
- void [slotSearchTextSettingsAboutToChange](#) (bool searched, bool willSearch)
- void [slotSearchTextSettingsChanged](#) (bool wasSearching, bool searching)
- void [slotSelectionChanged](#) ()

### Protected Member Functions inherited from [Digikam::AbstractCheckableAlbumTreeView](#)

- void [middleButtonPressed](#) ([Album](#) \*a) override
- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([CheckableAlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCheckableAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractCountingAlbumTreeView](#)

- void [rowsInserted](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractCountingAlbumModel](#) \*const model)

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeView](#)

- virtual void [addCustomContextMenuActions](#) ([ContextMenuHelper](#) &cmh, [Album](#) \*album)
- virtual [QPixmap](#) [contextMenuIcon](#) () const
- virtual [QString](#) [contextMenuTitle](#) () const
- void [dragEnterEvent](#) ([QDragEnterEvent](#) \*e) override
- void [dragLeaveEvent](#) ([QDragLeaveEvent](#) \*e) override
- void [dragMoveEvent](#) ([QDragMoveEvent](#) \*e) override
- void [dropEvent](#) ([QDropEvent](#) \*e) override
- virtual void [handleCustomContextMenuAction](#) ([QAction](#) \*action, const [AlbumPointer](#)< [Album](#) > &album)
- void [mousePressEvent](#) ([QMouseEvent](#) \*e) override

*Other helper methods.*

- virtual [QPixmap](#) [pixmapForDrag](#) (const [QStyleOptionViewItem](#) &option, [QList](#)< [QModelIndex](#) > indexes)
- void [rowsAboutToBeRemoved](#) (const [QModelIndex](#) &parent, int start, int end) override
- void [rowsInserted](#) (const [QModelIndex](#) &index, int start, int end) override
- void [setAlbumFilterModel](#) ([AlbumFilterModel](#) \*const filterModel)
- void [setAlbumModel](#) ([AbstractSpecificAlbumModel](#) \*const model)
- virtual bool [showContextMenuAt](#) ([QContextMenuEvent](#) \*event, [Album](#) \*albumForEvent)
- void [startDrag](#) ([Qt::DropActions](#) supportedActions) override

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

### 6.1427.1 Member Function Documentation

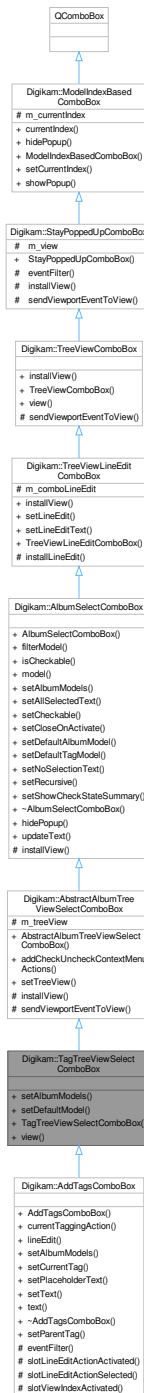
#### 6.1427.1.1 [filteredModel\(\)](#)

```
TagPropertiesFilterModel * Digikam::TagTreeView::filteredModel ( ) const
```

Contains only the tags filtered by properties - prefer to albumModel()

## 6.1428 Digikam::TagTreeViewSelectComboBox Class Reference

Inheritance diagram for Digikam::TagTreeViewSelectComboBox:



### Public Member Functions

- void **setAlbumModels** ([TagModel](#) \*model, [TagPropertiesFilterModel](#) \*filteredModel=nullptr, [CheckableAlbumFilterModel](#) \*filterModel=nullptr)

- void **setDefaultModel** ()
- **TagTreeViewSelectComboBox** (QWidget \*const parent=nullptr)
- **TagTreeView** \* **view** () const

### Public Member Functions inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- [AbstractAlbumTreeViewSelectComboBox](#) (QWidget \*const parent=nullptr)
- void [addCheckUncheckContextMenuActions](#) ()
- void [setTreeView](#) ([AbstractAlbumTreeView](#) \*const treeView)

### Public Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- **AlbumSelectComboBox** (QWidget \*const parent=nullptr)
- QSortFilterProxyModel \* [filterModel](#) () const
- bool **isCheckable** () const
- [AbstractCheckableAlbumModel](#) \* [model](#) () const
- void **setAlbumModels** ([AbstractCheckableAlbumModel](#) \*model, [AlbumFilterModel](#) \*filterModel=nullptr)
- void [setAllSelectedText](#) (bool all)
- void [setCheckable](#) (bool checkable)
- void [setCloseOnActivate](#) (bool close)
- void [setDefaultAlbumModel](#) ()
- void **setDefaultTagModel** ()
- void [setNoSelectionText](#) (const QString &text)
- void [setRecursive](#) (bool recursive)
- void [setShowCheckStateSummary](#) (bool show)

### Public Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- void [installView](#) (QAbstractItemView \*view=nullptr) override
- void **setLineEdit** (QLineEdit \*edit)
- void [setLineEditText](#) (const QString &text)
- [TreeViewLineEditComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::TreeViewComboBox](#)

- [TreeViewComboBox](#) (QWidget \*parent=nullptr)
- QTreeView \* [view](#) () const

### Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex **currentIndex** () const
- void **hidePopup** () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

## Additional Inherited Members

### Public Slots inherited from [Digikam::AlbumSelectComboBox](#)

- void `hidePopup` () override
- virtual void `updateText` ()

### Protected Member Functions inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- void `installView` (QAbstractItemView \*`view`=nullptr) override
- void `sendViewportEventToView` (QEvent \*`e`) override

### Protected Member Functions inherited from [Digikam::AlbumSelectComboBox](#)

- void `installView` (QAbstractItemView \*`view`=nullptr) override

### Protected Member Functions inherited from [Digikam::TreeViewLineEditComboBox](#)

- virtual void `installLineEdit` ()

### Protected Member Functions inherited from [Digikam::TreeViewComboBox](#)

- void `sendViewportEventToView` (QEvent \*`e`) override

### Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool `eventFilter` (QObject \*`watched`, QEvent \*`event`) override
- void `installView` (QAbstractItemView \*`view`)

### Protected Attributes inherited from [Digikam::AbstractAlbumTreeViewSelectComboBox](#)

- [AbstractAlbumTreeView](#) \* `m_treeView` = nullptr

### Protected Attributes inherited from [Digikam::TreeViewLineEditComboBox](#)

- QLineEdit \* `m_comboLineEdit` = nullptr

### Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

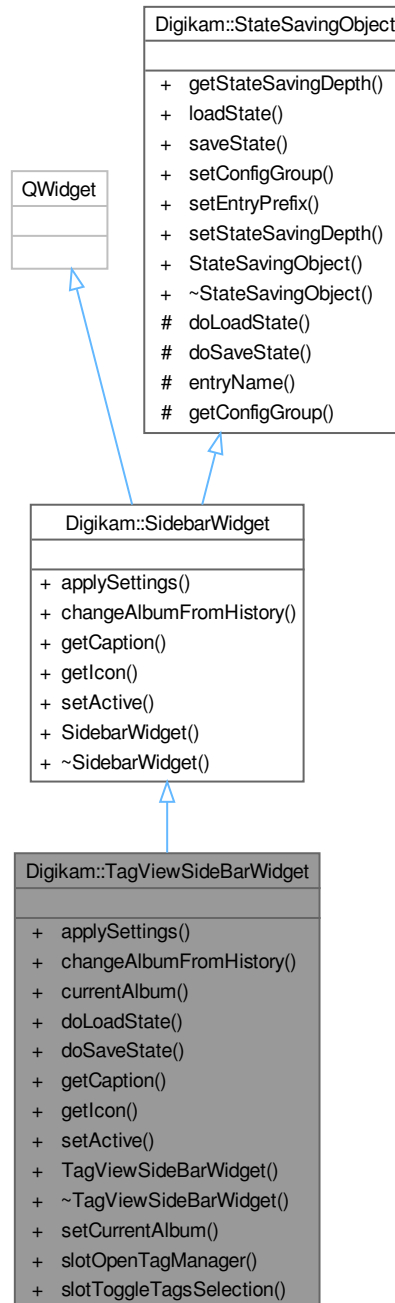
- QAbstractItemView \* `m_view` = nullptr

### Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex `m_currentIndex`

## 6.1429 Digikam::TagViewSideBarWidget Class Reference

Inheritance diagram for Digikam::TagViewSideBarWidget:



### Public Slots

- void `setCurrentAlbum` (`TAlbum *album`)
- void `slotOpenTagManager` ()
- void `slotToggleTagsSelection` (int `radioClicked`)



## Signals

- void **signalFindDuplicates** (const QList< [TAlbum](#) \* > &albums)

## Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

## Public Member Functions

- void [applySettings](#) () override
- void [changeAlbumFromHistory](#) (const QList< [Album](#) \* > &album) override
- [AlbumPointer](#)< [TAlbum](#) > **currentAlbum** () const
- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- void [setActive](#) (bool active) override
- [TagViewSideBarWidget](#) (QWidget \*const parent, [TagModel](#) \*const model)

## Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.1429.1 Member Function Documentation

### 6.1429.1.1 applySettings()

```
void Digikam::TagViewSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).

### 6.1429.1.2 changeAlbumFromHistory()

```
void Digikam::TagViewSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.1429.1.3 doLoadState()

```
void Digikam::TagViewSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1429.1.4 doSaveState()

```
void Digikam::TagViewSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1429.1.5 getCaption()

```
const QString Digikam::TagViewSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.1429.1.6 `getIcon()`

```
const QIcon Digikam::TagViewSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.1429.1.7 `setActive()`

```
void Digikam::TagViewSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

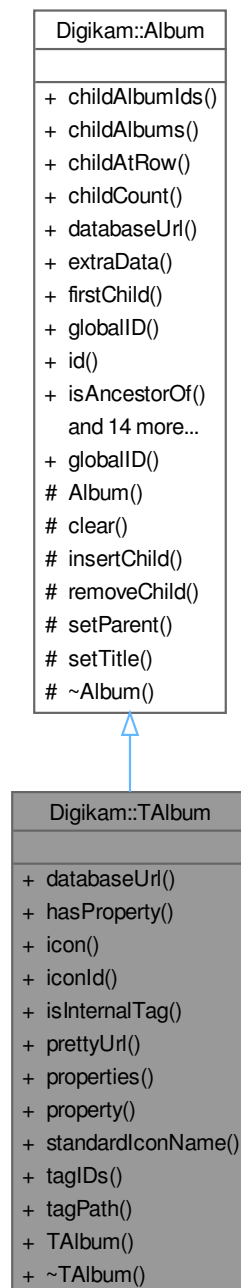
## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.1430 Digikam::TAlbum Class Reference

Inheritance diagram for Digikam::TAlbum:



## Public Member Functions

- [CoreDbUrl databaseUrl](#) () const override
- bool **hasProperty** (const QString &key) const
- QString **icon** () const
- qlonglong **iconId** () const
- bool **isInternalTag** () const
- QString **prettyUrl** () const
- QMap< QString, QString > **properties** () const
- QString **property** (const QString &key) const
- QString **standardIconName** () const
- QList< int > **tagIDs** () const
- QString **tagPath** (bool leadingSlash=true) const
- **Album** (const QString &title, int id, bool root=false)

## Public Member Functions inherited from [Digikam::Album](#)

- QList< int > [childAlbumIds](#) (bool recursive=false)
- AlbumList [childAlbums](#) (bool recursive=false)
- [Album](#) \* [childAtRow](#) (int row) const
- int [childCount](#) () const
- void \* [extraData](#) (const void \*const key) const
- [Album](#) \* [firstChild](#) () const
- int [globalID](#) () const
- int [id](#) () const
- bool [isAncestorOf](#) ([Album](#) \*const album) const
- bool [isRoot](#) () const
- bool [isTrashAlbum](#) () const
- bool [isUsedByLabelsTree](#) () const
- [Album](#) \* [lastChild](#) () const
- [Album](#) \* [next](#) () const
- [Album](#) \* [parent](#) () const
- void [prepareForDeletion](#) ()
- [Album](#) \* [prev](#) () const
- void [removeExtraData](#) (const void \*const key)
- int [rowFromAlbum](#) () const
- void [setExtraData](#) (const void \*const key, void \*const value)
- void [setUsedByLabelsTree](#) (bool isUsed)
- QString [title](#) () const
- [Type](#) [type](#) () const

## Friends

- class [AlbumManager](#)

## Additional Inherited Members

## Public Types inherited from [Digikam::Album](#)

- enum [Type](#) {  
[PHYSICAL](#) = 0 , [TAG](#) , [DATE](#) , [SEARCH](#) ,  
[FACE](#) }

## Static Public Member Functions inherited from [Digikam::Album](#)

- static int [globalID](#) ([Type](#) type, int id)  
*Produces the global id.*

## Protected Member Functions inherited from [Digikam::Album](#)

- [Album](#) ([Album::Type](#) type, int id, bool root)
- void [clear](#) ()
- void [insertChild](#) ([Album](#) \*const child)
- void [removeChild](#) ([Album](#) \*const child)
- void [setParent](#) ([Album](#) \*const parent)
- void [setTitle](#) (const QString &title)
- virtual [~Album](#) ()

### 6.1430.1 Detailed Description

A Tag [Album](#) representation

### 6.1430.2 Member Function Documentation

#### 6.1430.2.1 [databaseUrl\(\)](#)

```
CoreDbUrl Digikam::TAlbum::databaseUrl ( ) const [override], [virtual]
```

##### Returns

the kde url of the album

Implements [Digikam::Album](#).

#### 6.1430.2.2 [tagPath\(\)](#)

```
QString Digikam::TAlbum::tagPath (
    bool leadingSlash = true ) const
```

##### Returns

The tag path, e.g. "/People/Friend/John" if leadingSlash is true, "People/Friend/John" if leadingSlash if false.  
The root [TAlbum](#) returns "/" resp. "".

## 6.1431 Digikam::Template Class Reference

### Public Member Functions

- QStringList **authors** () const
- QString **authorsPosition** () const
- [IptcCoreContactInfo](#) **contactInfo** () const
- [MetaEngine::AltLangMap](#) **copyright** () const
- QString **credit** () const
- QString **instructions** () const
- QStringList **IptcSubjects** () const
- bool **isEmpty** () const
- bool **isNull** () const
- [IptcCoreLocationInfo](#) **locationInfo** () const
- void **merge** (const [Template](#) &t)
- bool **operator==** (const [Template](#) &t) const
- [MetaEngine::AltLangMap](#) **rightUsageTerms** () const
- void **setAuthors** (const QStringList &authors)
- void **setAuthorsPosition** (const QString &authorPosition)
- void **setContactInfo** (const [IptcCoreContactInfo](#) &inf)
- void **setCopyright** (const [MetaEngine::AltLangMap](#) &copyright)
- void **setCredit** (const QString &credit)
- void **setInstructions** (const QString &instructions)
- void **setIptcSubjects** (const QStringList &subjects)
- void **setLocationInfo** (const [IptcCoreLocationInfo](#) &inf)
- void **setRightUsageTerms** (const [MetaEngine::AltLangMap](#) &rightUsageTerms)
- void **setSource** (const QString &source)
- void **setTemplateTitle** (const QString &title)
- QString **source** () const
- QString **templateTitle** () const

### Static Public Member Functions

- static QString **removeTemplateTitle** ()

### Protected Attributes

- QStringList [m\\_authors](#)
- QString [m\\_authorsPosition](#)
- [IptcCoreContactInfo](#) [m\\_contactInfo](#)
- [MetaEngine::AltLangMap](#) [m\\_copyright](#)
- QString [m\\_credit](#)
- QString [m\\_instructions](#)
- [IptcCoreLocationInfo](#) [m\\_locationInfo](#)
- [MetaEngine::AltLangMap](#) [m\\_rightUsageTerms](#)
- QString [m\\_source](#)
- QStringList [m\\_subjects](#)
- QString [m\\_templateTitle](#)

## 6.1431.1 Member Function Documentation

### 6.1431.1.1 isEmpty()

```
bool Digikam::Template::isEmpty ( ) const
```

Return true if [Template](#) contents is empty

### 6.1431.1.2 isNull()

```
bool Digikam::Template::isNull ( ) const
```

Return true if [Template](#) title is null

### 6.1431.1.3 merge()

```
void Digikam::Template::merge (
    const Template & t )
```

Merge the metadata from another [Template](#)

### 6.1431.1.4 operator==( )

```
bool Digikam::Template::operator==(
    const Template & t ) const
```

Compare for metadata equality, not including "templateTitle" value.

## 6.1431.2 Member Data Documentation

### 6.1431.2.1 m\_authors

```
QStringList Digikam::Template::m_authors [protected]
```

List of author names.

### 6.1431.2.2 m\_authorsPosition

```
QString Digikam::Template::m_authorsPosition [protected]
```

Description of authors position.

### 6.1431.2.3 m\_contactInfo

```
IptcCoreContactInfo Digikam::Template::m_contactInfo [protected]
```

IPTC Contact Information.



#### 6.1431.2.4 m\_copyright

`MetaEngine::AltLangMap` Digikam::Template::m\_copyright [protected]

Language alternative copyright notices.

#### 6.1431.2.5 m\_credit

`QString` Digikam::Template::m\_credit [protected]

Credit description.

#### 6.1431.2.6 m\_instructions

`QString` Digikam::Template::m\_instructions [protected]

Special instructions to process with contents.

#### 6.1431.2.7 m\_locationInfo

`IptcCoreLocationInfo` Digikam::Template::m\_locationInfo [protected]

IPTC Location Information.

#### 6.1431.2.8 m\_rightUsageTerms

`MetaEngine::AltLangMap` Digikam::Template::m\_rightUsageTerms [protected]

Language alternative right term usages.

#### 6.1431.2.9 m\_source

`QString` Digikam::Template::m\_source [protected]

Descriptions of contents source.

#### 6.1431.2.10 m\_subjects

`QStringList` Digikam::Template::m\_subjects [protected]

IPTC Subjects Information.

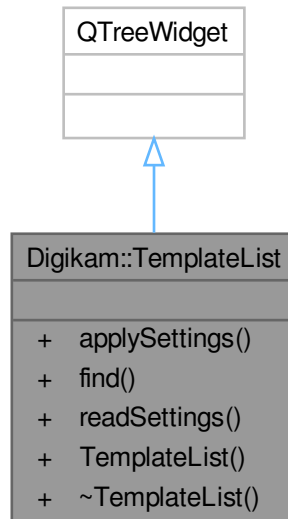
#### 6.1431.2.11 m\_templateTitle

`QString` Digikam::Template::m\_templateTitle [protected]

`Template` title used internally. This value always exist and cannot be empty.

## 6.1432 Digikam::TemplateList Class Reference

Inheritance diagram for Digikam::TemplateList:

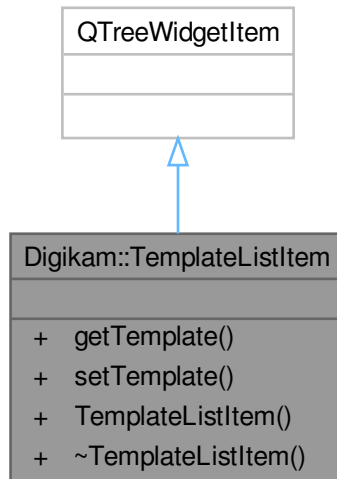


### Public Member Functions

- void **applySettings** ()
- [TemplateListItem](#) \* **find** (const QString &title)
- void **readSettings** ()
- **TemplateList** (QWidget \*const parent=nullptr)

## 6.1433 Digikam::TemplateListItem Class Reference

Inheritance diagram for Digikam::TemplateListItem:

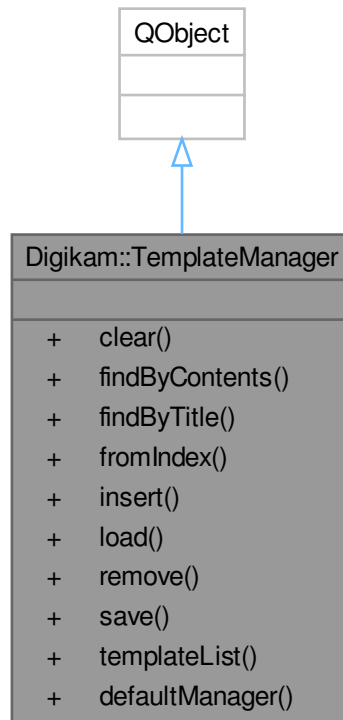


### Public Member Functions

- `Template` **getTemplate** () const
- void **setTemplate** (const `Template` &t)
- **TemplateListItem** (`QTreeWidgetItem *const parent`, const `Template` &t)

## 6.1434 Digikam::TemplateManager Class Reference

Inheritance diagram for Digikam::TemplateManager:



### Signals

- void **signalTemplateAdded** (const [Template](#) &)
- void **signalTemplateRemoved** (const [Template](#) &)

### Public Member Functions

- void **clear** ()
- [Template](#) **findByContents** (const [Template](#) &tref) const
- [Template](#) **findByTitle** (const QString &title) const
- [Template](#) **fromIndex** (int index) const
- void **insert** (const [Template](#) &t)
- bool **load** ()
- void **remove** (const [Template](#) &t)
- bool **save** ()
- QList< [Template](#) > **templateList** () const

### Static Public Member Functions

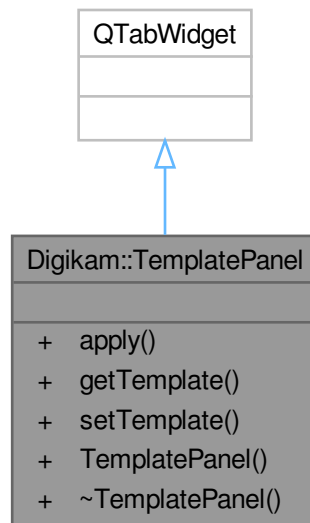
- static [TemplateManager](#) \* **defaultManager** ()

## Friends

- class `TemplateManagerCreator`

## 6.1435 Digikam::TemplatePanel Class Reference

Inheritance diagram for Digikam::TemplatePanel:



## Public Types

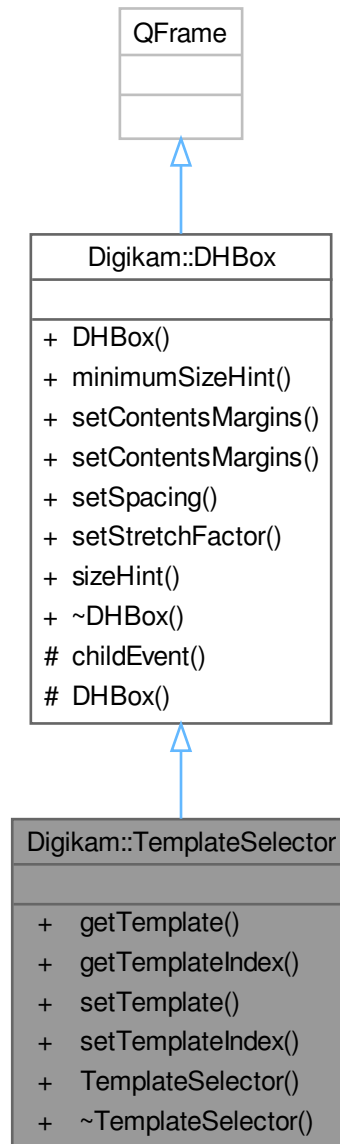
- enum `TemplateTab` { `RIGHTS =0` , `LOCATION` , `CONTACT` , `SUBJECTS` }

## Public Member Functions

- void `apply` ()
- `Template` `getTemplate` () const
- void `setTemplate` (const `Template` &t)
- `TemplatePanel` (`QWidget *const parent=nullptr`)

## 6.1436 Digikam::TemplateSelector Class Reference

Inheritance diagram for Digikam::TemplateSelector:



### Public Types

- enum **SelectorItems** { **REMOVETEMPLATE** = 0 , **DONTCHANGE** = 1 }

### Signals

- void **signalTemplateSelected** ()

### Public Member Functions

- [Template](#) **getTemplate** () const
- int **getTemplateIndex** () const
- void **setTemplate** (const [Template](#) &t)
- void **setTemplateIndex** (int i)
- **TemplateSelector** (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentMargins** (const QMargins &margins)
- void **setContentMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

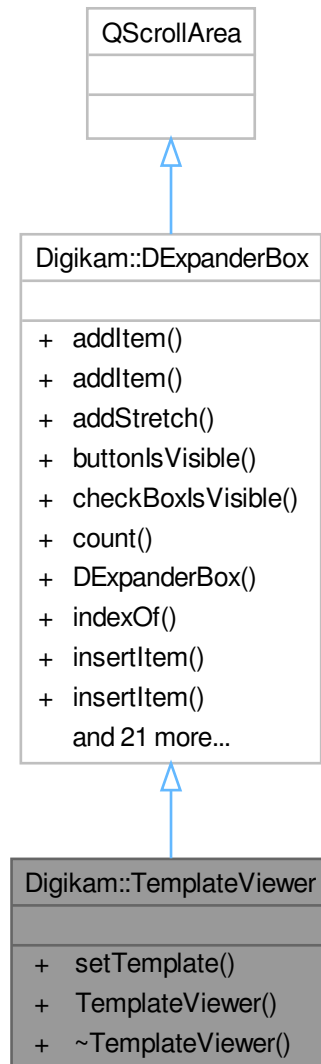
### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1437 Digikam::TemplateViewer Class Reference

Inheritance diagram for Digikam::TemplateViewer:



### Public Member Functions

- void **setTemplate** (const [Template](#) &t)
- **TemplateViewer** (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DExpanderBox](#)

- void **addItem** (QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)



- void **addItem** (QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **addStretch** ()
- bool **buttonIsVisible** (int index) const
- bool **checkboxIsVisible** (int index) const
- int **count** () const
- **DExpanderBox** (QWidget \*const parent=nullptr)
- int **indexOf** (DLabelExpander \*const widget) const
- void **insertItem** (int index, QWidget \*const w, const QIcon &icon, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertItem** (int index, QWidget \*const w, const QString &txt, const QString &objName, bool expandBydefault)
- void **insertStretch** (int index)
- bool **isChecked** (int index) const
- bool **isItemEnabled** (int index) const
- bool **isItemExpanded** (int index) const
- QIcon **itemIcon** (int index) const
- QString **itemText** (int index) const
- QString **itemToolTip** (int index) const
- virtual void **readSettings** (KConfigGroup &group)
- void **removeItem** (int index)
- void **setButtonIcon** (int index, const QIcon &icon)
- void **setButtonVisible** (int index, bool b)
- void **setCheckBoxVisible** (int index, bool b)
- void **setChecked** (int index, bool b)
- void **setItemEnabled** (int index, bool enabled)
- void **setItemExpanded** (int index, bool b)
- void **setItemIcon** (int index, const QIcon &icon)
- void **setItemText** (int index, const QString &txt)
- void **setItemToolTip** (int index, const QString &tip)
- DLabelExpander \* **widget** (int index) const
- virtual void **writeSettings** (KConfigGroup &group)

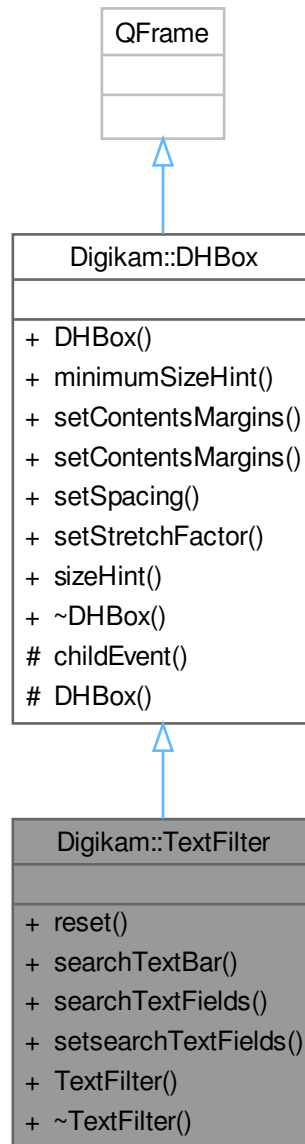
#### Additional Inherited Members

#### Signals inherited from [Digikam::DExpanderBox](#)

- void **signalItemButtonPressed** (int index)
- void **signalItemExpanded** (int index, bool b)
- void **signalItemToggled** (int index, bool b)

## 6.1438 Digikam::TextFilter Class Reference

Inheritance diagram for Digikam::TextFilter:



### Signals

- void **signalSearchTextFilterSettings** (const [SearchTextFilterSettings](#) &)

### Public Member Functions

- void **reset** ()

- [SearchTextBar](#) \* **searchTextBar** () const
- [SearchTextFilterSettings::TextFilterFields](#) **searchTextFields** ()
- void **setsearchTextFields** ([SearchTextFilterSettings::TextFilterFields](#) fields)
- **TextFilter** ([QWidget](#) \*const parent)

### Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** ([QWidget](#) \*const parent=nullptr)
- [QSize](#) **minimumSizeHint** () const override
- void **setContentsMargins** (const [QMargins](#) &argins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** ([QWidget](#) \*const widget, int stretch)
- [QSize](#) **sizeHint** () const override

### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** ([QChildEvent](#) \*e) override
- **DHBox** (bool vertical, [QWidget](#) \*const parent)

## 6.1439 Digikam::TextureContainer Class Reference

### Public Types

- enum **TextureTypes** {  
    **PaperTexture** = 0 , **Paper2Texture** , **FabricTexture** , **BurlapTexture** ,  
    **BricksTexture** , **Bricks2Texture** , **CanvasTexture** , **MarbleTexture** ,  
    **Marble2Texture** , **BlueJeanTexture** , **CellWoodTexture** , **MetalWireTexture** ,  
    **ModernTexture** , **WallTexture** , **MossTexture** , **StoneTexture** }

### Static Public Member Functions

- static [QString](#) **getTexturePath** (int texture)

### Public Attributes

- int **blendGain** = 200
- int **textureType** = **MarbleTexture**

## 6.1440 Digikam::TextureFilter Class Reference

Inheritance diagram for Digikam::TextureFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- void [readParameters](#) (const [FilterAction](#) &action) override
- [TextureFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [TextureContainer](#) &settings=[TextureContainer](#)())
- [TextureFilter](#) (QObject \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & [filterName](#) ()
- int [filterVersion](#) () const
- [DImg](#) [getTargetImage](#) ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString [readParametersError](#) (const [FilterAction](#) &actionThatFailed) const
- void [setFilterName](#) (const QString &name)
- void [setFilterVersion](#) (int version)
- void [setOriginalImage](#) (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > [supportedVersions](#) () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool [isFinished](#) () const
- bool [isRunning](#) () const
- QThread::Priority [priority](#) () const
- void [setEmitSignals](#) (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State [state](#) () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int [CurrentVersion](#) ()
- static QString [DisplayableName](#) ()
- static QString [FilterIdentifier](#) ()
- static QList< int > [SupportedVersions](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.1440.1 Member Function Documentation

### 6.1440.1.1 filterAction()

`FilterAction` Digikam::TextureFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1440.1.2 filterIdentifier()

`QString` Digikam::TextureFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

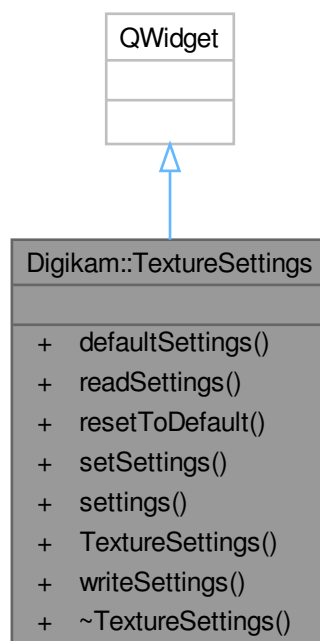
### 6.1440.1.3 readParameters()

```
void Digikam::TextureFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1441 Digikam::TextureSettings Class Reference

Inheritance diagram for Digikam::TextureSettings:



## Signals

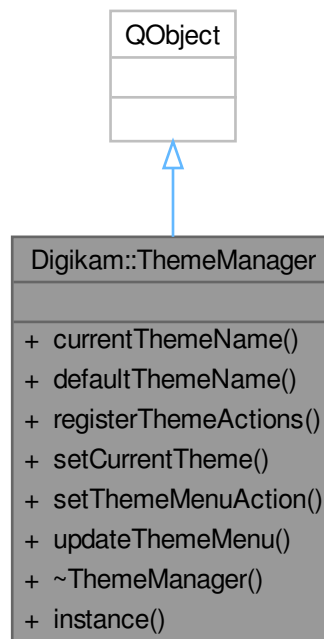
- void **signalSettingsChanged** ()

## Public Member Functions

- [TextureContainer](#) **defaultSettings** () const
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **setSettings** (const [TextureContainer](#) &settings)
- [TextureContainer](#) **settings** () const
- **TextureSettings** (QWidget \*const parent)
- void **writeSettings** (KConfigGroup &group)

## 6.1442 Digikam::ThemeManager Class Reference

Inheritance diagram for Digikam::ThemeManager:



## Classes

- class [Private](#)

## Signals

- void **signalThemeChanged** ()



### Public Member Functions

- QString **currentThemeName** () const
- QString **defaultThemeName** () const
- void **registerThemeActions** (DXmlGuiWindow \*const win)
- void **setCurrentTheme** (const QString &name)
- void **setThemeMenuAction** (QMenu \*const action)
- void **updateThemeMenu** ()

### Static Public Member Functions

- static ThemeManager \* **instance** ()

### Friends

- class ThemeManagerCreator

## 6.1443 Digikam::ThemeManager::Private Class Reference

### Public Member Functions

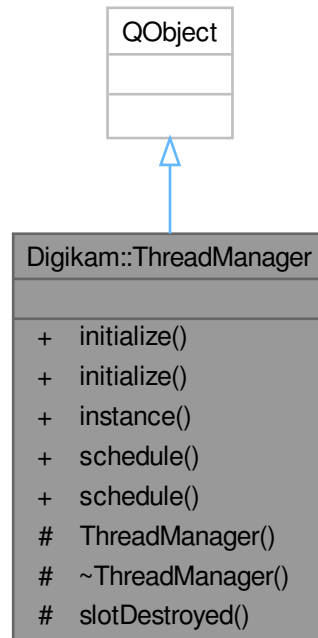
- QPixmap **createSchemePreviewIcon** (const KSharedConfigPtr &config) const

### Public Attributes

- const QString **defaultThemeName**
- QMap< QString, QString > **themeMap**  
*map< theme name, theme config path>*
- QMenu \* **themeMenuAction** = nullptr
- QActionGroup \* **themeMenuActionGroup** = nullptr

## 6.1444 Digikam::ThreadManager Class Reference

Inheritance diagram for Digikam::ThreadManager:



### Public Slots

- void **schedule** (QRunnable \*runnable)
- void **schedule** ([WorkerObject](#) \*object)

### Public Member Functions

- void **initialize** ([DynamicThread](#) \*const dynamicThread)
- void **initialize** ([WorkerObject](#) \*const object)

### Static Public Member Functions

- static [ThreadManager](#) \* **instance** ()

### Protected Slots

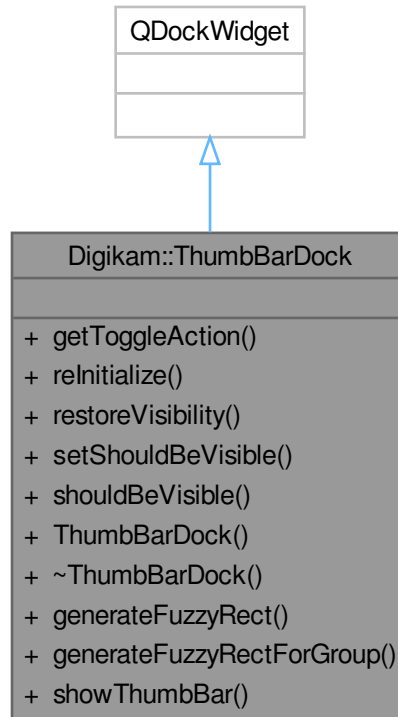
- void **slotDestroyed** (QObject \*object)

**Friends**

- class **ThreadManagerCreator**

**6.1445 Digikam::ThumbBarDock Class Reference**

Inheritance diagram for Digikam::ThumbBarDock:

**Public Types**

- enum **Visibility** { **WAS\_HIDDEN** , **WAS\_SHOWN** , **SHOULD\_BE\_HIDDEN** , **SHOULD\_BE\_SHOWN** }

**Public Slots**

- void **showThumbBar** (bool)

**Public Member Functions**

- QAction \* **getToggleAction** (QObject \*const parent, const QString &caption=QString()) const
- void **reinitialize** ()
- void **restoreVisibility** ()
- void **setShouldBeVisible** (bool)
- bool **shouldBeVisible** () const
- **ThumbBarDock** (QWidget \*const parent=nullptr, Qt::WindowFlags flags=Qt::WindowFlags())

## Static Public Member Functions

- static QPixmap **generateFuzzyRect** (const QSize &size, const QColor &color, int radius, const QColor &fill←  
Color=Qt::transparent)
- static QPixmap **generateFuzzyRectForGroup** (const QSize &size, const QColor &color, int radius)

### 6.1445.1 Detailed Description

A dock widget specifically designed for thumbnail bars (class ThumbNailView or one of its descendants). It provides the same look as a toolbar.

### 6.1445.2 Member Function Documentation

#### 6.1445.2.1 getToggleAction()

```
QAction * Digikam::ThumbBarDock::getToggleAction (
    QObject *const parent,
    const QString & caption = QString() ) const
```

Return an Action to show and hide the thumbnail bar.

#### 6.1445.2.2 reInitialize()

```
void Digikam::ThumbBarDock::reInitialize ( )
```

Measure the orientation and size of the widget and adjust the containing thumbnail bar accordingly. Normally not needed, but useful when the dock widget has changed location and/or size and the appropriate signals aren't emitted.

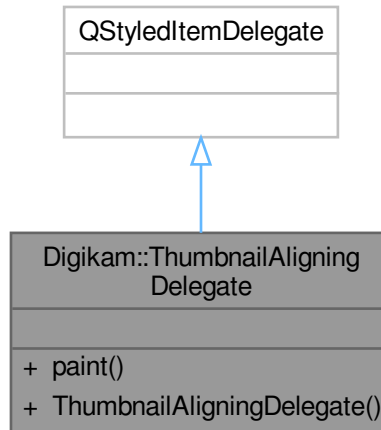
#### 6.1445.2.3 shouldBeVisible()

```
bool Digikam::ThumbBarDock::shouldBeVisible ( ) const
```

The normal show() and hide() functions don't apply that well, because there are two orthogonal reasons to hide the thumbbar: the user doesn't want it, and the window with the thumbbar isn't shown. The restoreVisibility() function will set the visibility status to what it should be according to the user setting. The setShouldBeVisible() function can change this setting. showThumbBar() can be used to hide and show the thumbbar according to the user preference. [shouldBeVisible\(\)](#) tells whether the thumbbar should be shown according to the user.

## 6.1446 Digikam::ThumbnailAligningDelegate Class Reference

Inheritance diagram for Digikam::ThumbnailAligningDelegate:



### Public Member Functions

- void **paint** (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- **ThumbnailAligningDelegate** (QObject \*const parent=nullptr)

## 6.1447 Digikam::ThumbnailCreator Class Reference

### Classes

- class [Private](#)

### Public Types

- enum **StorageMethod** { **NoThumbnailStorage** , **FreeDesktopStandard** , **ThumbnailDatabase** }

### Public Member Functions

- void [deleteThumbnailsFromDisk](#) (const QString &filePath) const
- QString [errorString](#) () const
- QImage [load](#) (const [ThumbnailIdentifier](#) &identifier, bool onlyStorage=false) const
- QImage [loadDetail](#) (const [ThumbnailIdentifier](#) &identifier, const QRect &detailRect, bool onlyStorage=false) const
- void [pregenerate](#) (const [ThumbnailIdentifier](#) &identifier) const
- void [pregenerateDetail](#) (const [ThumbnailIdentifier](#) &identifier, const QRect &detailRect) const

- void [setExifRotate](#) (bool rotate)
- void [setLoadingProperties](#) (DImgLoaderObserver \*const observer, const DRawDecoding &settings)
- void [setOnlyLargeThumbnails](#) (bool onlyLarge)
- void [setRemoveAlphaChannel](#) (bool removeAlpha)
- void [setThumbnailInfoProvider](#) (ThumbnailInfoProvider \*const provider)
- void [setThumbnailSize](#) (int thumbnailSize)
- void [store](#) (const QString &path, const QImage &image) const
- void [storeDetailThumbnail](#) (const QString &path, const QRect &detailRect, const QImage &image) const
- int [storedSize](#) () const
- [ThumbnailCreator](#) (int thumbnailSize, StorageMethod method)
- [ThumbnailCreator](#) (StorageMethod method)
- int [thumbnailSize](#) () const

### Static Public Member Functions

- static [ThumbnailInfo fileThumbnailInfo](#) (const QString &path)
- static QString [identifierForDetail](#) (const [ThumbnailInfo](#) &info, const QRect &rect)

## 6.1447.1 Constructor & Destructor Documentation

### 6.1447.1.1 ThumbnailCreator() [1/2]

```
Digikam::ThumbnailCreator::ThumbnailCreator (
    StorageMethod method ) [explicit]
```

Create a thumbnail creator object. You must call [setThumbnailSize](#) before load.

### 6.1447.1.2 ThumbnailCreator() [2/2]

```
Digikam::ThumbnailCreator::ThumbnailCreator (
    int thumbnailSize,
    StorageMethod method )
```

Create a thumbnail creator object, and set the thumbnail size.

## 6.1447.2 Member Function Documentation

### 6.1447.2.1 deleteThumbnailsFromDisk()

```
void Digikam::ThumbnailCreator::deleteThumbnailsFromDisk (
    const QString & filePath ) const
```

Deletes all available thumbnails from the on-disk thumbnail cache. A subsequent call to [load\(\)](#) will recreate the thumbnail.

### 6.1447.2.2 errorString()

```
QString Digikam::ThumbnailCreator::errorString ( ) const
```

Returns the last error that occurred. It is valid if load returned a null QImage object.

### 6.1447.2.3 fileThumbnailInfo()

```
ThumbnailInfo Digikam::ThumbnailCreator::fileThumbnailInfo (
    const QString & path ) [static]
```

Creates a default [ThumbnailInfo](#) for the given path using [QFileInfo](#) only

### 6.1447.2.4 identifierForDetail()

```
QString Digikam::ThumbnailCreator::identifierForDetail (
    const ThumbnailInfo & info,
    const QRect & rect ) [static]
```

Returns the customIdentifier for the detail thumbnail

### 6.1447.2.5 load()

```
QImage Digikam::ThumbnailCreator::load (
    const ThumbnailIdentifier & identifier,
    bool onlyStorage = false ) const
```

Create a thumbnail for the specified file.

### 6.1447.2.6 loadDetail()

```
QImage Digikam::ThumbnailCreator::loadDetail (
    const ThumbnailIdentifier & identifier,
    const QRect & detailRect,
    bool onlyStorage = false ) const
```

Creates a thumbnail for the specified detail of the file. A suitable custom identifier (for cache key etc.) is inserted as `image.text("customIdentifier")`.

### 6.1447.2.7 pregenerate()

```
void Digikam::ThumbnailCreator::pregenerate (
    const ThumbnailIdentifier & identifier ) const
```

Ensures that the thumbnail is pregenerated in the database, but does not load it from there.

### 6.1447.2.8 setExifRotate()

```
void Digikam::ThumbnailCreator::setExifRotate (
    bool rotate )
```

Set the Exif rotation property. If `exifRotate` is true, the thumbnail will be rotated according to the Exif information. Default value is true.

### 6.1447.2.9 setLoadingProperties()

```
void Digikam::ThumbnailCreator::setLoadingProperties (
    DImgLoaderObserver *const observer,
    const DRawDecoding & settings )
```

If you plan to load thumbnail from the context of the threadimageio framework, you can specify the relevant parameters. They will be passed if a thumbnail is created by loading with [DImg](#). Note that [DImg](#) is not used in most cases (Raw files, JPEG)

### 6.1447.2.10 setOnlyLargeThumbnails()

```
void Digikam::ThumbnailCreator::setOnlyLargeThumbnails (
    bool onlyLarge )
```

If you enable this property, the thumbnail creator will create only large thumbnails on disk (256x256 as described in FreeDesktop paper). Normally, for requested sizes below 128, thumbnails of 128x128 will be cached on disk. Default value is false.

### 6.1447.2.11 setRemoveAlphaChannel()

```
void Digikam::ThumbnailCreator::setRemoveAlphaChannel (
    bool removeAlpha )
```

If you enable this property, the returned QImage objects will not have an alpha channel. Images with transparency will be blended over an opaque background.

### 6.1447.2.12 setThumbnailInfoProvider()

```
void Digikam::ThumbnailCreator::setThumbnailInfoProvider (
    ThumbnailInfoProvider *const provider )
```

Set a [ThumbnailInfoProvider](#) to provide custom ThumbnailInfos

### 6.1447.2.13 setThumbnailSize()

```
void Digikam::ThumbnailCreator::setThumbnailSize (
    int thumbnailSize )
```

Sets the thumbnail size. This is the maximum size of the QImage returned by load.

### 6.1447.2.14 store()

```
void Digikam::ThumbnailCreator::store (
    const QString & path,
    const QImage & image ) const
```

Store the given image as thumbnail of the given path. Image should at least have [storedSize\(\)](#).



### 6.1447.2.15 storedSize()

```
int Digikam::ThumbnailCreator::storedSize ( ) const
```

Return the stored image size, the size of the image that is stored on disk (according to Storage Method). This size is possibly larger than thumbnailSize. Possible values: 128 or 256.

### 6.1447.2.16 thumbnailSize()

```
int Digikam::ThumbnailCreator::thumbnailSize ( ) const
```

Return the thumbnail size, the maximum size of the QImage returned by load.

## 6.1448 Digikam::ThumbnailCreator::Private Class Reference

### Public Member Functions

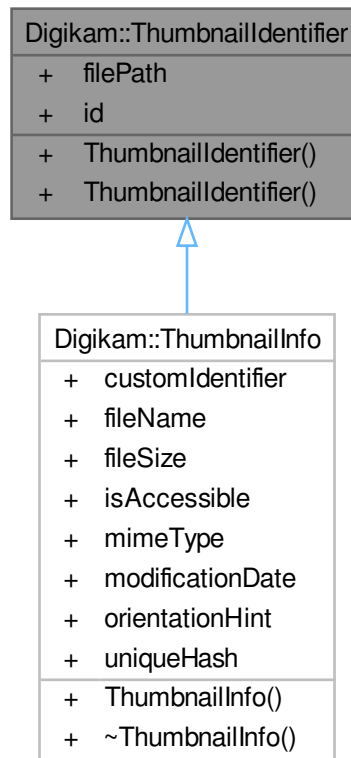
- int **storageSize** ( ) const

### Public Attributes

- QImage **alphaImage**
- QString **bigThumbPath**
- int **dbIdForReplacement** = -1
- QString **digiKamFingerPrint** = QLatin1String("Digikam Thumbnail Generator")
- QString **error**
- bool **exifRotate** = true
- [DRawDecoding](#) **fastRawSettings**
- [ThumbnailInfoProvider](#) \* **infoProvider** = nullptr
- [DImgLoaderObserver](#) \* **observer** = nullptr
- bool **onlyLargeThumbnails** = false
- [DRawDecoding](#) **rawSettings**
- bool **removeAlphaChannel** = true
- QString **smallThumbPath**
- int **thumbnailSize** = 0
- ThumbnailCreator::StorageMethod **thumbnailStorage** = ThumbnailCreator::FreeDesktopStandard

## 6.1449 Digikam::ThumbnailIdentifier Class Reference

Inheritance diagram for Digikam::ThumbnailIdentifier:



### Public Member Functions

- **ThumbnailIdentifier** (const QString &path)

### Public Attributes

- QString `filePath`
- qlonglong `id` = 0

### 6.1449.1 Member Data Documentation

#### 6.1449.1.1 filePath

QString Digikam::ThumbnailIdentifier::filePath

The file path from which the thumbnail shall be generated

## 6.1449.1.2 id

```
qlonglong Digikam::ThumbnailIdentifier::id = 0
```

The database id, which needs to be translated to uniqueHash + fileSize

## 6.1450 Digikam::ThumbnailImage Class Reference

## Public Member Functions

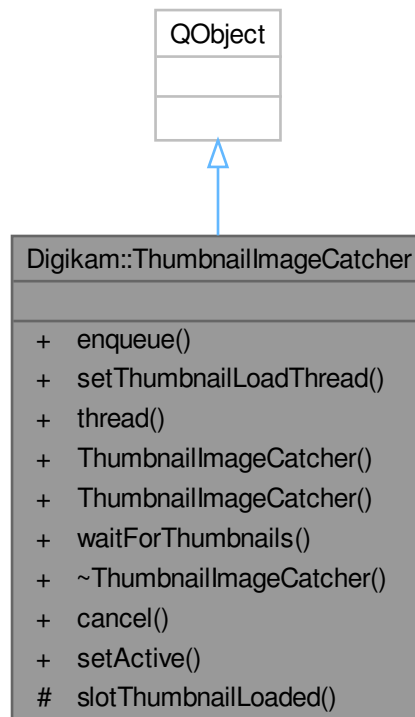
- bool **isNull** () const

## Public Attributes

- int **exifOrientation** = DMetadata::ORIENTATION\_UNSPECIFIED
- QImage **qimage**

## 6.1451 Digikam::ThumbnailImageCatcher Class Reference

Inheritance diagram for Digikam::ThumbnailImageCatcher:



**Classes**

- class [Private](#)

**Public Slots**

- void [cancel](#) ()
- void [setActive](#) (bool active)

**Public Member Functions**

- int [enqueue](#) ()
- void **setThumbnailLoadThread** ([ThumbnailLoadThread](#) \*const thread)
- [ThumbnailLoadThread](#) \* **thread** () const
- [ThumbnailImageCatcher](#) (QObject \*const parent=nullptr)
- **ThumbnailImageCatcher** ([ThumbnailLoadThread](#) \*const thread, QObject \*const parent=nullptr)
- QList< QImage > **waitForThumbnails** ()

**Protected Slots**

- void **slotThumbnailLoaded** (const [LoadingDescription](#) &, const QImage &)

**6.1451.1 Constructor & Destructor Documentation****6.1451.1.1 ThumbnailImageCatcher()**

```
Digikam::ThumbnailImageCatcher::ThumbnailImageCatcher (
    QObject *const parent = nullptr ) [explicit]
```

Use this class to get a thumbnail synchronously.

1. Create the [ThumbnailImageCatcher](#) object with your [ThumbnailLoadThread](#)
2. a) Request a thumbnail b) Call [enqueue\(\)](#)
3. Call [waitForThumbnails](#) which returns the thumbnail QImage(s).

Note: Not meant for loading QPixmap thumbnails.

**6.1451.2 Member Function Documentation****6.1451.2.1 cancel**

```
void Digikam::ThumbnailImageCatcher::cancel ( ) [slot]
```

If the catcher is waiting in [waitForThumbnails\(\)](#) in a different thread, cancels the waiting. The results will be returned as received so far.

### 6.1451.2.2 enqueue()

```
int Digikam::ThumbnailImageCatcher::enqueue ( )
```

After requesting a thumbnail from the thread, call [enqueue\(\)](#) each time. Enqueue records the requested loading operation in an internal list. A loading operation can result in the return of more than one thumbnail, so [enqueue\(\)](#) returns the number of expected results. Then call [waitForThumbnails](#). The returned list is the sum of previous calls to [enqueue](#), one entry per expected result, in order. If stopped prematurely or loading failed, the respective entries will be null.

### 6.1451.2.3 setActive

```
void Digikam::ThumbnailImageCatcher::setActive (
    bool active ) [slot]
```

The catcher is active per default after construction. Deactivate it if you use the catcher as a longer-lived object and do not use it for some time, then activate it before you request a thumbnail from the thread again.

## 6.1452 Digikam::ThumbnailImageCatcher::Private Class Reference

### Classes

- class [CatcherResult](#)

### Public Types

- enum [CatcherState](#) { [Inactive](#) , [Accepting](#) , [Waiting](#) , [Quitting](#) }

### Public Member Functions

- void [harvest](#) (const [LoadingDescription](#) &description, const QImage &image)
- void [reset](#) ()

### Public Attributes

- bool [active](#) = true
- QWaitCondition [condVar](#)
- QList< [Private::CatcherResult](#) > [intermediate](#)
- QMutex [mutex](#)
- CatcherState [state](#) = Inactive
- QList< [Private::CatcherResult](#) > [tasks](#)
- [ThumbnailLoadThread](#) \* [thread](#) = nullptr

## 6.1453 Digikam::ThumbnailImageCatcher::Private::CatcherResult Class Reference

### Public Member Functions

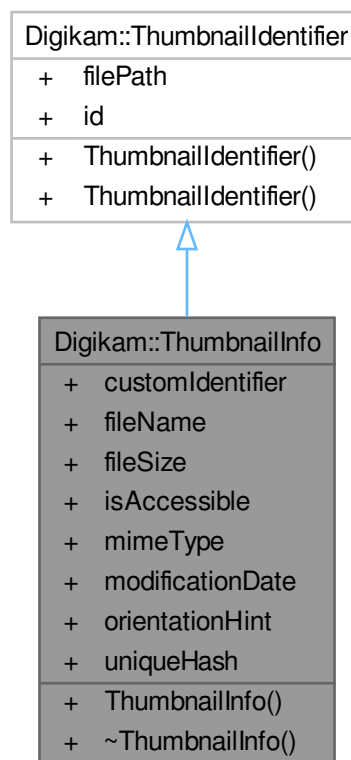
- [CatcherResult](#) (const [LoadingDescription](#) &d)
- [CatcherResult](#) (const [LoadingDescription](#) &d, const QImage &image)

**Public Attributes**

- [LoadingDescription](#) **description**
- QImage **image**
- bool **received** = false

**6.1454 Digikam::ThumbnailInfo Class Reference**

Inheritance diagram for Digikam::ThumbnailInfo:

**Public Attributes**

- QString [customIdentifier](#)
- QString [fileName](#)
- qlonglong **fileSize** = 0
- bool [isAccessible](#) = false
- QString [mimeType](#)
- QDateTime [modificationDate](#)
- int [orientationHint](#) = DMetadata::ORIENTATION\_UNSPECIFIED
- QString [uniqueHash](#)

## Public Attributes inherited from [Digikam::ThumbnailIdentifier](#)

- QString [filePath](#)
- qlonglong [id](#) = 0

## Additional Inherited Members

## Public Member Functions inherited from [Digikam::ThumbnailIdentifier](#)

- [ThumbnailIdentifier](#) (const QString &path)

## 6.1454.1 Member Data Documentation

### 6.1454.1.1 customIdentifier

```
QString Digikam::ThumbnailInfo::customIdentifier
```

A custom identifier, if neither filePath nor uniqueHash are applicable.

### 6.1454.1.2 fileName

```
QString Digikam::ThumbnailInfo::fileName
```

The file name (the name, not the directory)

### 6.1454.1.3 isAccessible

```
bool Digikam::ThumbnailInfo::isAccessible = false
```

If the original file is at all accessible on disk. May be false if a file on a removable device is used.

### 6.1454.1.4 mimeType

```
QString Digikam::ThumbnailInfo::mimeType
```

The mime type of the original file. Currently "image" or "video" otherwise empty.

### 6.1454.1.5 modificationDate

```
QDateTime Digikam::ThumbnailInfo::modificationDate
```

The modification date of the original file. Thumbnail will be regenerated if thumb's modification date is older than this.

### 6.1454.1.6 orientationHint

```
int Digikam::ThumbnailInfo::orientationHint = DMetadata::ORIENTATION_UNSPECIFIED
```

Gives a hint at the orientation of the image. This can be used to supersede the Exif information in the file. Will not be used if DMetadata::ORIENTATION\_UNSPECIFIED (default value)

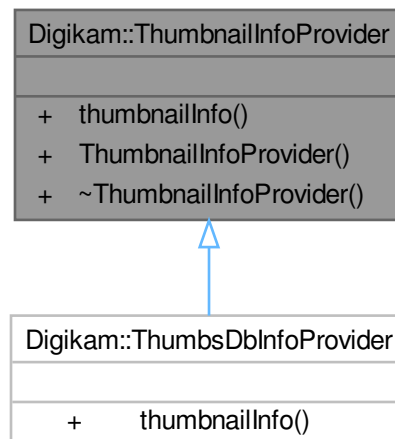
### 6.1454.1.7 uniqueHash

```
QString Digikam::ThumbnailInfo::uniqueHash
```

If available, the uniqueHash + fileSize pair for identification of the original file by content.

## 6.1455 Digikam::ThumbnailInfoProvider Class Reference

Inheritance diagram for Digikam::ThumbnailInfoProvider:



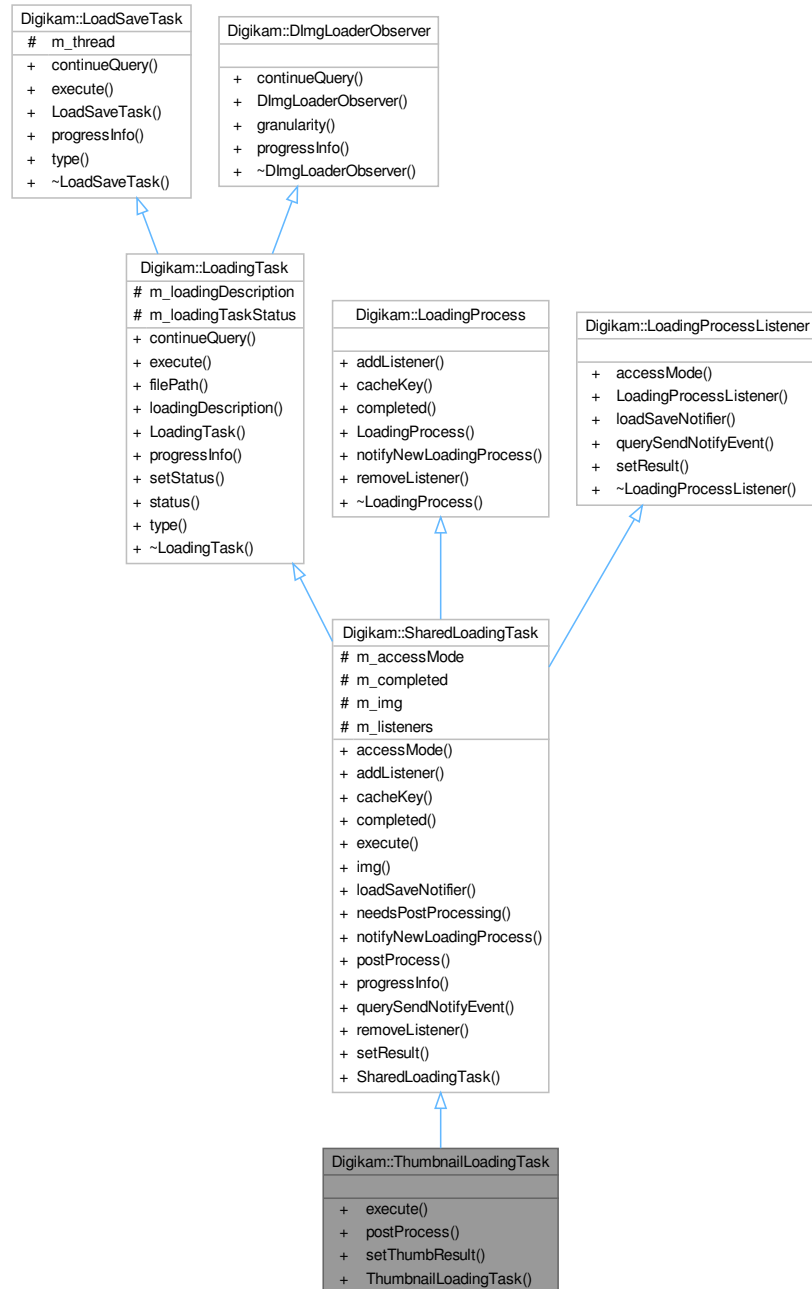
### Public Member Functions

- virtual [ThumbnailInfo](#) **thumbnailInfo** (const [ThumbnailIdentifier](#) &)=0



## 6.1456 Digikam::ThumbnailLoadingTask Class Reference

Inheritance diagram for Digikam::ThumbnailLoadingTask:



### Public Member Functions

- void `execute` () override
- void `postProcess` () override
- void `setThumbResult` (const `LoadingDescription` &loadingDescription, const `QImage` &qimage)
- `ThumbnailLoadingTask` (`LoadSaveThread` \*const thread, const `LoadingDescription` &description)

## Public Member Functions inherited from [Digikam::SharedLoadingTask](#)

- [LoadSaveThread::AccessMode](#) `accessMode` () const override
- void `addListener` ([LoadingProcessListener](#) \*const listener) override
- QString `cacheKey` () const override
- bool `completed` () const override
- [DImg](#) `img` () const
- [LoadSaveNotifier](#) \* `loadSaveNotifier` () const override
- bool `needsPostProcessing` () const
- void `notifyNewLoadingProcess` ([LoadingProcess](#) \*const process, const [LoadingDescription](#) &description) override
- void `progressInfo` (float progress) override
- bool `querySendNotifyEvent` () const override
- void `removeListener` ([LoadingProcessListener](#) \*const listener) override
- void `setResult` (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img) override
- **SharedLoadingTask** ([LoadSaveThread](#) \*const thread, const [LoadingDescription](#) &description, [LoadSaveThread::AccessMode](#) mode=[LoadSaveThread::AccessModeReadWrite](#), [LoadingTaskStatus](#) loadingTaskStatus=[LoadingTask↔ StatusLoading](#))

## Public Member Functions inherited from [Digikam::LoadingTask](#)

- bool `continueQuery` () override
- QString `filePath` () const
- const [LoadingDescription](#) & `loadingDescription` () const
- **LoadingTask** ([LoadSaveThread](#) \*const thread, const [LoadingDescription](#) &description, [LoadingTaskStatus](#) loadingTaskStatus=[LoadingTaskStatusLoading](#))
- void `setStatus` ([LoadingTaskStatus](#) status)
- [LoadingTaskStatus](#) `status` () const
- [TaskType](#) `type` () override

## Public Member Functions inherited from [Digikam::LoadSaveTask](#)

- **LoadSaveTask** ([LoadSaveThread](#) \*const thread)

## Public Member Functions inherited from [Digikam::DImgLoaderObserver](#)

- virtual float `granularity` ()

## Additional Inherited Members

## Public Types inherited from [Digikam::LoadingTask](#)

- enum **LoadingTaskStatus** { [LoadingTaskStatusLoading](#) , [LoadingTaskStatusPreloading](#) , [Loading↔ TaskStatusStopping](#) }

## Public Types inherited from [Digikam::LoadSaveTask](#)

- enum **TaskType** { [TaskTypeLoading](#) , [TaskTypeSaving](#) }

### Protected Attributes inherited from [Digikam::SharedLoadingTask](#)

- [LoadSaveThread::AccessMode](#) **m\_accessMode** = [LoadSaveThread::AccessModeReadWrite](#)
- volatile bool **m\_completed** = false
- [DImg](#) **m\_img**
- [QList< LoadingProcessListener \\* >](#) **m\_listeners**

### Protected Attributes inherited from [Digikam::LoadingTask](#)

- [LoadingDescription](#) **m\_loadingDescription**
- volatile [LoadingTaskStatus](#) **m\_loadingTaskStatus** = [LoadingTaskStatusLoading](#)

### Protected Attributes inherited from [Digikam::LoadSaveTask](#)

- [LoadSaveThread](#) \* **m\_thread** = nullptr

## 6.1456.1 Member Function Documentation

### 6.1456.1.1 execute()

```
void Digikam::ThumbnailLoadingTask::execute ( ) [override], [virtual]
```

Reimplemented from [Digikam::SharedLoadingTask](#).

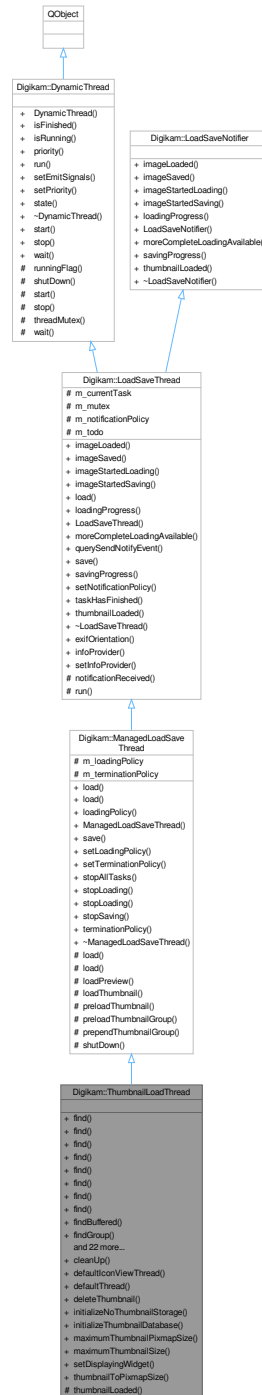
### 6.1456.1.2 postProcess()

```
void Digikam::ThumbnailLoadingTask::postProcess ( ) [override], [virtual]
```

Reimplemented from [Digikam::SharedLoadingTask](#).

## 6.1457 Digikam::ThumbnailLoadThread Class Reference

Inheritance diagram for Digikam::ThumbnailLoadThread:



### Classes

- class [Private](#)

## Signals

- void **signalThumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QPixmap &pix)  
*NOTE: See [LoadSaveThread](#) for a QImage-based [thumbnailLoaded\(\)](#) signal.*
- void **ThumbnailLoaded** (const [LoadingDescription](#) &, const QImage &)
- void **thumbnailsAvailable** ()  
*NOTE: For internal use only.*

## Signals inherited from [Digikam::LoadSaveThread](#)

- void [signalImageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img)
- void **signalImageSaved** (const QString &filePath, bool success)
- void [signalImageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription)
- void **signalImageStartedSaving** (const QString &filePath)
- void [signalLoadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress)
- void [signalMoreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription)
- void **signalSavingProgress** (const QString &filePath, float progress)
- void **signalThumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QImage &img)

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void [starting](#) ()

## Public Member Functions

- void [find](#) (const [ThumbnailIdentifier](#) &identifier)
- void **find** (const [ThumbnailIdentifier](#) &identifier, const QRect &rect)
- void **find** (const [ThumbnailIdentifier](#) &identifier, const QRect &rect, int size)
- bool [find](#) (const [ThumbnailIdentifier](#) &identifier, const QRect &rect, QPixmap &pixmap)
- bool **find** (const [ThumbnailIdentifier](#) &identifier, const QRect &rect, QPixmap &pixmap, int size, bool onlyStorage=false)
- void [find](#) (const [ThumbnailIdentifier](#) &identifier, int size)
- bool [find](#) (const [ThumbnailIdentifier](#) &identifier, QPixmap &pixmap)
- bool [find](#) (const [ThumbnailIdentifier](#) &identifier, QPixmap &pixmap, int size, bool onlyStorage=false)
- bool [findBuffered](#) (const [ThumbnailIdentifier](#) &identifier, const QRect &rect, QPixmap &pixmap, int size)
- void **findGroup** (const QList< QPair< [ThumbnailIdentifier](#), QRect > > &filePathAndRects)
- void **findGroup** (const QList< QPair< [ThumbnailIdentifier](#), QRect > > &filePathsAndRects, int size)
- void [findGroup](#) (QList< [ThumbnailIdentifier](#) > &identifiers)
- void **findGroup** (QList< [ThumbnailIdentifier](#) > &identifiers, int size)
- QList< [LoadingDescription](#) > [lastDescriptions](#) () const
- void [load](#) (const [LoadingDescription](#) &description)
- int [pixmapToThumbnailSize](#) (int size) const
- void [pregenerateGroup](#) (const QList< [ThumbnailIdentifier](#) > &identifiers)
- void **pregenerateGroup** (const QList< [ThumbnailIdentifier](#) > &identifiers, int size)
- void [preload](#) (const [ThumbnailIdentifier](#) &identifier)
- void **preload** (const [ThumbnailIdentifier](#) &identifier, int size)
- void **preloadGroup** (QList< [ThumbnailIdentifier](#) > &identifiers)
- void **preloadGroup** (QList< [ThumbnailIdentifier](#) > &identifiers, int size)
- void [setHighlightPixmap](#) (bool highlight)
- void [setPixmapRequested](#) (bool wantPixmap)

- void [setSendSurrogatePixmap](#) (bool send)
  - void [setThumbnailSize](#) (int size, bool forFace=false)
  - void [storeDetailThumbnail](#) (const QString &filePath, const QRect &detailRect, const QImage &image, bool isFace=false)
  - int **storedSize** () const
  - [ThumbnailCreator](#) \* **thumbnailCreator** () const
- NOTE: For internal use - may only be used from the thread.*
- **ThumbnailLoadThread** (QObject \*const parent=nullptr)
  - int [thumbnailToPixmapSize](#) (int size) const

### Public Member Functions inherited from [Digikam::ManagedLoadSaveThread](#)

- void [load](#) (const [LoadingDescription](#) &description)
- void **load** (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- [LoadingPolicy](#) **loadingPolicy** () const
- [ManagedLoadSaveThread](#) (QObject \*const parent=nullptr)
- void [save](#) (const [DImg](#) &image, const QString &filePath, const QString &format)
- void [setLoadingPolicy](#) ([LoadingPolicy](#) policy)
- void **setTerminationPolicy** ([TerminationPolicy](#) terminationPolicy)
- void **stopAllTasks** ()
- void [stopLoading](#) (const [LoadingDescription](#) &desc, [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopLoading](#) (const QString &filePath=QString(), [LoadingTaskFilter](#) filter=[LoadingTaskFilterAll](#))
- void [stopSaving](#) (const QString &filePath=QString())
- [TerminationPolicy](#) **terminationPolicy** () const

### Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- void [imageLoaded](#) (const [LoadingDescription](#) &loadingDescription, const [DImg](#) &img) override
- void [imageSaved](#) (const QString &filePath, bool success) override
- void [imageStartedLoading](#) (const [LoadingDescription](#) &loadingDescription) override
- void [imageStartedSaving](#) (const QString &filePath) override
- void [load](#) (const [LoadingDescription](#) &description)
- void [loadingProgress](#) (const [LoadingDescription](#) &loadingDescription, float progress) override
- **LoadSaveThread** (QObject \*const parent=nullptr)
- void [moreCompleteLoadingAvailable](#) (const [LoadingDescription](#) &oldLoadingDescription, const [LoadingDescription](#) &newLoadingDescription) override
- virtual bool **querySendNotifyEvent** () const
- void [save](#) (const [DImg](#) &image, const QString &filePath, const QString &format)
- void [savingProgress](#) (const QString &filePath, float progress) override
- void **setNotificationPolicy** ([NotificationPolicy](#) notificationPolicy)
- virtual void **taskHasFinished** ()
- [~LoadSaveThread](#) () override

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- [QThread::Priority](#) **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static void **cleanUp** ()
- static [ThumbnailLoadThread](#) \* **defaultIconViewThread** ()
- static [ThumbnailLoadThread](#) \* **defaultThread** ()
- static void **deleteThumbnail** (const QString &filePath)
- static void **initializeNoThumbnailStorage** ()
- static void **initializeThumbnailDatabase** (const [DbEngineParameters](#) &params, [ThumbnailInfoProvider](#) \*const provider=nullptr)
- static int **maximumThumbnailPixmapSize** (bool withHighlighting)
- static int **maximumThumbnailSize** ()
- static void **setDisplayingWidget** (QWidget \*const widget)
- static int **thumbnailToPixmapSize** (bool withHighlight, int size)

### Static Public Member Functions inherited from [Digikam::LoadSaveThread](#)

- static int **exifOrientation** (const QString &filePath, const [DMetadata](#) &metadata, bool isRaw, bool fromRaw↔ EmbeddedPreview)
- static [LoadSaveFileInfoProvider](#) \* **infoProvider** ()
- static void **setInfoProvider** ([LoadSaveFileInfoProvider](#) \*const infoProvider)

### Protected Member Functions

- void **thumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QImage &img) override

### Protected Member Functions inherited from [Digikam::ManagedLoadSaveThread](#)

- void **load** (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void **load** (const [LoadingDescription](#) &description, [LoadingMode](#) loadingMode, [LoadingPolicy](#) policy, [AccessMode](#) mode=[AccessModeReadWrite](#))
- void **loadPreview** (const [LoadingDescription](#) &description, [LoadingPolicy](#) policy)
- void **loadThumbnail** (const [LoadingDescription](#) &description)
- void **preloadThumbnail** (const [LoadingDescription](#) &description)
- void **preloadThumbnailGroup** (const QList< [LoadingDescription](#) > &descriptions)
- void **prependThumbnailGroup** (const QList< [LoadingDescription](#) > &descriptions)
- void **shutDown** ()

### Protected Member Functions inherited from [Digikam::LoadSaveThread](#)

- void **notificationReceived** ()
- void **run** () override

### Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Additional Inherited Members

### Public Types inherited from [Digikam::ManagedLoadSaveThread](#)

- enum [LoadingMode](#) { [LoadingModeNormal](#) , [LoadingModeShared](#) }
- enum [LoadingPolicy](#) { [LoadingPolicyFirstRemovePrevious](#) , [LoadingPolicyPrepend](#) , [LoadingPolicySimplePrepend](#) , [LoadingPolicyAppend](#) , [LoadingPolicySimpleAppend](#) , [LoadingPolicyPreload](#) }
- enum [LoadingTaskFilter](#) { [LoadingTaskFilterAll](#) , [LoadingTaskFilterPreloading](#) }
- enum [TerminationPolicy](#) { [TerminationPolicyTerminateLoading](#) , [TerminationPolicyTerminatePreloading](#) , [TerminationPolicyWait](#) , [TerminationPolicyTerminateAll](#) }

### Public Types inherited from [Digikam::LoadSaveThread](#)

- enum [AccessMode](#) { [AccessModeRead](#) , [AccessModeReadWrite](#) }
- enum [NotificationPolicy](#) { [NotificationPolicyDirect](#) , [NotificationPolicyTimeLimited](#) }

### Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

### Public Slots inherited from [Digikam::DynamicThread](#)

- void [start](#) ()
- void [stop](#) ()
- void [wait](#) ()

### Protected Attributes inherited from [Digikam::ManagedLoadSaveThread](#)

- [LoadingPolicy](#) [m\\_loadingPolicy](#) = [LoadingPolicyAppend](#)
- [TerminationPolicy](#) [m\\_terminationPolicy](#) = [TerminationPolicyTerminateLoading](#)

### Protected Attributes inherited from [Digikam::LoadSaveThread](#)

- [LoadSaveTask](#) \* [m\\_currentTask](#) = nullptr
- QMutex [m\\_mutex](#)
- [NotificationPolicy](#) [m\\_notificationPolicy](#) = [NotificationPolicyTimeLimited](#)
- QList< [LoadSaveTask](#) \* > [m\\_todo](#)

## 6.1457.1 Member Function Documentation

### 6.1457.1.1 [defaultThread\(\)](#)

```
ThumbnailLoadThread * Digikam::ThumbnailLoadThread::defaultThread ( ) [static]
```

Return application-wide default thumbnail threads. It is perfectly all right to create an extra object of the class, but it is useful to have default object



### 6.1457.1.2 deleteThumbnail()

```
void Digikam::ThumbnailLoadThread::deleteThumbnail (
    const QString & filePath ) [static]
```

This is a tool to force regeneration of thumbnails. All thumbnail files for the given file will be removed from disk, and the cached instances will be removed as well. Use this method if you know that the contents of the file has changed. This method works independently from the multithreaded thumbnail loading.

### 6.1457.1.3 find() [1/5]

```
void Digikam::ThumbnailLoadThread::find (
    const ThumbnailIdentifier & identifier )
```

Find a thumbnail. This method sends the signals and does not return values like the method above. If you certainly need asynchronous return, connect with Qt::QueuedConnection to the signals. If you connect directly, the signals may be sent from within the method call.

### 6.1457.1.4 find() [2/5]

```
bool Digikam::ThumbnailLoadThread::find (
    const ThumbnailIdentifier & identifier,
    const QRect & rect,
    QPixmap & pixmap )
```

All tastes of [find\(\)](#) methods, for loading the thumbnail of a detail

### 6.1457.1.5 find() [3/5]

```
void Digikam::ThumbnailLoadThread::find (
    const ThumbnailIdentifier & identifier,
    int size )
```

Same as above, but does not use the global size, but an extra specified size.

### 6.1457.1.6 find() [4/5]

```
bool Digikam::ThumbnailLoadThread::find (
    const ThumbnailIdentifier & identifier,
    QPixmap & pixmap )
```

Find a thumbnail. If the pixmap is found in the cache, returns true and sets pixmap to the found QPixmap. If the pixmap is not found in the cache, [load\(\)](#) is called to start the loading process, false is returned and pixmap is not touched.

**6.1457.1.7 find() [5/5]**

```
bool Digikam::ThumbnailLoadThread::find (
    const ThumbnailIdentifier & identifier,
    QPixmap & pixmap,
    int size,
    bool onlyStorage = false )
```

Same as above, but does not use the global size, but an extra specified size.

**6.1457.1.8 findBuffered()**

```
bool Digikam::ThumbnailLoadThread::findBuffered (
    const ThumbnailIdentifier & identifier,
    const QRect & rect,
    QPixmap & pixmap,
    int size )
```

Find the thumbnail pixmap in the buffered cache to avoid flickering while loading a new thumbnail.

**6.1457.1.9 findGroup()**

```
void Digikam::ThumbnailLoadThread::findGroup (
    QList< ThumbnailIdentifier > & identifiers )
```

Find a group of thumbnails. The items will be loaded in order and signals will be sent. Can be used to ensure that thumbnails are loaded in a particular order

**6.1457.1.10 initializeNoThumbnailStorage()**

```
void Digikam::ThumbnailLoadThread::initializeNoThumbnailStorage ( ) [static]
```

Disable storing thumbnails in the disk cache. This shall be called once at application startup. This need not be called, then the FreeDesktop standard is used.

**6.1457.1.11 initializeThumbnailDatabase()**

```
void Digikam::ThumbnailLoadThread::initializeThumbnailDatabase (
    const DbEngineParameters & params,
    ThumbnailInfoProvider *const provider = nullptr ) [static]
```

Enable loading of thumbnails from a thumbnail database. This shall be called once at application startup. This need not be called, then the FreeDesktop standard is used. You can optionally provide a thumbnail info provider.

**6.1457.1.12 lastDescriptions()**

```
QList< LoadingDescription > Digikam::ThumbnailLoadThread::lastDescriptions ( ) const
```

Returns the descriptions used by the last call to any of the above methods. After calling single-thumbnail methods (find, preload) the list will have size 1, after the group methods (findGroup, preloadGroup, pregenerateGroup) the list can be larger than 1. There is no information if the description was ever scheduled in the thread, already processed, skipped or canceled.

### 6.1457.1.13 load()

```
void Digikam::ThumbnailLoadThread::load (
    const LoadingDescription & description )
```

Load a thumbnail. You do not need to use this method directly, it will not access the pixmap cache. Use [find\(\)](#). The [LoadingDescription](#) shall be constructed with the constructor for preview/thumbnail jobs. (in the description constructor, you need to specify file path, thumbnail size and Exif rotation)

### 6.1457.1.14 maximumThumbnailSize()

```
int Digikam::ThumbnailLoadThread::maximumThumbnailSize ( ) [static]
```

Returns the maximum possible size of a thumbnail. If you request a larger size, the thumbnail will not load. The size of the pixmap can slightly differ, especially when highlighting.

### 6.1457.1.15 pixmapToThumbnailSize()

```
int Digikam::ThumbnailLoadThread::pixmapToThumbnailSize (
    int size ) const
```

Computes the thumbnail size for the give pixmap size.

### 6.1457.1.16 pregenerateGroup()

```
void Digikam::ThumbnailLoadThread::pregenerateGroup (
    const QList< ThumbnailIdentifier > & identifiers )
```

Pregenerate the thumbnail group. No signals will be emitted when these are loaded.

### 6.1457.1.17 preload()

```
void Digikam::ThumbnailLoadThread::preload (
    const ThumbnailIdentifier & identifier )
```

Preload the thumbnail or thumbnail group. This is essentially the same as loading, but with a lower priority.

### 6.1457.1.18 setDisplayingWidget()

```
void Digikam::ThumbnailLoadThread::setDisplayingWidget (
    QWidget *const widget ) [static]
```

For color management, this sets the widget the thumbnails will be color managed for. (currently it is only possible to set one global widget)

### 6.1457.1.19 setHighlightPixmap()

```
void Digikam::ThumbnailLoadThread::setHighlightPixmap (
    bool highlight )
```

If you enable this, a highlighting border will be drawn around the pixmap. This option has only an effect if `pixmapRequested` is true. Default value: Enabled.

### 6.1457.1.20 setPixmapRequested()

```
void Digikam::ThumbnailLoadThread::setPixmapRequested (
    bool wantPixmap )
```

If you enable this, the signal `thumbnailLoaded(LoadingDescription, QPixmap)` will be emitted. If you do not enable this, only the QImage-based signal (see [LoadSaveThread](#)) will be emitted.

If you disable this, pay attention to the (global) setting of the [LoadingCache](#), which per default does not cache the images !!

Default value: Enabled.

### 6.1457.1.21 setSendSurrogatePixmap()

```
void Digikam::ThumbnailLoadThread::setSendSurrogatePixmap (
    bool send )
```

If you enable this, the thread will try hard to send a pixmap if thumbnail loading failed. It will use standard system icons to replace the real thumbnail. If you disable this, a null QPixmap will be sent. This does not influence the QImage-based signal; this signal will be emitted with a null QImage regardless of this setting here, if the loading failed. Default value: Enabled.

### 6.1457.1.22 setThumbnailSize()

```
void Digikam::ThumbnailLoadThread::setThumbnailSize (
    int size,
    bool forFace = false )
```

#### Note

If the thread is currently loading thumbnails, there is no guarantee as to when the property change by one of the following methods takes effect. Set the requested thumbnail size. Default value: 128

### 6.1457.1.23 storeDetailThumbnail()

```
void Digikam::ThumbnailLoadThread::storeDetailThumbnail (
    const QString & filePath,
    const QRect & detailRect,
    const QImage & image,
    bool isFace = false )
```

Stores the given detail thumbnail on disk. Use this if possible because generation of detail thumbnails is potentially slower. The image should at least have `storedSize()`.

### 6.1457.1.24 thumbnailLoaded()

```
void Digikam::ThumbnailLoadThread::thumbnailLoaded (
    const LoadingDescription & loadingDescription,
    const QImage & img ) [override], [protected], [virtual]
```

virtual method overridden from [LoadSaveNotifier](#), implemented first by [LoadSaveThread](#) called by ThumbnailTask from working thread

Reimplemented from [Digikam::LoadSaveThread](#).

### 6.1457.1.25 thumbnailToPixmapSize()

```
int Digikam::ThumbnailLoadThread::thumbnailToPixmapSize (
    int size ) const
```

Computes the pixmap size for the give thumbnail size. These can differ when highlighting is turned on.

## 6.1458 Digikam::ThumbnailLoadThread::Private Class Reference

### Public Member Functions

- bool **checkDescription** (const [LoadingDescription](#) &description)
- [LoadingDescription](#) **createLoadingDescription** (const [ThumbnailIdentifier](#) &identifier, int size, bool setLastDescription=true)
- [LoadingDescription](#) **createLoadingDescription** (const [ThumbnailIdentifier](#) &identifier, int size, const QRect &detailRect, bool setLastDescription=true)
- bool **hasHighlightingBorder** () const
- QList< [LoadingDescription](#) > **makeDescriptions** (const QList< QPair< [ThumbnailIdentifier](#), QRect > > &idsAndRects, int size)
- QList< [LoadingDescription](#) > **makeDescriptions** (const QList< [ThumbnailIdentifier](#) > &identifiers, int size)
- int **pixmapSizeForThumbnailSize** (int thumbnailSize) const
- int **thumbnailSizeForPixmapSize** (int pixmapSize) const

### Public Attributes

- QHash< QString, [ThumbnailResult](#) > **collectedResults**
- [ThumbnailCreator](#) \* **creator** = nullptr
- bool **highlight** = true
- QList< [LoadingDescription](#) > **lastDescriptions**
- bool **notifiedForResults** = false
- QMutex **resultsMutex**
- bool **sendSurrogate** = true
- int **size** = ThumbnailSize::maxThumbsSize()
- bool **wantPixmap** = true

## 6.1459 Digikam::ThumbnailLoadThreadStaticPriv Class Reference

### Public Member Functions

- [ThumbnailLoadThreadStaticPriv](#) & **operator=** (const [ThumbnailLoadThreadStaticPriv](#) &)=delete
- [ThumbnailLoadThreadStaticPriv](#) (const [ThumbnailLoadThreadStaticPriv](#) &)=delete

**Public Attributes**

- bool **firstThreadCreated** = false
- [IccProfile](#) **profile** = [IccProfile::sRGB\(\)](#)
- [ThumbnailInfoProvider](#) \* **provider** = nullptr
- [ThumbnailCreator::StorageMethod](#) **storageMethod** = [ThumbnailCreator::FreeDesktopStandard](#)

**6.1460 Digikam::ThumbnailResult Class Reference****Public Member Functions**

- **ThumbnailResult** (const [LoadingDescription](#) &description, const QImage &image)

**Public Attributes**

- QImage **image**
- [LoadingDescription](#) **loadingDescription**

**6.1461 Digikam::ThumbnailSize Class Reference****Public Types**

- enum [Size](#) {  
**Step** = 8 , **Tiny** = 32 , **VerySmall** = 64 , **MediumSmall** = 80 ,  
**Small** = 100 , **Medium** = 142 , **Large** = 160 , **Huge** = 256 ,  
**HD** = 512 , **MAX** = 1024 }

**Public Member Functions**

- bool **operator!=** (const [ThumbnailSize](#) &thumbsize) const
- [ThumbnailSize](#) & **operator=** (const [ThumbnailSize](#) &thumbsize)
- bool **operator==** (const [ThumbnailSize](#) &thumbsize) const
- int **size** () const
- [ThumbnailSize](#) (const [ThumbnailSize](#) &thumbsize)
- [ThumbnailSize](#) (int size)

**Static Public Member Functions**

- static bool **getUseLargeThumbs** ()
- static int **maxThumbsSize** ()
- static void **readSettings** (const KConfigGroup &group)
- static void **saveSettings** (KConfigGroup &group, bool val)
- static void **setUseLargeThumbs** (bool val)

**6.1461.1 Member Enumeration Documentation****6.1461.1.1 Size**

```
enum Digikam::ThumbnailSize::Size
```

## Enumerator

Small	Most usable small size of thumbnails to prevent overloaded overlays show under thumbs (as Pick label and Group indicator) See bugs #321337 and #275381 for details.
-------	---

## 6.1462 Digikam::ThumbsDb Class Reference

### Public Member Functions

- `QList< int > findAll ()`
- `ThumbsDbInfo findByCustomIdentifier (const QString &id)`
- `ThumbsDbInfo findByFilePath (const QString &path)`
- `ThumbsDbInfo findByFilePath (const QString &path, const QString &uniqueHash)`
- `ThumbsDbInfo findByHash (const QString &uniqueHash, qlonglong fileSize)`
- `QHash< QString, int > getFilePathsWithThumbnail ()`
- `QString getLegacySetting (const QString &keyword)`
- `QString getSetting (const QString &keyword)`
- `BdEngineBackend::QueryState insertCustomIdentifier (const QString &id, int thumbId)`
- `BdEngineBackend::QueryState insertFilePath (const QString &path, int thumbId)`
- `BdEngineBackend::QueryState insertThumbnail (const ThumbsDbInfo &info, QVariant *const lastInsertId=nullptr)`
- `BdEngineBackend::QueryState insertUniqueHash (const QString &uniqueHash, qlonglong fileSize, int thumbId)`
- `bool integrityCheck ()`
- `BdEngineBackend::QueryState remove (int thumbId)`
- `BdEngineBackend::QueryState removeByCustomIdentifier (const QString &id)`
- `BdEngineBackend::QueryState removeByFilePath (const QString &path)`
- `BdEngineBackend::QueryState removeByUniqueHash (const QString &uniqueHash, qlonglong fileSize)`
- `BdEngineBackend::QueryState renameByFilePath (const QString &oldPath, const QString &newPath)`
- `BdEngineBackend::QueryState replaceThumbnail (const ThumbsDbInfo &info)`
- `void replaceUniqueHash (const QString &oldUniqueHash, int oldFileSize, const QString &newUniqueHash, int newFileSize)`
- `bool setSetting (const QString &keyword, const QString &value)`
- `BdEngineBackend::QueryState updateModificationDate (int thumbId, const QDateTime &modificationDate)`
- `void vacuum ()`

### Friends

- class `ThumbsDbAccess`

## 6.1462.1 Member Function Documentation

### 6.1462.1.1 findAll()

```
QList< int > Digikam::ThumbsDb::findAll ( )
```

Returns the thumbnail ids of all thumbnails in the database.

### 6.1462.1.2 findByFilePath()

```
ThumbsDbInfo Digikam::ThumbsDb::findByFilePath (
    const QString & path,
    const QString & uniqueHash )
```

This is findByFilePath with extra security: Pass the uniqueHash which you have. If an entry is found by file path, and the entry is referenced by any uniqueHash, which is different from the given hash, a null info is returned. If uniqueHash is null, equivalent to the simple findByFilePath.

### 6.1462.1.3 integrityCheck()

```
bool Digikam::ThumbsDb::integrityCheck ( )
```

Returns true if the integrity of the database is preserved.

### 6.1462.1.4 removeByFilePath()

```
BdEngineBackend::QueryState Digikam::ThumbsDb::removeByFilePath (
    const QString & path )
```

Removes thumbnail data associated to the given file path

### 6.1462.1.5 removeByUniqueHash()

```
BdEngineBackend::QueryState Digikam::ThumbsDb::removeByUniqueHash (
    const QString & uniqueHash,
    qlonglong fileSize )
```

Removes thumbnail data associated to the given uniqueHash/fileSize

### 6.1462.1.6 vacuum()

```
void Digikam::ThumbsDb::vacuum ( )
```

Shrinks the database.

## 6.1463 Digikam::ThumbsDbAccess Class Reference

### Public Member Functions

- [ThumbsDbBackend](#) \* **backend** () const
- [ThumbsDb](#) \* **db** () const
- QString **lastError** () const
- void [setLastError](#) (const QString &error)
- [ThumbsDbAccess](#) ()



## Static Public Member Functions

- static bool **checkReadyForUse** ([InitializationObserver](#) \*const observer)
- static void **cleanUpDatabase** ()
- static void **initDbEngineErrorHandler** ([DbEngineErrorHandler](#) \*const errorhandler)
- static bool **isInitialized** ()
- static [DbEngineParameters](#) **parameters** ()
- static void **setParameters** (const [DbEngineParameters](#) &parameters)

## 6.1463.1 Constructor & Destructor Documentation

### 6.1463.1.1 ThumbsDbAccess()

```
Digikam::ThumbsDbAccess::ThumbsDbAccess ( )
```

This class is written in analogy to [CoreDbAccess](#) (some features stripped off). For documentation, see `coredbaccess.h`

## 6.1463.2 Member Function Documentation

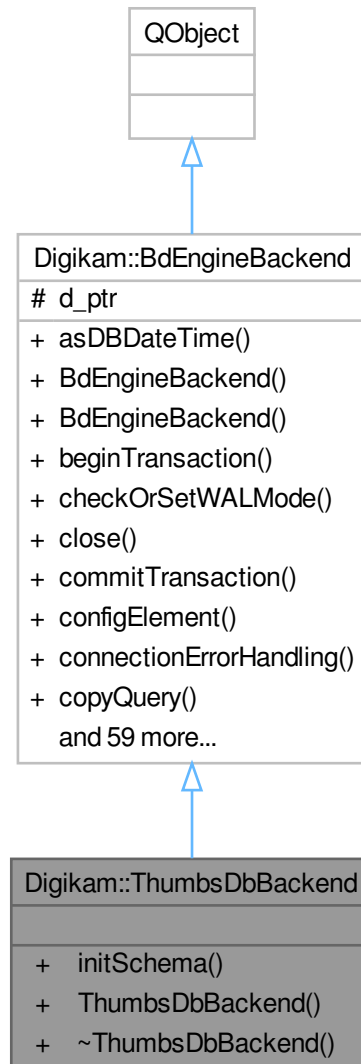
### 6.1463.2.1 setLastError()

```
void Digikam::ThumbsDbAccess::setLastError (
    const QString & error )
```

Set the "last error" message. This method is not for public use.

## 6.1464 Digikam::ThumbsDbBackend Class Reference

Inheritance diagram for Digikam::ThumbsDbBackend:



### Public Member Functions

- bool [initSchema](#) ([ThumbsDbSchemaUpdater](#) \*const updater)
- **ThumbsDbBackend** ([DbEngineLocking](#) \*const locking, const QString &backendName=QLatin1↔String("thumbnailDatabase-"))

### Public Member Functions inherited from [Digikam::BdEngineBackend](#)

- QDateTime [asDBDateTime](#) (const QDateTime &dateTime) const

- [BdEngineBackend](#) (const QString &backendName, [DbEngineLocking](#) \*const locking)
- **BdEngineBackend** (const QString &backendName, [DbEngineLocking](#) \*const locking, [BdEngineBackendPrivate](#) &dd)
- [BdEngineBackend::QueryState](#) [beginTransaction](#) ()
- bool [checkOrSetWALMode](#) ()
- void [close](#) ()
- [BdEngineBackend::QueryState](#) [commitTransaction](#) ()
- [DbEngineConfigSettings](#) [configElement](#) () const
- bool [connectionErrorHandling](#) (int retries)
- [DbEngineSqlQuery](#) [copyQuery](#) (const [DbEngineSqlQuery](#) &old)
- DbType [databaseType](#) () const
- bool [exec](#) ([DbEngineSqlQuery](#) &query)
- bool **execBatch** ([DbEngineSqlQuery](#) &query)
- [QueryState](#) [execDBAction](#) (const [DbEngineAction](#) &action, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) [execDBAction](#) (const [DbEngineAction](#) &action, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execDBAction** (const QString &action, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execDBAction** (const QString &action, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- QSqlQuery [execDBActionQuery](#) (const [DbEngineAction](#) &action, const QMap< QString, QVariant > &bindingMap)
- QSqlQuery **execDBActionQuery** (const QString &action, const QMap< QString, QVariant > &bindingMap)
- [QueryState](#) [execDirectSql](#) (const QString &query)
- [QueryState](#) [execDirectSqlWithResult](#) (const QString &query, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [DbEngineSqlQuery](#) [execQuery](#) (const QString &sql)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QList< QVariant > &boundValues)
- [DbEngineSqlQuery](#) [execQuery](#) (const QString &sql, const QMap< QString, QVariant > &bindingMap)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- [DbEngineSqlQuery](#) **execQuery** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QList< QVariant > &boundValues)
- void [execQuery](#) ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3)
- void **execQuery** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4)
- [QueryState](#) **execSql** (const QString &sql, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) [execSql](#) (const QString &sql, const QMap< QString, QVariant > &bindingMap, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)

- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** (const QString &sql, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QList< QVariant > &boundValues, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, const QVariant &boundValue4, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, const QVariant &boundValue3, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, const QVariant &boundValue2, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execSql** ([DbEngineSqlQuery](#) &preparedQuery, const QVariant &boundValue1, QList< QVariant > \*const values=nullptr, QVariant \*const lastInsertId=nullptr)
- [QueryState](#) **execUpsertDBAction** (const [DbEngineAction](#) &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [QueryState](#) **execUpsertDBAction** (const QString &action, const QVariant &id, const QStringList &fieldNames, const QList< QVariant > &values)
- [DbEngineAction](#) **getDBAction** (const QString &actionName) const
- [DbEngineSqlQuery](#) **getQuery** ()
- [QueryState](#) **handleQueryResult** ([DbEngineSqlQuery](#) &query, QList< QVariant > \*const values, QVariant \*const lastInsertId)
- bool **isCompatible** (const [DbEngineParameters](#) &parameters)
- bool **isInTransaction** () const
- bool **isOpen** () const
- bool **isReady** () const
- QString **lastError** ()
- QSqlError **lastSQLError** ()
- int **maximumBoundValues** () const
- bool **open** (const [DbEngineParameters](#) &parameters)
- [DbEngineSqlQuery](#) **prepareQuery** (const QString &sql)
- bool **queryErrorHandling** ([DbEngineSqlQuery](#) &query, int retries)
- QList< QVariant > **readToList** ([DbEngineSqlQuery](#) &query)
- void **rollbackTransaction** ()
- void **setDbEngineErrorHandler** ([DbEngineErrorHandler](#) \*const handler)
- void **setForeignKeyChecks** (bool check)
- [Status](#) **status** () const
- QStringList **tables** ()
- bool **transactionErrorHandling** (const QSqlError &lastError, int retries)

### Additional Inherited Members

### Public Types inherited from [Digikam::BdEngineBackend](#)

- enum **DbType** { [SQLite](#) , [MySQL](#) }
- enum **QueryOperationStatus** { [ExecuteNormal](#) , [Wait](#) , [AbortQueries](#) }
- enum **QueryStateEnum** { [NoErrors](#) , [SQLError](#) , [ConnectionError](#) }
- enum **Status** { [Unavailable](#) , [Open](#) , [OpenSchemaChecked](#) }

## Protected Attributes inherited from [Digikam::BdEngineBackend](#)

- [BdEngineBackendPrivate](#) \*const **d\_ptr** = nullptr

### 6.1464.1 Member Function Documentation

#### 6.1464.1.1 `initSchema()`

```
bool Digikam::ThumbsDbBackend::initSchema (
    ThumbsDbSchemaUpdater *const updater )
```

Initialize the database schema to the current version, carry out upgrades if necessary. Shall only be called from the thread that called `open()`.

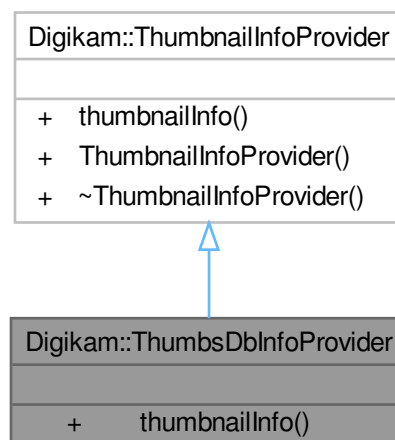
## 6.1465 Digikam::ThumbsDbInfo Class Reference

### Public Attributes

- QByteArray **data**
- int **id** = -1
- QDateTime **modificationDate**
- int **orientationHint** = 0
- DatabaseThumbnail::Type **type** = DatabaseThumbnail::UndefinedType

## 6.1466 Digikam::ThumbsDbInfoProvider Class Reference

Inheritance diagram for Digikam::ThumbsDbInfoProvider:



## Public Member Functions

- [ThumbnailInfo thumbnailInfo](#) (const [ThumbnailIdentifier](#) &identifier) override

## 6.1466.1 Member Function Documentation

### 6.1466.1.1 thumbnailInfo()

```
ThumbnailInfo Digikam::ThumbsDbInfoProvider::thumbnailInfo (  
    const ThumbnailIdentifier & identifier ) [override], [virtual]
```

Implements [Digikam::ThumbnailInfoProvider](#).

## 6.1467 Digikam::ThumbsDbSchemaUpdater Class Reference

### Public Member Functions

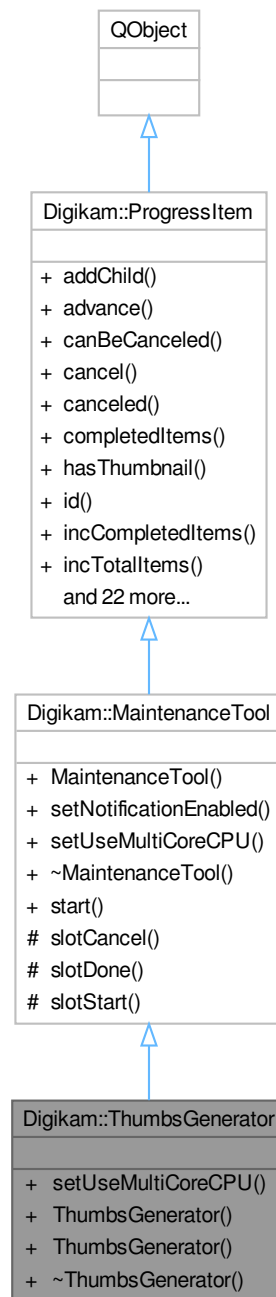
- void **setObserver** ([InitializationObserver](#) \*const observer)
- **ThumbsDbSchemaUpdater** ([ThumbsDbAccess](#) \*const dbAccess)
- bool **update** ()

### Static Public Member Functions

- static int **schemaVersion** ()

## 6.1468 Digikam::ThumbsGenerator Class Reference

Inheritance diagram for Digikam::ThumbsGenerator:



### Public Member Functions

- void [setUseMultiCoreCPU](#) (bool b) override
- [ThumbsGenerator](#) (const bool rebuildAll, const AlbumList &list, [ProgressItem](#) \*const parent=nullptr)
- [ThumbsGenerator](#) (const bool rebuildAll, int albumId, [ProgressItem](#) \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::MaintenanceTool](#)

- **MaintenanceTool** (const QString &id, [ProgressItem](#) \*const parent=nullptr)
- void [setNotificationEnabled](#) (bool b)

## Public Member Functions inherited from [Digikam::ProgressItem](#)

- void **addChild** ([ProgressItem](#) \*const kiddo)
- bool [advance](#) (unsigned int v)
 

*Advance total items processed by n values and update percentage in progressbar.*
- bool [canBeCanceled](#) () const
- void **cancel** ()
- bool **canceled** () const
- unsigned int **completedItems** () const
- bool [hasThumbnail](#) () const
- const QString & [id](#) () const
- bool **incCompletedItems** (unsigned int v=1)
- void **incTotalItems** (unsigned int v=1)
- const QString & [label](#) () const
- [ProgressItem](#) \* [parent](#) () const
- unsigned int [progress](#) () const
- **ProgressItem** ([ProgressItem](#) \*const parent, const QString &id, const QString &label, const QString &status, bool [canBeCanceled](#), bool hasThumb)
- void **removeChild** ([ProgressItem](#) \*const kiddo)
- void **reset** ()
 

*Reset the progress value of this item to 0 and the status string to the empty string.*
- void **setComplete** ()
 

*Tell the item it has finished. This will emit [progressItemCompleted\(\)](#) result in the destruction of the item after all slots connected to this signal have executed. This is the only way to get rid of an item and needs to be called even if the item is canceled. Don't use the item after this has been called on it.*
- bool **setCompletedItems** (unsigned int v)
- void [setLabel](#) (const QString &v)
- void [setProgress](#) (unsigned int v)
 

*Set the progress (percentage of completion) value of this item.*
- void [setShowAtStart](#) (bool [showAtStart](#))
 

*Set the property to pop-up item when it's added in progress manager. Use this method if you consider that item is important to be notified to end-user.*
- void [setStatus](#) (const QString &v)
 

*Set the string to be used for showing this item's current status.*
- void [setThumbnail](#) (const QIcon &icon)
 

*Sets whether this item has a thumbnail.*
- void **setTotalItems** (unsigned int v)
- void [setUsesBusyIndicator](#) (bool useBusyIndicator)
 

*Sets whether this item uses a busy indicator instead of real progress for its progress bar. If it uses a busy indicator, you are still responsible for calling [setProgress\(\)](#) from time to time to update the busy indicator.*
- bool [showAtStart](#) () const
- const QString & [status](#) () const
- bool **totalCompleted** () const
- unsigned int **totalItems** () const
- void **updateProgress** ()
 

*Recalculate progress according to total/completed items and update.*
- bool [usesBusyIndicator](#) () const



## Additional Inherited Members

### Public Slots inherited from [Digikam::MaintenanceTool](#)

- void **start** ()

### Signals inherited from [Digikam::MaintenanceTool](#)

- void **signalCanceled** ()
- void **signalComplete** ()

### Signals inherited from [Digikam::ProgressItem](#)

- void **progressItemAdded** ([ProgressItem](#) \*item)  
*Emitted when a new [ProgressItem](#) is added.*
- void **progressItemCanceled** ([ProgressItem](#) \*item)  
*Emitted when an item was canceled. It will not go away immediately, only when the owner sets it complete, which will usually happen. Can be used to visually indicate the canceled status of an item. Should be used by the owner of the item to make sure it is set completed even if it is canceled. There is a [ProgressManager::slotStandardCancelHandler](#) which simply sets the item completed and can be used if no other work needs to be done on cancel.*
- void **progressItemCanceledById** (const [QString](#) &id)
- void **progressItemCompleted** ([ProgressItem](#) \*item)  
*Emitted when a progress item was completed. The item will be deleted afterwards, so slots connected to this are the last chance to work with this item.*
- void **progressItemLabel** ([ProgressItem](#) \*item, const [QString](#) &label)  
*Emitted when the label of an item changed. Should be used by progress dialogs to update the label of an item.*
- void **progressItemProgress** ([ProgressItem](#) \*item, unsigned int v)  
*Emitted when the progress value of an item changes.*
- void **progressItemStatus** ([ProgressItem](#) \*item, const [QString](#) &mess)  
*Emitted when the status message of an item changed. Should be used by progress dialogs to update the status message for an item.*
- void **progressItemThumbnail** ([ProgressItem](#) \*item, const [QPixmap](#) &thumb)  
*Emitted when the thumbnail data must be set in item.*
- void **progressItemUsesBusyIndicator** ([ProgressItem](#) \*item, bool value)  
*Emitted when the busy indicator state of an item changes. Should be used by progress dialogs so that they can adjust the display of the progress bar to the new mode.*

### Protected Slots inherited from [Digikam::MaintenanceTool](#)

- virtual void **slotCancel** ()
- virtual void **slotDone** ()
- virtual void **slotStart** ()

## 6.1468.1 Constructor & Destructor Documentation

### 6.1468.1.1 ThumbsGenerator() [1/2]

```
Digikam::ThumbsGenerator::ThumbsGenerator (
    const bool rebuildAll,
    int albumId,
    ProgressItem *const parent = nullptr ) [explicit]
```

Constructor using [Album](#) Id as argument. If Id = -1, whole Albums collection is processed.

### 6.1468.1.2 ThumbsGenerator() [2/2]

```
Digikam::ThumbsGenerator::ThumbsGenerator (
    const bool rebuildAll,
    const AlbumList & list,
    ProgressItem *const parent = nullptr )
```

Constructor using AlbumList as argument. If list is empty, whole Albums collection is processed.

## 6.1468.2 Member Function Documentation

### 6.1468.2.1 setUseMultiCoreCPU()

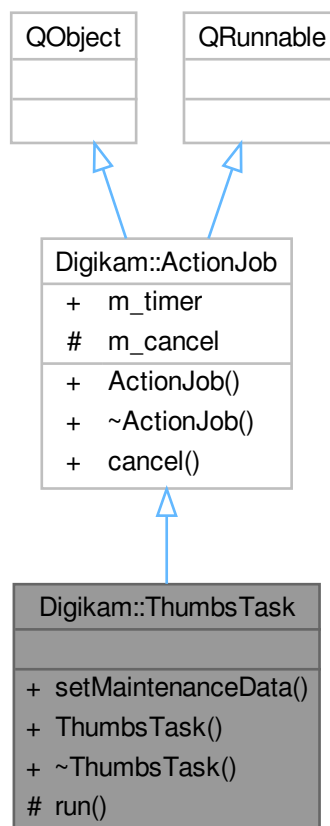
```
void Digikam::ThumbsGenerator::setUseMultiCoreCPU (
    bool ) [override], [virtual]
```

Re-implement this method if your tool is able to use multi-core CPU to process item in parallel

Reimplemented from [Digikam::MaintenanceTool](#).

## 6.1469 Digikam::ThumbsTask Class Reference

Inheritance diagram for Digikam::ThumbsTask:



## Signals

- void **signalFinished** (const [ItemInfo](#) &, const QImage &)

## Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

## Public Member Functions

- void **setMaintenanceData** ([MaintenanceData](#) \*const data=nullptr)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void **run** () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

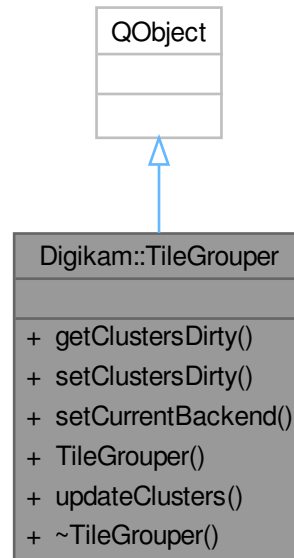
- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.1470 Digikam::TileGrouper Class Reference

Inheritance diagram for Digikam::TileGrouper:



### Public Member Functions

- bool `getClustersDirty` () const
- void `setClustersDirty` ()
- void `setCurrentBackend` ([MapBackend](#) \*const backend)
- `TileGrouper` (const [QExplicitlySharedDataPointer](#)< [GeofaceSharedData](#) > &sharedData, [QObject](#) \*const parent)
- void `updateClusters` ()

### 6.1470.1 Member Function Documentation

#### 6.1470.1.1 updateClusters()

```
void Digikam::TileGrouper::updateClusters ( )
```

## 6.1471 Digikam::TileIndex Class Reference

### Public Types

- enum `Constants` { `MaxLevel` = 9 , `MaxIndexCount` = `MaxLevel`+1 , `Tiling` = 10 , `MaxLinearIndex` = `Tiling`\*`↔`  
`Tiling` }
- enum `CornerPosition` { `CornerNW` = 1 , `CornerSW` = 2 , `CornerNE` = 3 , `CornerSE` = 4 }
- typedef `QList`< [TileIndex](#) > `List`

### Public Member Functions

- void **appendLatLonIndex** (const int latIndex, const int lonIndex)
- void **appendLinearIndex** (const int newIndex)
- int **at** (const int getLevel) const
- void **clear** ()
- int **indexCount** () const
- int **indexLat** (const int getLevel) const
- int **indexLon** (const int getLevel) const
- int **lastIndex** () const
- QPoint **latLonIndex** (const int getLevel) const
- void **latLonIndex** (const int getLevel, int \*const latIndex, int \*const lonIndex) const
- int **level** () const
- int **linearIndex** (const int getLevel) const
- [TileIndex](#) **mid** (const int first, const int len) const
- void **oneUp** ()
- [GeoCoordinates](#) **toCoordinates** () const
- [GeoCoordinates](#) **toCoordinates** (const CornerPosition ofCorner) const
- QList< [TileIndex](#) > **toIntList** () const

### Static Public Member Functions

- static [TileIndex](#) **fromCoordinates** (const [Digikam::GeoCoordinates](#) &coordinate, const int getLevel)
- static [TileIndex](#) **fromIntList** (const QList< [TileIndex](#) > &intList)
- static bool **indicesEqual** (const [TileIndex](#) &a, const [TileIndex](#) &b, const int upToLevel)
- static QList< QList< [TileIndex](#) > > **listToIntListList** (const QList< [TileIndex](#) > &tileIndexList)

## 6.1472 Digikam::TimeAdjustContainer Class Reference

### Public Types

- enum **AdjType** { **COPYVALUE** = 0 , **ADDVALUE** , **SUBVALUE** , **INTERVAL** }
- enum **UseDataSource** { **APPDATE** = 0 , **FILENAME** , **FILEDATE** , **METADATADATE** , **CUSTOMDATE** }
- enum **UseFileType** { **FILELASTMOD** = 0 , **FILECREATED** }
- enum **UseMetaDataType** { **EXIFPTCXMP** = 0 , **EXIFCREATED** , **EXIFORIGINAL** , **EXIFDIGITIZED** , **IPTCCREATED** , **XMPCREATED** , **FUZZYCREATED** , **FUZZYORIGINAL** , **FUZZYDIGITIZED** }

### Public Member Functions

- bool **atLeastOneUpdateToProcess** () const
- QDateTime **calculateAdjustedDate** (const QDateTime &originalTime, int index=0)
- QDateTime **getDateFromTimeString** (const QString &dateStr) const
- QMap< QString, bool > **getDateTagsMap** () const

## Public Attributes

- int **adjustmentDays** = 0
- QDateTime **adjustmentTime** = QDateTime()
- int **adjustmentType** = COPYVALUE
- QDateTime **customDate** = QDateTime::currentDateTime()
- QDateTime **customTime** = QDateTime::currentDateTime()
- int **dateSource** = APPDATE
- bool **enableExifTool** = false
- int **fileDateSource** = FILELASTMOD
- int **metadataSource** = EXIFIPTCXMP
- bool **updEXIFDigDate** = false
- bool **updEXIFModDate** = false
- bool **updEXIFOriDate** = false
- bool **updEXIFThmDate** = false
- bool **updFileModDate** = false
- bool **updIfAvailable** = true
- bool **updIPTCDate** = false
- bool **updUseExifTool** = false
- bool **updXMPDate** = false
- bool **updXMPVideo** = false

### 6.1472.1 Detailed Description

Container that store all timestamp adjustments.

### 6.1472.2 Member Function Documentation

#### 6.1472.2.1 atLeastOneUpdateToProcess()

```
bool Digikam::TimeAdjustContainer::atLeastOneUpdateToProcess ( ) const
```

Check if at least one option is selected

### 6.1472.3 Member Data Documentation

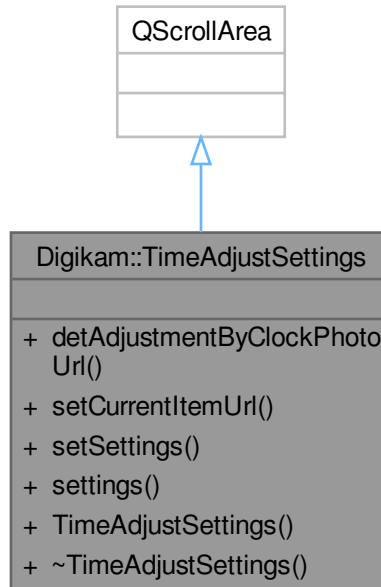
#### 6.1472.3.1 enableExifTool

```
bool Digikam::TimeAdjustContainer::enableExifTool = false
```

Only a temporary variable, will not be saved

## 6.1473 Digikam::TimeAdjustSettings Class Reference

Inheritance diagram for Digikam::TimeAdjustSettings:



### Signals

- void **signalSettingsChanged** ()
- void **signalSettingsChangedTool** ()
- void **signalSrcTimestampChanged** ()

### Public Member Functions

- void [detAdjustmentByClockPhotoUrl](#) (const `QUrl` &url)
- void **setCurrentItemUrl** (const `QUrl` &url)
- void **setSettings** (const [TimeAdjustContainer](#) &settings)
- [TimeAdjustContainer](#) **settings** () const
- **TimeAdjustSettings** (`QWidget` \*const parent, bool timeAdjustTool=false)

### 6.1473.1 Member Function Documentation

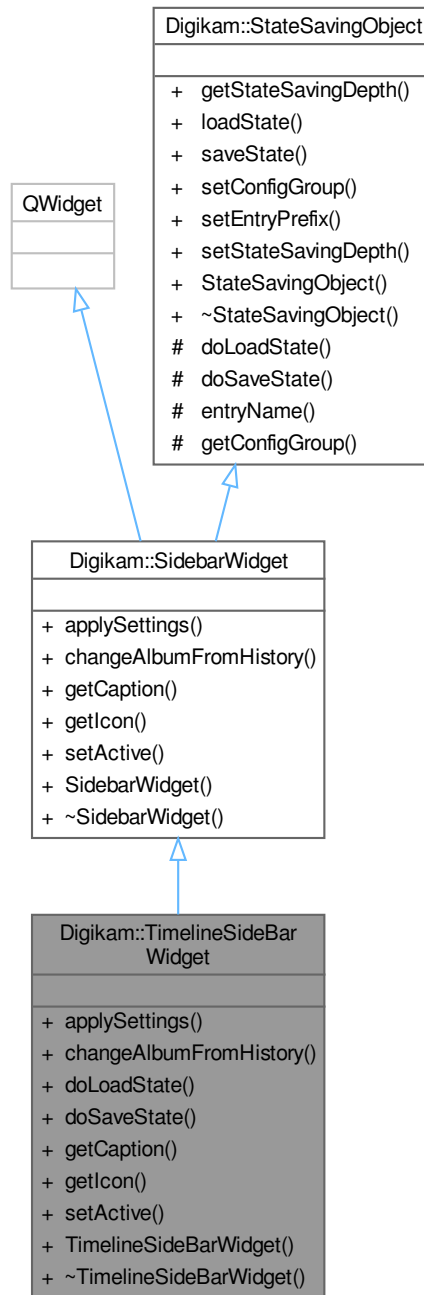
#### 6.1473.1.1 detAdjustmentByClockPhotoUrl()

```
void Digikam::TimeAdjustSettings::detAdjustmentByClockPhotoUrl (
    const QUrl & url )
```

When user press the clock photo button, a dialog is displayed and set the results to the proper widgets.

## 6.1474 Digikam::TimelineSideBarWidget Class Reference

Inheritance diagram for Digikam::TimelineSideBarWidget:



### Public Member Functions

- void `applySettings()` override
- void `changeAlbumFromHistory(const QList< Album * > &album)` override



- void [doLoadState](#) () override
- void [doSaveState](#) () override
- const QString [getCaption](#) () override
- const QIcon [getIcon](#) () override
- void [setActive](#) (bool active) override
- **TimelineSideBarWidget** (QWidget \*const parent, [searchModel](#) \*const searchModel, [searchModificationHelper](#) \*const searchModificationHelper)

### Public Member Functions inherited from [Digikam::SidebarWidget](#)

- [SidebarWidget](#) (QWidget \*const parent)
- [~SidebarWidget](#) () override=default

### Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) [getStateSavingDepth](#) () const
- void [loadState](#) ()
- void [saveState](#) ()
- virtual void [setConfigGroup](#) (const KConfigGroup &group)
- virtual void [setEntryPrefix](#) (const QString &prefix)
- void [setStateSavingDepth](#) (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

### Additional Inherited Members

### Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

### Signals inherited from [Digikam::SidebarWidget](#)

- void [requestActiveTab](#) ([SidebarWidget](#) \*)
- void [signalNotificationError](#) (const QString &message, int type)

### Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString [entryName](#) (const QString &base) const
- KConfigGroup [getConfigGroup](#) () const

## 6.1474.1 Member Function Documentation

### 6.1474.1.1 [applySettings\(\)](#)

```
void Digikam::TimelineSideBarWidget::applySettings ( ) [override], [virtual]
```

This method is invoked when the application settings should be (re-) applied to this widget.

Implements [Digikam::SidebarWidget](#).

### 6.1474.1.2 `changeAlbumFromHistory()`

```
void Digikam::TimelineSideBarWidget::changeAlbumFromHistory (
    const QList< Album * > & album ) [override], [virtual]
```

This is called on this widget when the history requires to move back to the specified album

Implements [Digikam::SidebarWidget](#).

### 6.1474.1.3 `doLoadState()`

```
void Digikam::TimelineSideBarWidget::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1474.1.4 `doSaveState()`

```
void Digikam::TimelineSideBarWidget::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1474.1.5 `getCaption()`

```
const QString Digikam::TimelineSideBarWidget::getCaption ( ) [override], [virtual]
```

Must be implemented to return the title of this sidebar's tab.

#### Returns

localized title string

Implements [Digikam::SidebarWidget](#).

### 6.1474.1.6 `getIcon()`

```
const QIcon Digikam::TimelineSideBarWidget::getIcon ( ) [override], [virtual]
```

Must be implemented and return the icon that shall be visible for this sidebar widget.

#### Returns

pixmap icon

Implements [Digikam::SidebarWidget](#).

### 6.1474.1.7 `setActive()`

```
void Digikam::TimelineSideBarWidget::setActive (
    bool active ) [override], [virtual]
```

This method is called if the visible sidebar widget is changed.

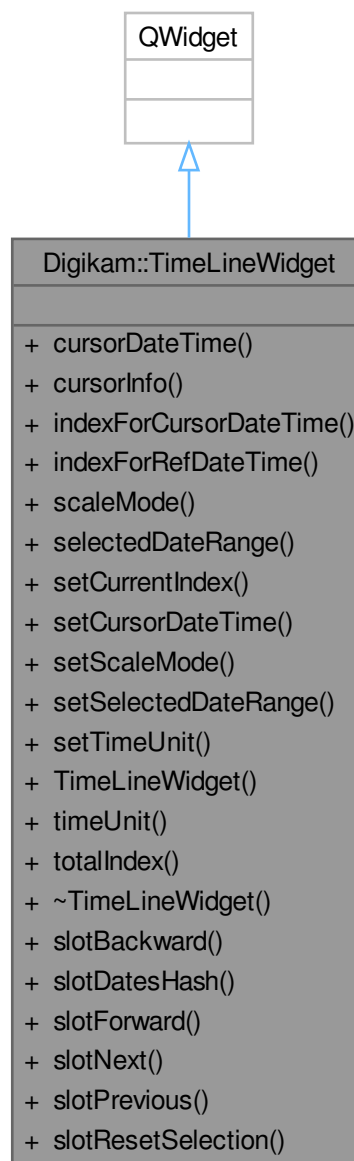
## Parameters

<i>active</i>	if true, this widget is the new active widget, if false another widget is active
---------------	--

Implements [Digikam::SidebarWidget](#).

## 6.1475 Digikam::TimeLineWidget Class Reference

Inheritance diagram for Digikam::TimeLineWidget:



## Public Types

- enum [ScaleMode](#) { [LinScale](#) = 0 , [LogScale](#) }
- enum [SelectionMode](#) { [Unselected](#) = 0 , [FuzzySelection](#) , [Selected](#) }
- enum [TimeUnit](#) { [Day](#) = 0 , [Week](#) , [Month](#) , [Year](#) }

## Public Slots

- void [slotBackward](#) ()
- void [slotDatesHash](#) (const QHash< QDateTime, int > &)
- void [slotForward](#) ()
- void [slotNext](#) ()
- void [slotPrevious](#) ()
- void [slotResetSelection](#) ()

## Signals

- void [signalCursorPositionChanged](#) ()
- void [signalDateMapChanged](#) ()
- void [signalRefDateTimeChanged](#) ()
- void [signalSelectionChanged](#) ()

## Public Member Functions

- QDateTime [cursorDateTime](#) () const
- int [cursorInfo](#) (QString &infoDate) const
- int [indexForCursorDateTime](#) () const
- int [indexForRefDateTime](#) () const
- [ScaleMode](#) [scaleMode](#) () const
- [DateRangeList](#) [selectedDateRange](#) (int &totalCount) const
- void [setCurrentIndex](#) (int index)
- void [setCursorDateTime](#) (const QDateTime &dateTime)
- void [setScaleMode](#) ([ScaleMode](#) scaleMode)
- void [setSelectedDateRange](#) (const [DateRangeList](#) &list)
- void [setTimeUnit](#) (TimeUnit timeUnit)
- [TimeLineWidget](#) (QWidget \*const parent=nullptr)
- TimeUnit [timeUnit](#) () const
- int [totalIndex](#) () const

## 6.1475.1 Member Enumeration Documentation

### 6.1475.1.1 ScaleMode

enum [Digikam::TimeLineWidget::ScaleMode](#)

#### Enumerator

<a href="#">LinScale</a>	Linear scale.
<a href="#">LogScale</a>	Logarithmic scale.

### 6.1475.1.2 SelectionMode

```
enum Digikam::TimeLineWidget::SelectionMode
```

#### Enumerator

Unselected	No selection.
FuzzySelection	Partially selected.
Selected	Fully selected.

## 6.1475.2 Member Function Documentation

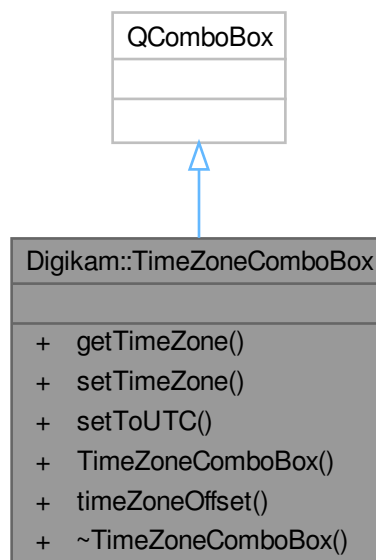
### 6.1475.2.1 selectedDateRange()

```
DateRangeList Digikam::TimeLineWidget::selectedDateRange (
    int & totalCount ) const
```

Return a list of Date-Range based on selection performed on days-map

## 6.1476 Digikam::TimeZoneComboBox Class Reference

Inheritance diagram for Digikam::TimeZoneComboBox:



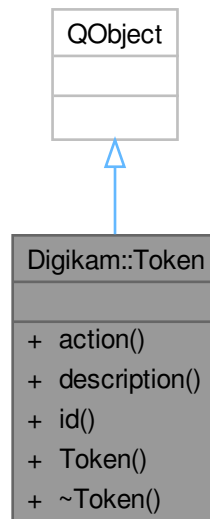
### Public Member Functions

- QString **getTimeZone** () const
- void **setTimeZone** (const QString &timeStr)
- void **setToUTC** ()
- **TimeZoneComboBox** (QWidget \*const parent)
- int **timeZoneOffset** () const

## 6.1477 Digikam::Token Class Reference

Token is the smallest parsing unit in AdvancedRename utility

Inheritance diagram for Digikam::Token:



### Signals

- void [signalTokenTriggered](#) (const QString &)

### Public Member Functions

- QAction \* [action](#) () const
- QString [description](#) () const
- QString [id](#) () const
- **Token** (const QString &[id](#), const QString &[description](#))

## 6.1477.1 Detailed Description

The Token class represents the smallest parsing unit for the [Parser](#) class. Every string you enter as a renaming pattern is a combination of tokens and literal text. For example  
`"[file]{upper}_###_abc.[ext]{lower}"`

is composed of five tokens

```
[file]
{upper}
###
.[ext]
{lower}
```

and two literals

```
_
_abc
```

A rule must assign at least one token object, to make parsing work. More than one token can be assigned to a Rule.

See also

[Rule::addToken\(\)](#)

## 6.1477.2 Member Function Documentation

### 6.1477.2.1 action()

```
QAction * Digikam::Token::action ( ) const
```

#### Returns

The action of the token. This action can be connected to a button or menu item. If triggered, high-level classes like [AdvancedRenameWidget](#) can connect to the signal and display the emitted text in the line edit input widget.

### 6.1477.2.2 description()

```
QString Digikam::Token::description ( ) const
```

#### Returns

The description of the token. It can be used for example in the tooltip of the [AdvancedRenameWidget](#).

### 6.1477.2.3 id()

```
QString Digikam::Token::id ( ) const
```

#### Returns

The ID of the token. This is the actual token string, for example  
`"[file]"`

This id will be emitted as a signal by `slotTriggered()`.

#### 6.1477.2.4 signalTokenTriggered

```
void Digikam::Token::signalTokenTriggered (
    const QString & ) [signal]
```

This signal is emitted when the action of the token is triggered.

## 6.1478 Digikam::TonalityContainer Class Reference

### Public Attributes

- int **blueMask** = 0
- int **greenMask** = 0
- int **redMask** = 0



## 6.1479 Digikam::TonalityFilter Class Reference

Inheritance diagram for Digikam::TonalityFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- void `readParameters` (const `FilterAction` &action) override
- `TonalityFilter` (`DImg` \*const orgImage, `QObject` \*const parent=nullptr, const `TonalityContainer` &settings=`TonalityContainer`())
- `TonalityFilter` (`QObject` \*const parent=nullptr)

### Public Member Functions inherited from `Digikam::DImgThreadedFilter`

- virtual void `cancelFilter` ()
- `DImgThreadedFilter` (`DImg` \*const orgImage, `QObject` \*const parent, const `QString` &name=`QString`())
- `DImgThreadedFilter` (`QObject` \*const parent=nullptr, const `QString` &name=`QString`())
- const `QString` & `filterName` ()
- int `filterVersion` () const
- `DImg` `getTargetImage` ()
- `QList`< int > `multithreadedSteps` (int stop, int start=0) const
- virtual bool `parametersSuccessfullyRead` () const
- virtual `QString` `readParametersError` (const `FilterAction` &actionThatFailed) const
- void `setFilterName` (const `QString` &name)
- void `setFilterVersion` (int version)
- void `setOriginalImage` (const `DImg` &orgImage)
- void `setupAndStartDirectly` (const `DImg` &orgImage, `DImgThreadedFilter` \*const master, int progressBegin=0, int progressEnd=100)
- void `setupFilter` (const `DImg` &orgImage)
- virtual void `startFilter` ()
- virtual void `startFilterDirectly` ()
- virtual `QList`< int > `supportedVersions` () const

### Public Member Functions inherited from `Digikam::DynamicThread`

- `DynamicThread` (`QObject` \*const parent=nullptr)
- bool `isFinished` () const
- bool `isRunning` () const
- `QThread::Priority` `priority` () const
- void `setEmitSignals` (bool emitThem)
- void `setPriority` (`QThread::Priority` priority)
- State `state` () const
- `~DynamicThread` () override

### Static Public Member Functions

- static int `CurrentVersion` ()
- static `QString` `DisplayableName` ()
- static `QString` `FilterIdentifier` ()
- static `QList`< int > `SupportedVersions` ()

### Additional Inherited Members

### Public Types inherited from `Digikam::DynamicThread`

- enum `State` { `Inactive` , `Scheduled` , `Running` , `Deactivating` }

## Public Slots inherited from Digikam::DynamicThread

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from Digikam::DImgThreadedFilter

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from Digikam::DynamicThread

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from Digikam::DImgThreadedFilter

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from Digikam::DynamicThread

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from Digikam::DImgThreadedFilter

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int m\_progressCurrent = 0
  - To prevent signals bombarding with progress indicator value in `postProgress()`.*
- int m\_progressSpan = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int m\_version = 1
- bool m\_wasCancelled = false

## 6.1479.1 Member Function Documentation

### 6.1479.1.1 filterAction()

`FilterAction` Digikam::TonalityFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1479.1.2 filterIdentifier()

`QString` Digikam::TonalityFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

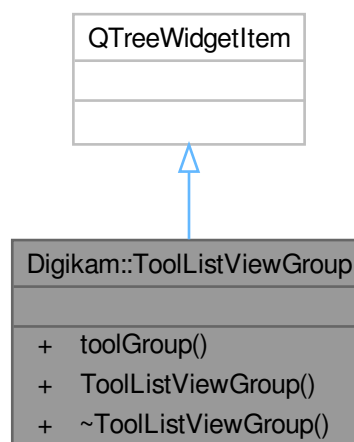
### 6.1479.1.3 readParameters()

```
void Digikam::TonalityFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1480 Digikam::ToolListViewGroup Class Reference

Inheritance diagram for Digikam::ToolListViewGroup:

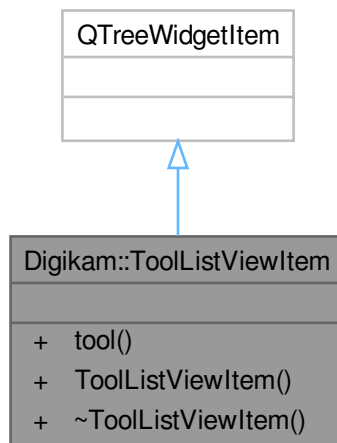


### Public Member Functions

- [BatchTool::BatchToolGroup](#) **toolGroup** () const
- **ToolListViewGroup** (QTreeWidget \*const parent, [BatchTool::BatchToolGroup](#) group)

## 6.1481 Digikam::ToolListViewItem Class Reference

Inheritance diagram for Digikam::ToolListViewItem:

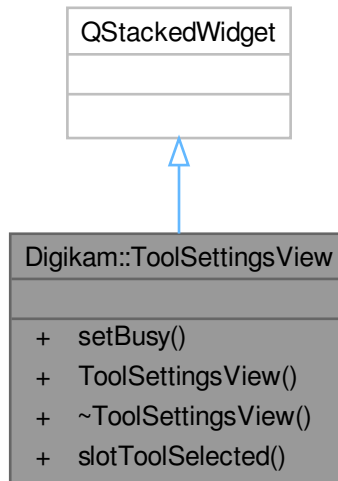


### Public Member Functions

- [BatchTool](#) \* **tool** () const
- **ToolListViewItem** ([ToolListViewGroup](#) \*const parent, [BatchTool](#) \*const tool)

## 6.1482 Digikam::ToolSettingsView Class Reference

Inheritance diagram for Digikam::ToolSettingsView:



### Public Slots

- void **slotToolSelected** (const [BatchToolSet](#) &)

### Signals

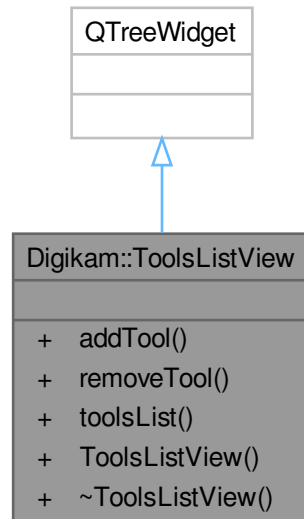
- void **signalSettingsChanged** (const [BatchToolSet](#) &)

### Public Member Functions

- void **setBusy** (bool b)
- **ToolSettingsView** (QWidget \*const parent=nullptr)

## 6.1483 Digikam::ToolsListView Class Reference

Inheritance diagram for Digikam::ToolsListView:



### Signals

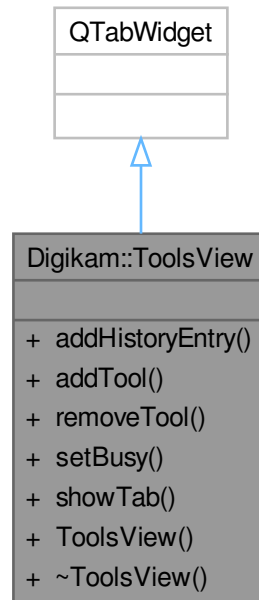
- void **signalAssignTools** (const QMap< int, QString > &)

### Public Member Functions

- void **addTool** ([BatchTool](#) \*const tool)
- bool **removeTool** ([BatchTool](#) \*const tool)
- [BatchToolsList](#) **toolsList** ()
- **ToolsListView** (QWidget \*const parent)

## 6.1484 Digikam::ToolsView Class Reference

Inheritance diagram for Digikam::ToolsView:



### Public Types

- enum `ViewTabs` { `TOOLS = 0` , `WORKFLOW` , `HISTORY` }

### Signals

- void `signalAssignQueueSettings` (QString)
- void `signalAssignTools` (const QMap< int, QString > &)
- void `signalHistoryEntryClicked` (int, qlonglong)
- void `signalUpdateQueueSettings` (QString)

### Public Member Functions

- void `addHistoryEntry` (const QString &msg, DHistoryView::EntryType type, int queueId=-1, qlonglong itemId=-1)
- void `addTool` ([BatchTool](#) \*const tool)
- bool `removeTool` ([BatchTool](#) \*const tool)
- void `setBusy` (bool b)
- void `showTab` (ViewTabs t)
- `ToolsView` (QWidget \*const parent=nullptr)



## 6.1485 Digikam::TooltipCreator Class Reference

### Public Member Functions

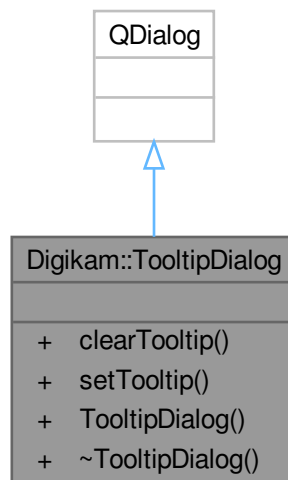
- QString **tooltip** (const Parser \*const parser)

### Static Public Member Functions

- static TooltipCreator & **getInstance** ()

## 6.1486 Digikam::TooltipDialog Class Reference

Inheritance diagram for Digikam::TooltipDialog:

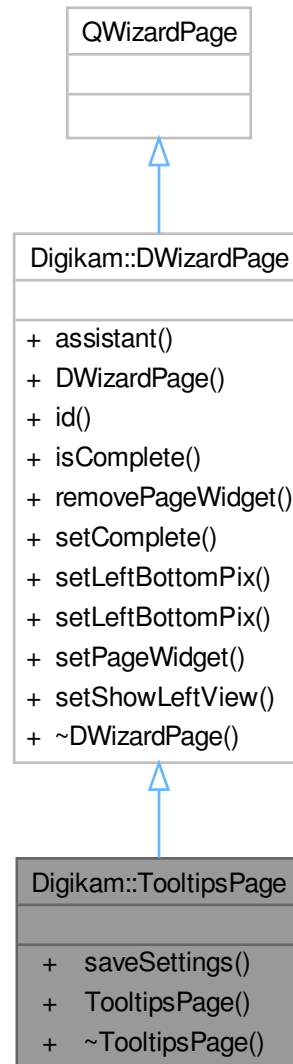


### Public Member Functions

- void **clearTooltip** ()
- void **setTooltip** (const QString &tooltip)
- **TooltipDialog** (QWidget \*const parent)

## 6.1487 Digikam::TooltipsPage Class Reference

Inheritance diagram for Digikam::TooltipsPage:



### Public Member Functions

- void **saveSettings** ()
- **TooltipsPage** (QWizard \*const dlg)

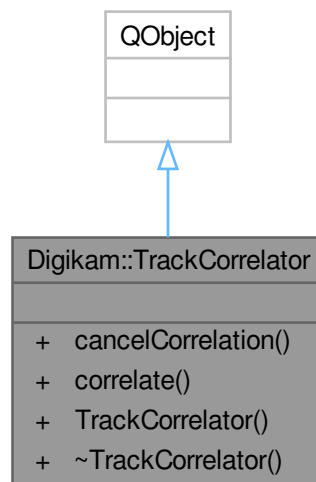
### Public Member Functions inherited from [Digikam::DWizardPage](#)

- QWizard \* **assistant** () const
- **DWizardPage** (QWizard \*const dlg, const QString &title)

- int **id** () const
- bool **isComplete** () const override
- void **removePageWidget** (QWidget \*const w)
- void **setComplete** (bool b)
- void **setLeftBottomPix** (const QIcon &icon)
- void **setLeftBottomPix** (const QPixmap &pix)
- void **setPageWidget** (QWidget \*const w)
- void **setShowLeftView** (bool v)

## 6.1488 Digikam::TrackCorrelator Class Reference

Inheritance diagram for Digikam::TrackCorrelator:



### Classes

- class [Correlation](#)
- class [CorrelationOptions](#)

### Public Types

- enum **CorrelationFlags** { **CorrelationFlagCoordinates** = 1 , **CorrelationFlagInterpolated** = 2 , **CorrelationFlagAltitude** = 3 }

### Signals

- void **signalAllItemsCorrelated** ()
- void **signalCorrelationCanceled** ()
- void **signalItemsCorrelated** (const Digikam::TrackCorrelator::Correlation::List &correlatedItems)

### Public Member Functions

- void **cancelCorrelation** ()
- void **correlate** (const Correlation::List &itemsToCorrelate, const [CorrelationOptions](#) &options)  
*GPS-correlate items.*
- **TrackCorrelator** ([TrackManager](#) \*const trackManager, QObject \*const parent=nullptr)

## 6.1489 Digikam::TrackCorrelator::Correlation Class Reference

### Public Types

- typedef QList< [Correlation](#) > **List**

### Public Attributes

- [GeoCoordinates](#) **coordinates**
- QDateTime **dateTime**
- int **fixType** = -1
- CorrelationFlags **flags** = CorrelationFlagCoordinates
- qreal **hDop** = -1.0
- int **nSatellites** = -1
- qreal **pDop** = -1.0
- qreal **speed** = -1.0
- QVariant **userData**

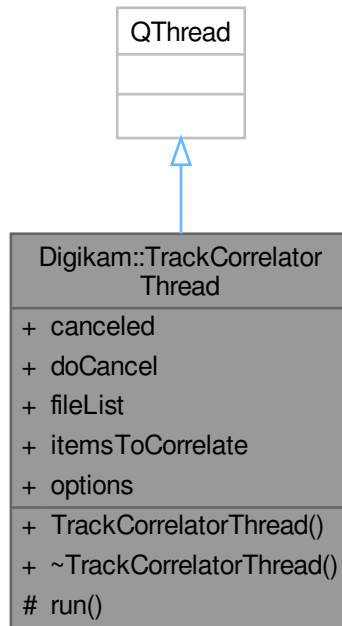
## 6.1490 Digikam::TrackCorrelator::CorrelationOptions Class Reference

### Public Attributes

- bool **interpolate** = false
- int **interpolationDstTime** = 0
- int **maxGapTime** = 0
- int **secondsOffset** = 0
- int **timeZoneOffset** = 0

## 6.1491 Digikam::TrackCorrelatorThread Class Reference

Inheritance diagram for Digikam::TrackCorrelatorThread:



### Signals

- void **signalItemsCorrelated** (const Digikam::TrackCorrelator::Correlation::List &correlatedItems)

### Public Member Functions

- **TrackCorrelatorThread** (QObject \*const parent=nullptr)

### Public Attributes

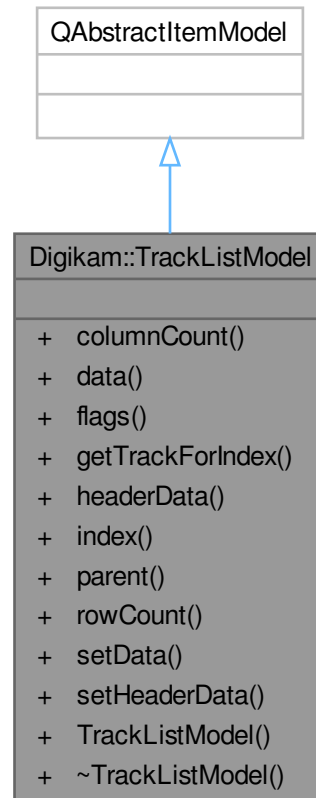
- bool **canceled** = false
- bool **doCancel** = false
- TrackManager::Track::List **fileList**
- TrackCorrelator::Correlation::List **itemsToCorrelate**
- [TrackCorrelator::CorrelationOptions](#) **options**

### Protected Member Functions

- void **run** () override

## 6.1492 Digikam::TrackListModel Class Reference

Inheritance diagram for Digikam::TrackListModel:



### Public Member Functions

- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- [TrackManager::Track](#) **getTrackForIndex** (const QModelIndex &index) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role) const override
- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex **parent** (const QModelIndex &index) const override
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role) override
- bool **setHeaderData** (int section, Qt::Orientation orientation, const QVariant &value, int role) override
- **TrackListModel** ([TrackManager](#) \*const trackManager, QObject \*const parent)

## 6.1492.1 Member Function Documentation

### 6.1492.1.1 headerData()

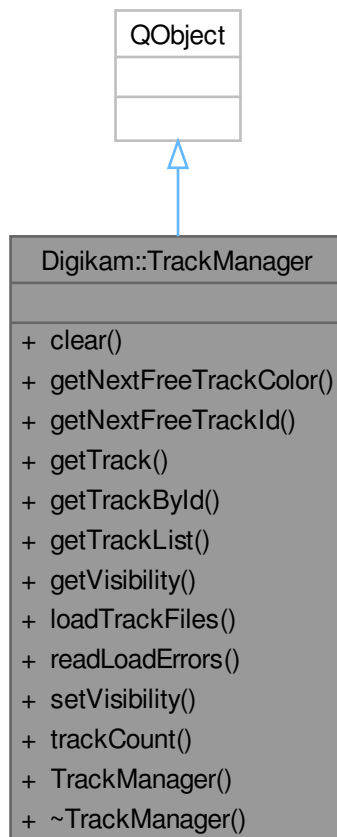
```
QVariant Digikam::TrackListModel::headerData (
    int section,
    Qt::Orientation orientation,
    int role ) const [override]
```

### 6.1492.1.2 index()

```
QModelIndex Digikam::TrackListModel::index (
    int row,
    int column,
    const QModelIndex & parent = QModelIndex() ) const [override]
```

## 6.1493 Digikam::TrackManager Class Reference

Inheritance diagram for Digikam::TrackManager:



## Classes

- class [Track](#)
- class [TrackPoint](#)

## Public Types

- enum **ChangeFlag** { **ChangeTrackPoints** = 1 , **ChangeMetadata** = 2 , **ChangeRemoved** = 4 , **ChangeAdd** = ChangeTrackPoints | ChangeMetadata }
- typedef quint32 **Id**  
*NOTE: we assume here that we will never load more than uint32\_max tracks.*
- typedef QPair< [Id](#), ChangeFlag > **TrackChanges**

## Signals

- void **signalAllTrackFilesReady** ()
- void **signalTrackFilesReadyAt** (const int startIndex, const int endIndex)
- void **signalTracksChanged** (const QList< TrackManager::TrackChanges > &trackChanges)
- void **signalVisibilityChanged** (const bool newValue)

## Public Member Functions

- void [clear](#) ()
- QColor **getNextFreeTrackColor** ()
- quint64 **getNextFreeTrackId** ()
- const [Track](#) & **getTrack** (const int index) const
- [Track](#) **getTrackById** (const quint64 trackId) const
- Track::List **getTrackList** () const
- bool **getVisibility** () const
- void **loadTrackFiles** (const QList< QUrl > &urls)
- QList< QPair< QUrl, QString > > **readLoadErrors** ()
- void **setVisibility** (const bool value)
- int **trackCount** () const
- **TrackManager** (QObject \*const parent=nullptr)

## 6.1493.1 Member Function Documentation

### 6.1493.1.1 clear()

```
void Digikam::TrackManager::clear ( )
```

## 6.1494 Digikam::TrackManager::Track Class Reference

### Public Types

- enum **Flags** { **FlagVisible** = 1 , **FlagDefault** = FlagVisible }
- typedef QList< [Track](#) > **List**



**Public Attributes**

- QColor **color** = Qt::red
- Flags **flags** = FlagDefault
- Id **id** = 0  
*0 means no track id assigned yet*
- QList< TrackPoint > **points**
- QUrl **url**

**6.1495 Digikam::TrackManager::TrackPoint Class Reference****Public Types**

- typedef QList< TrackPoint > **List**

**Static Public Member Functions**

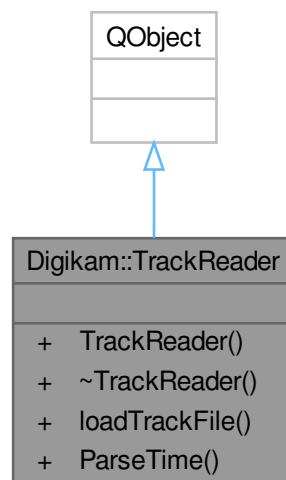
- static bool **EarlierThan** (const TrackPoint &a, const TrackPoint &b)

**Public Attributes**

- GeoCoordinates **coordinates**
- QDateTime **dateTime**
- int **fixType** = -1
- qreal **hDop** = -1.0
- int **nSatellites** = -1
- qreal **pDop** = -1.0
- qreal **speed** = -1.0

**6.1496 Digikam::TrackReader Class Reference**

Inheritance diagram for Digikam::TrackReader:



**Classes**

- class [TrackReadResult](#)

**Public Member Functions**

- **TrackReader** ([TrackReadResult](#) \*const dataTarget)

**Static Public Member Functions**

- static [TrackReadResult](#) **loadTrackFile** (const QUrl &url)
- static QDateTime **ParseTime** (const QString &tstring)

**Friends**

- class **::TestTracks**

## 6.1497 Digikam::TrackReader::TrackReadResult Class Reference

**Public Types**

- typedef QList< [TrackReadResult](#) > **List**

**Public Attributes**

- bool **isValid** = false
- QString **loadError**
- [TrackManager::Track](#) **track**

## 6.1498 Digikam::TrainerWorker Class Reference

Inheritance diagram for Digikam::TrainerWorker:



### Public Slots

- void [process](#) (const FacePipelineExtendedPackage::Ptr &package)

## Public Slots inherited from [Digikam::WorkerObject](#)

- void [deactivate](#) ([DeactivatingMode](#) mode=[FlushSignals](#))
- void [schedule](#) ()

## Signals

- void **processed** (const [FacePipelineExtendedPackage::Ptr](#) &package)

## Signals inherited from [Digikam::WorkerObject](#)

- void **finished** ()
- void **started** ()

## Public Member Functions

- **TrainerWorker** ([FacePipeline::Private](#) \*const dd)

## Public Member Functions inherited from [Digikam::WorkerObject](#)

- bool [connectAndSchedule](#) (const [QObject](#) \*sender, const char \*signal, const char \*method, [Qt::](#)↳ [ConnectionType](#) type=[Qt::AutoConnection](#)) const
- [QThread::Priority](#) **priority** () const
- void [setPriority](#) ([QThread::Priority](#) priority)
- State **state** () const
- void **wait** ()
- [WorkerObject](#) ()

## Protected Member Functions

- void [aboutToDeactivate](#) () override

## Protected Member Functions inherited from [Digikam::WorkerObject](#)

- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** ([WorkerObjectRunnable](#) \*loop)
- bool **event** ([QEvent](#) \*e) override
- void **removeRunnable** ([WorkerObjectRunnable](#) \*loop)
- void **run** ()
- void **setEventLoop** ([QEventLoop](#) \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

## Protected Attributes

- [FacePipeline::Private](#) \*const **d** = nullptr
- [FaceltemRetriever](#) **imageRetriever**
- [FacialRecognitionWrapper](#) **recognizer**

## Additional Inherited Members

### Public Types inherited from [Digikam::WorkerObject](#)

- enum [DeactivatingMode](#) { [FlushSignals](#) , [KeepSignals](#) , [PhaseOut](#) }
- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }

### Static Public Member Functions inherited from [Digikam::WorkerObject](#)

- static bool [connectAndSchedule](#) (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool [disconnectAndSchedule](#) (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

## 6.1498.1 Member Function Documentation

### 6.1498.1.1 [aboutToDeactivate\(\)](#)

```
void Digikam::TrainerWorker::aboutToDeactivate ( ) [override], [protected], [virtual]
```

Called from [deactivate\(\)](#), typically from a different thread than the worker thread, possibly the UI thread. You can stop any extra controlled threads here. Immediately afterwards, an event will be sent to the working thread which will cause the event loop to quit. ([aboutToQuitLoop\(\)](#))

Reimplemented from [Digikam::WorkerObject](#).

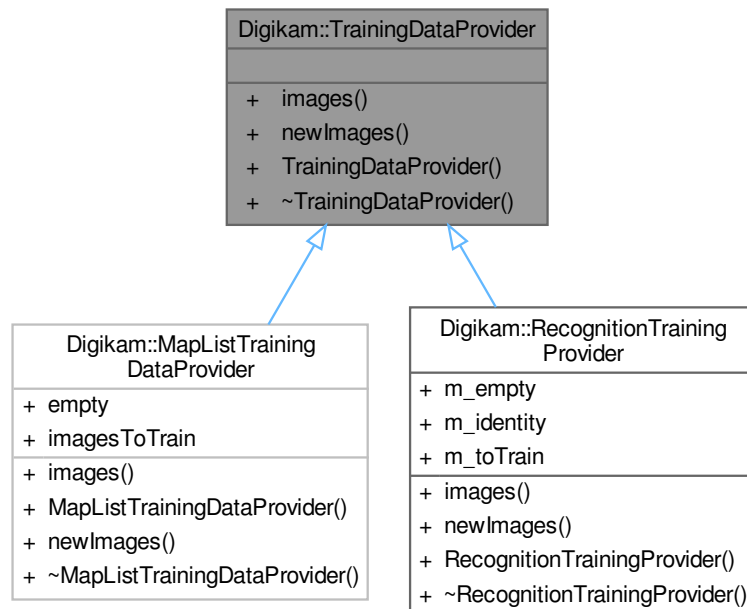
### 6.1498.1.2 [process](#)

```
void Digikam::TrainerWorker::process (
    const FacePipelineExtendedPackage::Ptr & package ) [slot]
```

TODO: investigate this method.

## 6.1499 Digikam::TrainingDataProvider Class Reference

Inheritance diagram for Digikam::TrainingDataProvider:



### Public Member Functions

- virtual [ImageListProvider](#) \* `images` (const [Identity](#) &identity)=0
- virtual [ImageListProvider](#) \* `newImages` (const [Identity](#) &identity)=0

### 6.1499.1 Detailed Description

A [TrainingDataProvider](#) provides a call-back interface for the training process to retrieve the necessary information. It is not specified, but depends on the backend which of the methods in which order and for which identities will be called.

### 6.1499.2 Member Function Documentation

#### 6.1499.2.1 `images()`

```
virtual ImageListProvider * Digikam::TrainingDataProvider::images (
    const Identity & identity ) [pure virtual]
```

Provides all images known for the given identity. Ownership of the returned object stays with the [TrainingDataProvider](#).

Implemented in [Digikam::RecognitionTrainingProvider](#).

### 6.1499.2.2 newImages()

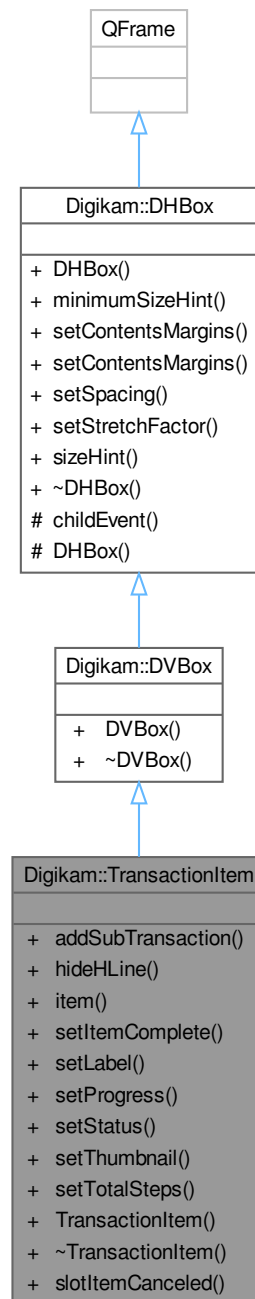
```
virtual ImageListProvider * Digikam::TrainingDataProvider::newImages (
    const Identity & identity ) [pure virtual]
```

Provides those images for the given identity that have not yet been supplied for training. Ownership of the returned object stays with the [TrainingDataProvider](#).

Implemented in [Digikam::RecognitionTrainingProvider](#).

## 6.1500 Digikam::TransactionItem Class Reference

Inheritance diagram for Digikam::TransactionItem:



### Public Slots

- void **slotItemCanceled** ()



## Public Member Functions

- void **addSubTransaction** ([ProgressItem](#) \*const item)
- void **hideHLine** ()
- [ProgressItem](#) \* **item** () const
- void **setItemComplete** ()  
*The progressitem is deleted immediately, we take 5s to go out, so better not use mItem during this time.*
- void **setLabel** (const QString &)
- void **setProgress** (int progress)
- void **setStatus** (const QString &)
- void **setThumbnail** (const QPixmap &)
- void **setTotalSteps** (int totalSteps)
- **TransactionItem** (QWidget \*const parent, [ProgressItem](#) \*const item, bool first)

## Public Member Functions inherited from [Digikam::DVBox](#)

- **DVBox** (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DHBox](#)

- **DHBox** (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentMargins** (const QMargins &margins)
- void **setContentMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1500.1 Member Function Documentation

### 6.1500.1.1 setStatus()

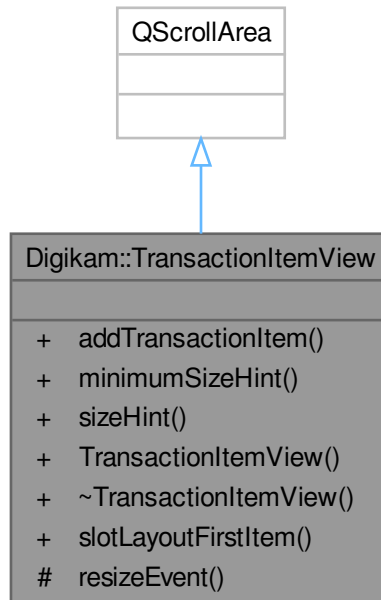
```
void Digikam::TransactionItem::setStatus (  
    const QString & status )
```

#### Note

the given text is interpreted as RichText, so you might need to use `.toHtmlEscaped()` it before passing.

## 6.1501 Digikam::TransactionItemView Class Reference

Inheritance diagram for Digikam::TransactionItemView:



### Public Slots

- void `slotLayoutFirstItem ()`

### Signals

- void `signalTransactionViewsIsEmpty ()`

### Public Member Functions

- `TransactionItem * addTransactionItem (ProgressItem *item, bool first)`
- `QSize minimumSizeHint ()` const override
- `QSize sizeHint ()` const override
- `TransactionItemView (QWidget *const parent=nullptr, const QString &name=QString())`

### Protected Member Functions

- void `resizeEvent (QResizeEvent *event)` override

## 6.1502 Digikam::TransitionMngr Class Reference

### Classes

- class [Private](#)

### Public Types

- enum **TransType** {  
**None** = 0 , **ChessBoard** , **MeltDown** , **Sweep** ,  
**Mosaic** , **Cubism** , **Growing** , **HorizontalLines** ,  
**VerticalLines** , **CircleOut** , **MultiCircleOut** , **SpiralIn** ,  
**Blobs** , **Fade** , **SlideL2R** , **SlideR2L** ,  
**SlideT2B** , **SlideB2T** , **PushL2R** , **PushR2L** ,  
**PushT2B** , **PushB2T** , **SwapL2R** , **SwapR2L** ,  
**SwapT2B** , **SwapB2T** , **BlurIn** , **BlurOut** ,  
**Random** }

### Public Member Functions

- QImage **currentFrame** (int &tmout)
- void **setInImage** (const QImage &iimg)
- void **setOutImage** (const QImage &oimg)
- void **setOutputSize** (const QSize &size)
- void **setTransition** (TransType type)

### Static Public Member Functions

- static QMap< TransType, QString > **transitionNames** ()

## 6.1503 Digikam::TransitionMngr::Private Class Reference

### Public Types

- typedef int(TransitionMngr::Private::\* **TransMethod**) (bool)

### Public Member Functions

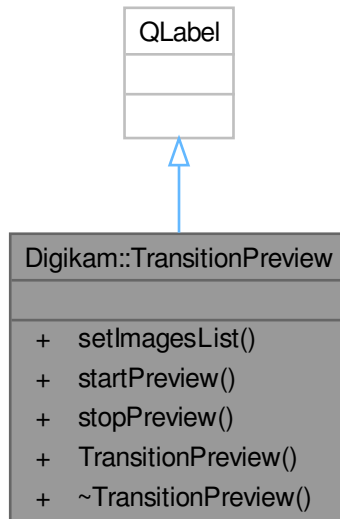
- TransitionMngr::TransType **getRandomTransition** () const
- void **registerTransitions** ()

## Public Attributes

- double **eff\_alpha** = 0
- QImage **eff\_curFrame**
- TransitionMngr::TransType **eff\_curTransition** = TransitionMngr::None
- int **eff\_dx** = 0
- int **eff\_dy** = 0
- double **eff\_fd** = 0
- double **eff\_fx** = 0
- double **eff\_fy** = 0
- int **eff\_h** = 0
- int **eff\_i** = 0
- QImage **eff\_inImage**
- int \* **eff\_intArray** = nullptr
- bool **eff\_isRunning** = false
- int **eff\_ix** = 0
- int **eff\_iy** = 0
- int **eff\_j** = 0
- QImage **eff\_outImage**
- QSize **eff\_outSize**
- QPolygon **eff\_pa** = QPolygon(4)
- bool **eff\_pdone** = false
- bool \*\* **eff\_pixelMatrix** = nullptr
- int **eff\_psx** = 0
- int **eff\_psy** = 0
- int **eff\_px** = 0
- int **eff\_py** = 0
- int **eff\_subType** = 0
- QMap< TransitionMngr::TransType, TransMethod > **eff\_transList**
- int **eff\_w** = 0
- int **eff\_wait** = 0
- int **eff\_x** = 0
- int **eff\_x0** = 0
- int **eff\_x1** = 0
- int **eff\_y** = 0
- int **eff\_y0** = 0
- int **eff\_y1** = 0

## 6.1504 Digikam::TransitionPreview Class Reference

Inheritance diagram for Digikam::TransitionPreview:

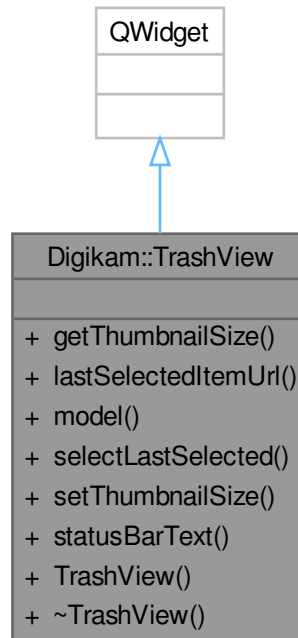


### Public Member Functions

- void **setImagesList** (const QList< QUrl > &images)
- void **startPreview** (TransitionMngr::TransType eff)
- void **stopPreview** ()
- **TransitionPreview** (QWidget \*const parent=nullptr)

## 6.1505 Digikam::TrashView Class Reference

Inheritance diagram for Digikam::TrashView:



### Signals

- void **selectionChanged** ()
- void **signalEmptytrash** ()

### Public Member Functions

- [ThumbnailSize](#) `getThumbnailSize` () const
- [QUrl](#) `lastSelectedItemUrl` () const
- [DTrashItemModel](#) \* `model` () const
- void **selectLastSelected** ()  
*Highlights the last selected item when the view gets focus.*
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize)  
*set thumbnail size to give to model*
- [QString](#) `statusBarText` () const
- **TrashView** ([QWidget](#) \*const parent=nullptr)

## 6.1505.1 Member Function Documentation

### 6.1505.1.1 getThumbnailSize()

```
ThumbnailSize Digikam::TrashView::getThumbnailSize ( ) const
```

#### Returns

current thumbnail size

### 6.1505.1.2 lastSelectedItemUrl()

```
QUrl Digikam::TrashView::lastSelectedItemUrl ( ) const
```

#### Returns

QUrl to the last selected item in view

### 6.1505.1.3 model()

```
DTrashItemModel * Digikam::TrashView::model ( ) const
```

#### Returns

model used for the view

### 6.1505.1.4 setThumbnailSize()

```
void Digikam::TrashView::setThumbnailSize (
    const ThumbnailSize & thumbSize )
```

#### Parameters

<i>thumbSize</i>	size to set
------------------	-------------

### 6.1505.1.5 statusBarText()

```
QString Digikam::TrashView::statusBarText ( ) const
```

#### Returns

text for the main status bar

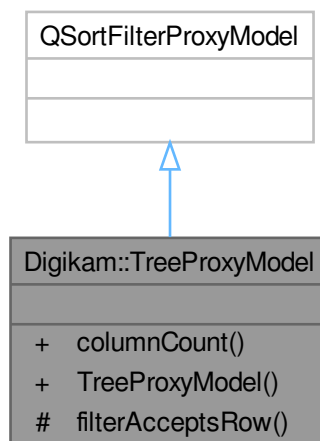
## 6.1506 Digikam::TreeBranch Class Reference

### Public Attributes

- QString **data**
- QString **help**
- QList< TreeBranch \* > **newChildren**
- QList< TreeBranch \* > **oldChildren**
- TreeBranch \* **parent** = nullptr
- QPersistentModelIndex **sourceIndex**
- QList< TreeBranch \* > **spacerChildren**
- Type **type** = TypeChild

## 6.1507 Digikam::TreeProxyModel Class Reference

Inheritance diagram for Digikam::TreeProxyModel:



### Signals

- void **signalFilterAccepts** (bool)

### Public Member Functions

- int **columnCount** (const QModelIndex &) const override
- **TreeProxyModel** (QObject \*const parent=nullptr)

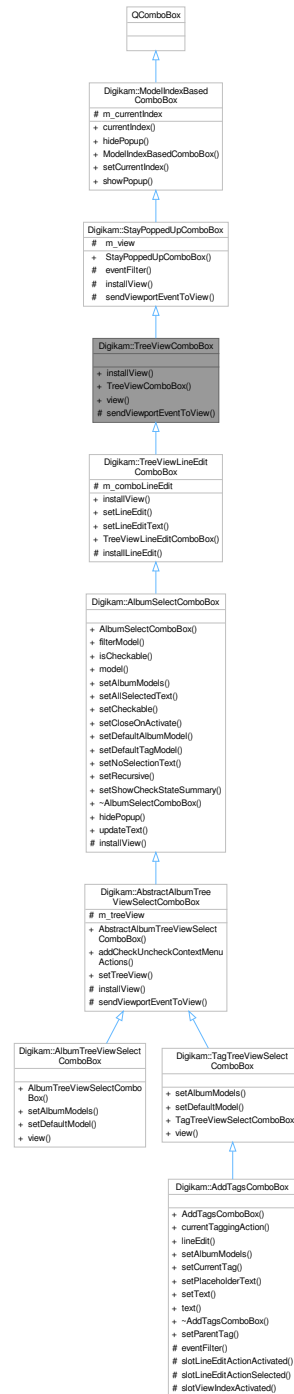
### Protected Member Functions

- bool **filterAcceptsRow** (int srow, const QModelIndex &sparent) const override



## 6.1508 Digikam::TreeViewComboBox Class Reference

Inheritance diagram for Digikam::TreeViewComboBox:



### Public Member Functions

- virtual void [installView](#) (QAbstractItemView \*[view](#)=nullptr)
- [TreeViewComboBox](#) (QWidget \*[parent](#)=nullptr)
- QTreeView \* [view](#) () const

## Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex **currentIndex** () const
- void **hidePopup** () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void **setCurrentIndex** (const QModelIndex &index)
- void **showPopup** () override

## Protected Member Functions

- void [sendViewportEventToView](#) (QEvent \*e) override

## Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool **eventFilter** (QObject \*watched, QEvent \*event) override
- void [installView](#) (QAbstractItemView \*view)

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

- QAbstractItemView \* **m\_view** = nullptr

## Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex **m\_currentIndex**

## 6.1508.1 Constructor & Destructor Documentation

### 6.1508.1.1 [TreeViewComboBox\(\)](#)

```
Digikam::TreeViewComboBox::TreeViewComboBox (
    QWidget * parent = nullptr ) [explicit]
```

This class provides a QComboBox with a QTreeView instead of the usual QListView. You need three steps↔: Construct the object, call setModel() with an appropriate QAbstractItemModel, then call [installView\(\)](#) to replace the standard combo box view with a QTreeView.

## 6.1508.2 Member Function Documentation

### 6.1508.2.1 installView()

```
void Digikam::TreeViewComboBox::installView (
    QAbstractItemView * view = nullptr ) [virtual]
```

Replace the standard combo box list view with a QTreeView. Call this after installing an appropriate model.

Reimplemented in [Digikam::AlbumSelectComboBox](#), [Digikam::AbstractAlbumTreeViewSelectComboBox](#), and [Digikam::TreeViewLineEditComboBox](#).

### 6.1508.2.2 sendViewportEventToView()

```
void Digikam::TreeViewComboBox::sendViewportEventToView (
    QEvent * e ) [override], [protected], [virtual]
```

Implement in subclass: Send the given event to the viewportEvent() method of m\_view. This method is protected for a usual QAbstractItemView. You can override, pass a view, and call parent implementation. The existing view will be used. You must then also reimplement sendViewportEventToView.

Implements [Digikam::StayPoppedUpComboBox](#).

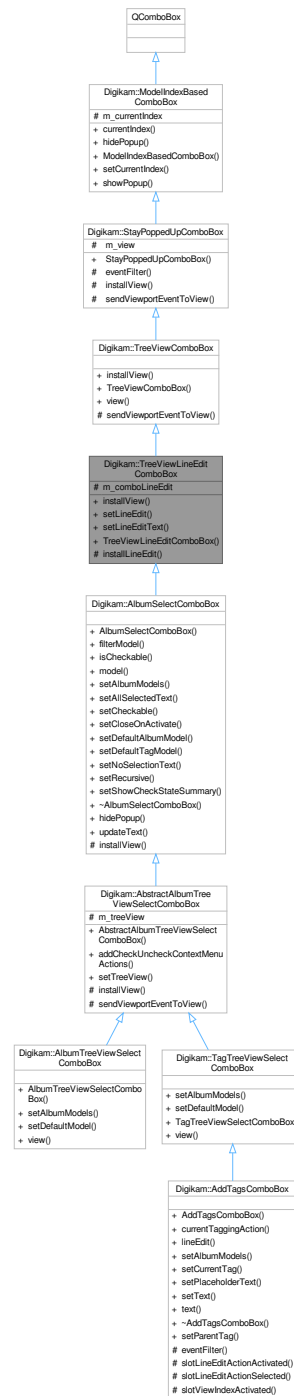
### 6.1508.2.3 view()

```
QTreeView * Digikam::TreeViewComboBox::view ( ) const
```

Returns the QTreeView of this class. Valid after [installView\(\)](#) has been called

## 6.1509 Digikam::TreeViewLineEditComboBox Class Reference

Inheritance diagram for Digikam::TreeViewLineEditComboBox:



### Public Member Functions

- void [installView](#) (QAbstractItemView \*[view](#)=nullptr) override
- void [setLineEdit](#) (QLineEdit \*[edit](#))
- void [setLineEditText](#) (const QString &[text](#))
- [TreeViewLineEditComboBox](#) (QWidget \*[parent](#)=nullptr)

### Public Member Functions inherited from [Digikam::TreeViewComboBox](#)

- [TreeViewComboBox](#) (QWidget \*parent=nullptr)
- QTreeView \* [view](#) () const

### Public Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- [StayPoppedUpComboBox](#) (QWidget \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::ModelIndexBasedComboBox](#)

- QModelIndex [currentIndex](#) () const
- void [hidePopup](#) () override
- [ModelIndexBasedComboBox](#) (QWidget \*const parent=nullptr)
- void [setCurrentIndex](#) (const QModelIndex &index)
- void [showPopup](#) () override

### Protected Member Functions

- virtual void [installLineEdit](#) ()

### Protected Member Functions inherited from [Digikam::TreeViewComboBox](#)

- void [sendViewportEventToView](#) (QEvent \*e) override

### Protected Member Functions inherited from [Digikam::StayPoppedUpComboBox](#)

- bool [eventFilter](#) (QObject \*watched, QEvent \*event) override
- void [installView](#) (QAbstractItemView \*view)

### Protected Attributes

- QLineEdit \* [m\\_comboLineEdit](#) = nullptr

### Protected Attributes inherited from [Digikam::StayPoppedUpComboBox](#)

- QAbstractItemView \* [m\\_view](#) = nullptr

### Protected Attributes inherited from [Digikam::ModelIndexBasedComboBox](#)

- QPersistentModelIndex [m\\_currentIndex](#)

## 6.1509.1 Constructor & Destructor Documentation

### 6.1509.1.1 `TreeViewLineEditComboBox()`

```
Digikam::TreeViewLineEditComboBox::TreeViewLineEditComboBox (
    QWidget *const parent = nullptr ) [explicit]
```

This class provides a [TreeViewComboBox](#) with a read-only line edit. The text in the line edit can be adjusted. The combo box will open on a click on the line edit. You need three steps: Construct the object, call `setModel()` with an appropriate `QAbstractItemModel`, then call [installView\(\)](#) to replace the standard combo box view with a `QTreeView`.

## 6.1509.2 Member Function Documentation

### 6.1509.2.1 `installLineEdit()`

```
void Digikam::TreeViewLineEditComboBox::installLineEdit ( ) [protected], [virtual]
```

Sets a line edit. Called by [installView\(\)](#). The default implementation is described above. An empty implementation will keep the default `QComboBox` line edit.

### 6.1509.2.2 `installView()`

```
void Digikam::TreeViewLineEditComboBox::installView (
    QAbstractItemView * view = nullptr ) [override], [virtual]
```

Replace the standard combo box list view with a `QTreeView`. Call this after installing an appropriate model.

Reimplemented from [Digikam::TreeViewComboBox](#).

### 6.1509.2.3 `setLineEditText()`

```
void Digikam::TreeViewLineEditComboBox::setLineEditText (
    const QString & text )
```

Set the text of the line edit (the text that is visible if the popup is not opened). Applicable only for default [installLineEdit\(\)](#) implementation.

## 6.1510 Digikam::TrimmedModifier Class Reference

Inheritance diagram for Digikam::TrimmedModifier:



### Public Member Functions

- QString [parseOperation](#) ([ParseSettings](#) &settings, const QRegularExpressionMatch &match) override

## Public Member Functions inherited from [Digikam::Modifier](#)

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- ParseResults **parse** (ParseSettings &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- virtual void **reset** ()
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from [Digikam::Rule](#)

- void **signalTokenTriggered** (const QString &)

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool **addToken** (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void **setUseTokenMenu** (bool value)

## 6.1510.1 Member Function Documentation

### 6.1510.1.1 parseOperation()

```
QString Digikam::TrimmedModifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [virtual]
```

TODO: describe me



## Parameters

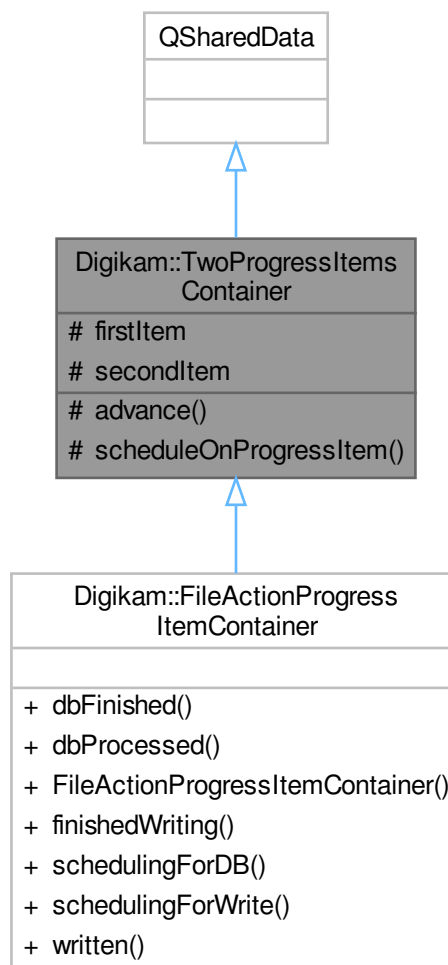
<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Modifier](#).

## 6.1511 Digikam::TwoProgressItemsContainer Class Reference

Inheritance diagram for Digikam::TwoProgressItemsContainer:



**Protected Member Functions**

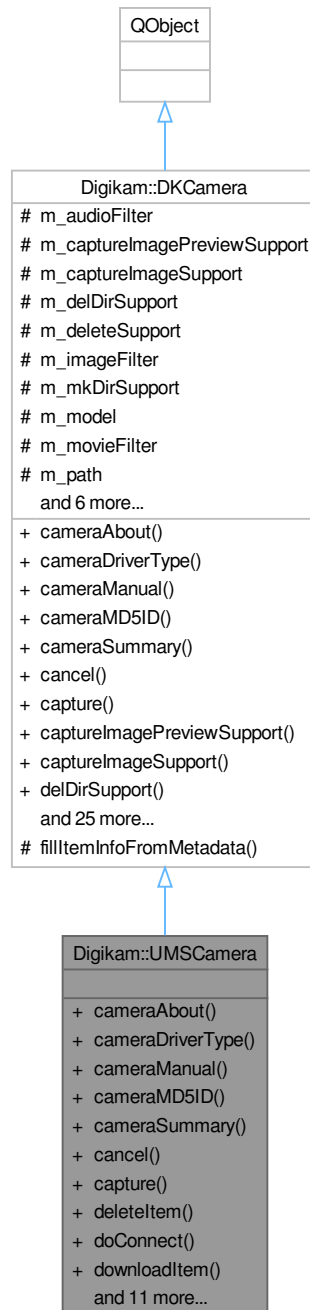
- void **advance** (QAtomicPointer< [ProgressItem](#) > &ptr, int n)
- void **scheduleOnProgressItem** (QAtomicPointer< [ProgressItem](#) > &ptr, int total, const QString &action, [FileActionProgressItemCreator](#) \*const creator)

**Protected Attributes**

- QAtomicPointer< [ProgressItem](#) > **firstItem**
- QAtomicPointer< [ProgressItem](#) > **secondItem**

## 6.1512 Digikam::UMSCamera Class Reference

Inheritance diagram for Digikam::UMSCamera:



### Public Member Functions

- bool `cameraAbout` (QString &about) override
- DKCamera::CameraDriverType `cameraDriverType` () override

- bool [cameraManual](#) (QString &manual) override
- QByteArray [cameraMD5ID](#) () override
- bool [cameraSummary](#) (QString &summary) override
- void [cancel](#) () override
- bool [capture](#) (CamItemInfo &itemInfo) override
  - Method not supported by UMS camera.*
- bool [deleteItem](#) (const QString &folder, const QString &itemName) override
- bool [doConnect](#) () override
- bool [downloadItem](#) (const QString &folder, const QString &itemName, const QString &saveFile) override
- bool [getFolders](#) (const QString &folder) override
- bool [getFreeSpace](#) (qint64 &bytesSize, qint64 &bytesAvail) override
  - NOTE: implemented in gui, outside the camera thread.*
- void [getItemInfo](#) (const QString &folder, const QString &itemName, [CamItemInfo](#) &info, bool useMetadata) override
- bool [getItemsInfoList](#) (const QString &folder, bool useMetadata, [CamItemInfoList](#) &infoList) override
  - If getImageDimensions is false, the camera shall set width and height to -1 if the values are not immediately available.*
- bool [getMetadata](#) (const QString &folder, const QString &itemName, [DMetadata](#) &meta) override
- bool [getPreview](#) (QImage &preview) override
  - Method not supported by UMS camera.*
- bool [getThumbnail](#) (const QString &folder, const QString &itemName, QImage &thumbnail) override
- bool [setLockItem](#) (const QString &folder, const QString &itemName, bool lock) override
- **UMSCamera** (const QString &title, const QString &model, const QString &port, const QString &path)
- bool [uploadItem](#) (const QString &folder, const QString &itemName, const QString &localFile, [CamItemInfo](#) &info) override

## Public Member Functions inherited from [Digikam::DKCamera](#)

- bool [captureImagePreviewSupport](#) () const
- bool [captureImageSupport](#) () const
- bool [delDirSupport](#) () const
- bool [deleteSupport](#) () const
- **DKCamera** (const QString &title, const QString &model, const QString &port, const QString &path)
- QString [mimeType](#) (const QString &fileext) const
- bool [mkDirSupport](#) () const
- QString [model](#) () const
- QString [path](#) () const
- QString [port](#) () const
- void [printSupportedFeatures](#) ()
- bool [thumbnailSupport](#) () const
- QString [title](#) () const
- bool [uploadSupport](#) () const
- QString [uuid](#) () const

## Additional Inherited Members

## Public Types inherited from [Digikam::DKCamera](#)

- enum [CameraDriverType](#) { [GPhotoDriver](#) = 0 , [UMSDriver](#) }

## Signals inherited from [Digikam::DKCamera](#)

- void [signalFolderList](#) (const QStringList &)

## Protected Member Functions inherited from [Digikam::DKCamera](#)

- void `fillItemInfoFromMetadata` ([CamItemInfo](#) &item, const [DMetadata](#) &meta) const

## Protected Attributes inherited from [Digikam::DKCamera](#)

- `QString` `m_audioFilter`
- `bool` `m_captureImagePreviewSupport` = false
- `bool` `m_captureImageSupport` = false
- `bool` `m_delDirSupport` = false
- `bool` `m_deleteSupport` = false
- `QString` `m_imageFilter`
- `bool` `m_mkDirSupport` = false
- `QString` `m_model`
- `QString` `m_movieFilter`
- `QString` `m_path`
- `QString` `m_port`
- `QString` `m_rawFilter`
- `bool` `m_thumbnailSupport` = false
- `QString` `m_title`
- `bool` `m_uploadSupport` = false
- `QString` `m_uuid`

### 6.1512.1 Detailed Description

USB Mass Storage camera Implementation of abstract type [DKCamera](#)

### 6.1512.2 Member Function Documentation

#### 6.1512.2.1 `cameraAbout()`

```
bool Digikam::UMSCamera::cameraAbout (
    QString & about ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.2 `cameraDriverType()`

```
DKCamera::CameraDriverType Digikam::UMSCamera::cameraDriverType ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.3 `cameraManual()`

```
bool Digikam::UMSCamera::cameraManual (
    QString & manual ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.4 cameraMD5ID()

```
QByteArray Digikam::UMSCamera::cameraMD5ID ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.5 cameraSummary()

```
bool Digikam::UMSCamera::cameraSummary (
    QString & summary ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.6 cancel()

```
void Digikam::UMSCamera::cancel ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.7 capture()

```
bool Digikam::UMSCamera::capture (
    CamItemInfo & itemInfo ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.8 deleteItem()

```
bool Digikam::UMSCamera::deleteItem (
    const QString & folder,
    const QString & itemName ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.9 doConnect()

```
bool Digikam::UMSCamera::doConnect ( ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

#### 6.1512.2.10 downloadItem()

```
bool Digikam::UMSCamera::downloadItem (
    const QString & folder,
    const QString & itemName,
    const QString & saveFile ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.11 getFolders()

```
bool Digikam::UMSCamera::getFolders (
    const QString & folder ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.12 getFreeSpace()

```
bool Digikam::UMSCamera::getFreeSpace (
    quint64 & bytesSize,
    quint64 & bytesAvail ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.13 getItemInfo()

```
void Digikam::UMSCamera::getItemInfo (
    const QString & folder,
    const QString & itemName,
    CamItemInfo & info,
    bool useMetadata ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.14 getItemsInfoList()

```
bool Digikam::UMSCamera::getItemsInfoList (
    const QString & folder,
    bool useMetadata,
    CamItemInfoList & infoList ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.15 getMetadata()

```
bool Digikam::UMSCamera::getMetadata (
    const QString & folder,
    const QString & itemName,
    DMetadata & meta ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.16 getPreview()

```
bool Digikam::UMSCamera::getPreview (
    QImage & preview ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.17 getThumbnail()

```
bool Digikam::UMSCamera::getThumbnail (
    const QString & folder,
    const QString & itemName,
    QImage & thumbnail ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.18 setLockItem()

```
bool Digikam::UMSCamera::setLockItem (
    const QString & folder,
    const QString & itemName,
    bool lock ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).

### 6.1512.2.19 uploadItem()

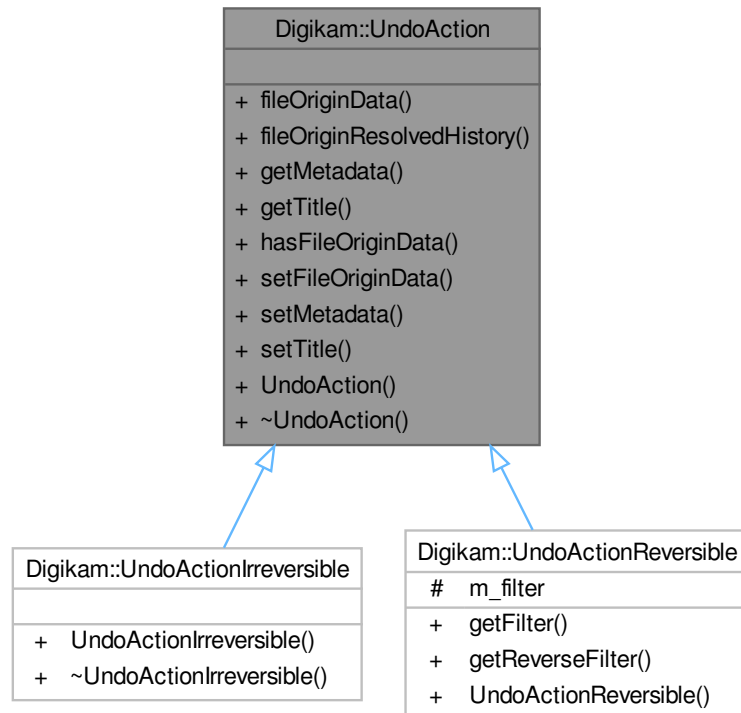
```
bool Digikam::UMSCamera::uploadItem (
    const QString & folder,
    const QString & itemName,
    const QString & localFile,
    CamItemInfo & info ) [override], [virtual]
```

Implements [Digikam::DKCamera](#).



## 6.1513 Digikam::UndoAction Class Reference

Inheritance diagram for Digikam::UndoAction:

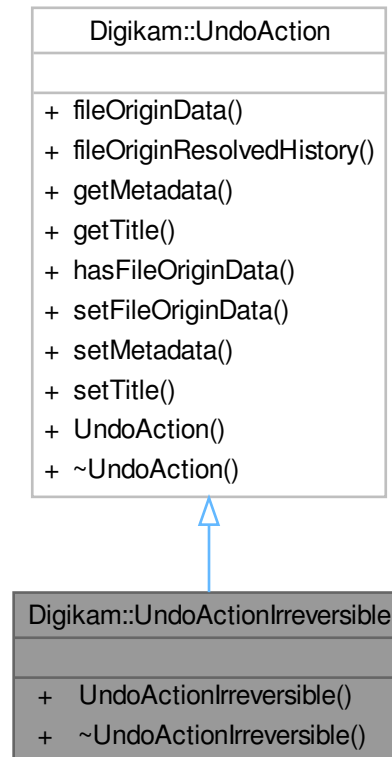


### Public Member Functions

- QVariant **fileOriginData** () const
- [DImageHistory](#) **fileOriginResolvedHistory** () const
- [UndoMetadataContainer](#) **getMetadata** () const
- QString **getTitle** () const
- bool **hasFileOriginData** () const
- void **setFileOriginData** (const QVariant &data, const [DImageHistory](#) &resolvedInitialHistory)
- void **setMetadata** (const [UndoMetadataContainer](#) &)
- void **setTitle** (const QString &title)
- **UndoAction** (const [EditorCore](#) \*const core)

## 6.1514 Digikam::UndoActionIrreversible Class Reference

Inheritance diagram for Digikam::UndoActionIrreversible:



### Public Member Functions

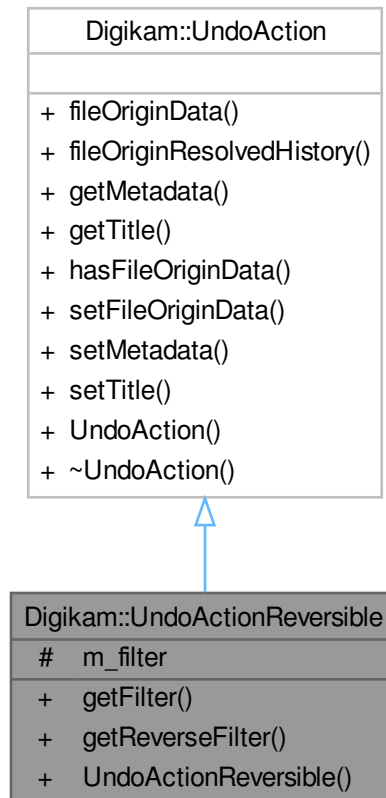
- **UndoActionIrreversible** (const [EditorCore](#) \*const core, const QString &title=QString())

### Public Member Functions inherited from [Digikam::UndoAction](#)

- QVariant **fileOriginData** () const
- [DImageHistory](#) **fileOriginResolvedHistory** () const
- [UndoMetadataContainer](#) **getMetadata** () const
- QString **getTitle** () const
- bool **hasFileOriginData** () const
- void **setFileOriginData** (const QVariant &data, const [DImageHistory](#) &resolvedInitialHistory)
- void **setMetadata** (const [UndoMetadataContainer](#) &)
- void **setTitle** (const QString &title)
- **UndoAction** (const [EditorCore](#) \*const core)

## 6.1515 Digikam::UndoActionReversible Class Reference

Inheritance diagram for Digikam::UndoActionReversible:



### Public Member Functions

- [DImgBuiltinFilter](#) **getFilter** () const
- [DImgBuiltinFilter](#) **getReverseFilter** () const
- **UndoActionReversible** (const [EditorCore](#) \*const core, const [DImgBuiltinFilter](#) &reversibleFilter)

### Public Member Functions inherited from [Digikam::UndoAction](#)

- QVariant **fileOriginData** () const
- [DImageHistory](#) **fileOriginResolvedHistory** () const
- [UndoMetadataContainer](#) **getMetadata** () const
- QString **getTitle** () const
- bool **hasFileOriginData** () const
- void **setFileOriginData** (const QVariant &data, const [DImageHistory](#) &resolvedInitialHistory)
- void **setMetadata** (const [UndoMetadataContainer](#) &)
- void **setTitle** (const QString &title)
- **UndoAction** (const [EditorCore](#) \*const core)

## Protected Attributes

- [DImgBuiltinFilter](#) `m_filter`

## 6.1516 Digikam::UndoCache Class Reference

### Public Member Functions

- void [clear](#) ()
- void [clearFrom](#) (int level)
- [DImg](#) [getData](#) (int level) const
- bool [putData](#) (int level, const [DImg](#) &img) const

### 6.1516.1 Member Function Documentation

#### 6.1516.1.1 clear()

```
void Digikam::UndoCache::clear ( )
```

Delete all cache files

#### 6.1516.1.2 clearFrom()

```
void Digikam::UndoCache::clearFrom (
    int level )
```

Delete all cache files starting from the given level upwards

#### 6.1516.1.3 getData()

```
DImg Digikam::UndoCache::getData (
    int level ) const
```

Get the image data from a cache file

#### 6.1516.1.4 putData()

```
bool Digikam::UndoCache::putData (
    int level,
    const DImg & img ) const
```

Write the image data into a cache file

## 6.1517 Digikam::UndoManager Class Reference

### Public Member Functions

- void **addAction** ([UndoAction](#) \*const action)
- bool **anyMoreRedo** () const
- bool **anyMoreUndo** () const
- int **availableRedoSteps** () const
- int **availableUndoSteps** () const
- void **clear** (bool clearCache=true)
- void **clearPreviousOriginData** ()
- [DImageHistory](#) **getImageHistoryOfFullRedo** () const  
*The history if all available redo steps are redone.*
- [QStringList](#) **getRedoHistory** () const
- [QStringList](#) **getUndoHistory** () const
- bool **hasChanges** () const
- bool **isAtOrigin** () const
- bool **putImageDataAndHistory** ([DImg](#) \*const img, int stepsBack) const
- void **redo** ()
- void **rollbackToOrigin** ()
- void **setOrigin** () const
- void **undo** ()
- [UndoManager](#) ([EditorCore](#) \*const core)

## 6.1518 Digikam::UndoMetadataContainer Class Reference

### Public Member Functions

- bool **changesIccProfile** (const [DImg](#) &target) const
- void **toImage** ([DImg](#) &img) const

### Static Public Member Functions

- static [UndoMetadataContainer](#) **fromImage** (const [DImg](#) &img)

### Public Attributes

- [DImageHistory](#) **history**
- [IccProfile](#) **profile**

### 6.1518.1 Member Function Documentation

#### 6.1518.1.1 fromImage()

```
UndoMetadataContainer Digikam::UndoMetadataContainer::fromImage (  
    const DImg & img ) [static]
```

Fill a container from the [DImg](#)

### 6.1518.1.2 toImage()

```
void Digikam::UndoMetadataContainer::toImage (
    DImg & img ) const
```

Write this container's values to the [DImg](#)

## 6.1519 Digikam::UndoState Class Reference

### Public Attributes

- bool **hasChanges** = false
- bool **hasRedo** = false
- bool **hasUndo** = false
- bool **hasUndoableChanges** = false

## 6.1520 Digikam::UniqueModifier Class Reference

Inheritance diagram for Digikam::UniqueModifier:



### Public Member Functions

- QString `parseOperation` (`ParseSettings &settings`, `const QRegularExpressionMatch &match`) override
- void `reset` () override

## Public Member Functions inherited from [Digikam::Modifier](#)

- **Modifier** (const QString &name, const QString &description)
- **Modifier** (const QString &name, const QString &description, const QString &icon)

## Public Member Functions inherited from [Digikam::Rule](#)

- QString **description** () const
- QPixmap **icon** (Rule::IconType type=Rule::Action) const
- bool **isValid** () const
- [ParseResults](#) **parse** ([ParseSettings](#) &settings)
- QRegularExpression & **regExp** () const
- QPushButton \* **registerButton** (QWidget \*parent)
- QAction \* **registerMenu** (QMenu \*parent)
- **Rule** (const QString &name)
- **Rule** (const QString &name, const QString &icon)
- TokenList & **tokens** () const
- bool **useTokenMenu** () const

## Additional Inherited Members

## Public Types inherited from [Digikam::Rule](#)

- enum **IconType** { **Action** = 0 , **Dialog** }

## Signals inherited from [Digikam::Rule](#)

- void **signalTokenTriggered** (const QString &)

## Static Public Member Functions inherited from [Digikam::Rule](#)

- static QString **escapeToken** (const QString &token)

## Protected Slots inherited from [Digikam::Rule](#)

- virtual void **slotTokenTriggered** (const QString &)

## Protected Member Functions inherited from [Digikam::Rule](#)

- bool **addToken** (const QString &id, const QString &description, const QString &actionName=QString())
- void **setDescription** (const QString &desc)
- void **setIcon** (const QString &pixmap)
- void **setRegExp** (const QRegularExpression &regExp)
- void **setUseTokenMenu** (bool value)

## 6.1520.1 Member Function Documentation

### 6.1520.1.1 parseOperation()

```
QString Digikam::UniqueModifier::parseOperation (
    ParseSettings & settings,
    const QRegularExpressionMatch & match ) [override], [virtual]
```

TODO: describe me



## Parameters

<i>settings</i>	contains settings
<i>match</i>	result of the regular expression match done in <code>Option::parse()</code>

## Returns

Implements [Digikam::Modifier](#).

**6.1520.1.2 reset()**

```
void Digikam::UniqueModifier::reset ( ) [override], [virtual]
```

Resets the parser to its initial state

Reimplemented from [Digikam::Rule](#).

## 6.1521 Digikam::UnsharpMaskFilter Class Reference

Inheritance diagram for Digikam::UnsharpMaskFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- void [readParameters](#) (const [FilterAction](#) &action) override
- **UnsharpMaskFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, double radius=1.0, double amount=1.0, double threshold=0.05, bool luma=false)
- **UnsharpMaskFilter** (QObject \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > [multithreadedSteps](#) (int stop, int start=0) const
- virtual bool [parametersSuccessfullyRead](#) () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void [setFilterVersion](#) (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void [setupAndStartDirectly](#) (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void [setupFilter](#) (const [DImg](#) &orgImage)
- virtual void [startFilter](#) ()
- virtual void [startFilterDirectly](#) ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void [setPriority](#) (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Additional Inherited Members

### Public Types inherited from [Digikam::DynamicThread](#)

- enum **State** { **Inactive** , **Scheduled** , **Running** , **Deactivating** }

## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void **cleanupFilter** ()
- **DImgThreadedFilter** (**DImgThreadedFilter** \*const master, const **DImg** &orgImage, const **DImg** &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void **initFilter** ()
- void **initMaster** ()
- void **initSlave** (**DImgThreadedFilter** \*const master, int progressBegin=0, int progressEnd=100)
- virtual int **modulateProgress** (int progress)
- void **postProgress** (int progress)
- virtual void **prepareDestImage** ()
- void **run** () override
- void **setSlave** (**DImgThreadedFilter** \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool **runningFlag** () const volatile
- void **shutDown** ()
- void **start** (QMutexLocker< QMutex > &locker)
- void **stop** (const QMutexLocker< QMutex > &locker)
- QMutex \* **threadMutex** () const
- void **wait** (QMutexLocker< QMutex > &locker)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- **DImg** m\_destImage
- **DImgThreadedFilter** \* m\_master = nullptr
- QString m\_name
- **DImg** m\_orgImage
- int m\_progressBegin = 0
- int **m\_progressCurrent** = 0
  - *To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int **m\_progressSpan** = 0
- **DImgThreadedFilter** \* m\_slave = nullptr
- int **m\_version** = 1
- bool **m\_wasCancelled** = false

## 6.1521.1 Member Function Documentation

### 6.1521.1.1 filterAction()

`FilterAction` Digikam::UnsharpMaskFilter::filterAction ( ) [override], [virtual]

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1521.1.2 filterIdentifier()

`QString` Digikam::UnsharpMaskFilter::filterIdentifier ( ) const [inline], [override], [virtual]

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1521.1.3 readParameters()

```
void Digikam::UnsharpMaskFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1522 Digikam::VersionFileInfo Class Reference

### Public Member Functions

- `QString` **filePath** ( ) const
- `QUrl` **fileUrl** ( ) const
- `bool` **isNull** ( ) const
- **VersionFileInfo** (const `QString` &path, const `QString` &fileName, const `QString` &format)

### Public Attributes

- `QString` **fileName**
- `QString` **format**
- `QString` **path**

## 6.1523 Digikam::VersionFileOperation Class Reference

### Public Types

- enum `Task` {  
    `NewFile` = 1 << 0 , `Replace` = 1 << 1 , `SaveAndDelete` = 1 << 2 , `MoveToIntermediate` = 1 << 3 ,  
    `StoreIntermediates` = 1 << 4 }

## Public Member Functions

- QStringList [allFilePaths](#) () const
- [VersionFileOperation](#) ()=default

## Public Attributes

- [VersionFileInfo](#) [intermediateForLoadedFile](#)
- QMap< int, [VersionFileInfo](#) > [intermediates](#)
- [VersionFileInfo](#) [loadedFile](#)
- [VersionFileInfo](#) [saveFile](#)
- Tasks [tasks](#)

## 6.1523.1 Member Enumeration Documentation

### 6.1523.1.1 Task

```
enum Digikam::VersionFileOperation::Task
```

#### Enumerator

NewFile	saveFile is a new file. Excludes Replace.
Replace	loadedFile and saveFile are the same - replace. Excludes NewFile.
SaveAndDelete	Similar to Replace, but the new file name differs from the old one, which should be removed.
MoveToIntermediate	Move loadedFile to loadedFileToIntermediate.
StoreIntermediates	Store additional snapshots from within history.

## 6.1523.2 Constructor & Destructor Documentation

### 6.1523.2.1 VersionFileOperation()

```
Digikam::VersionFileOperation::VersionFileOperation ( ) [explicit], [default]
```

This class describes an operation necessary for storing an image under version control. The loadedFile and current history is given to the [VersionManager](#). The saveFile is the destination of the save operation. If the loadedFile shall be moved to an intermediate, the name is given in intermediateForLoadedFile. The intermediates map may contain name of intermediates to save the state after action i of the history ( $\text{initialResolvedHistory.size()} \leq i < \text{currentHistory.size()} - 1$ ).

## 6.1523.3 Member Function Documentation

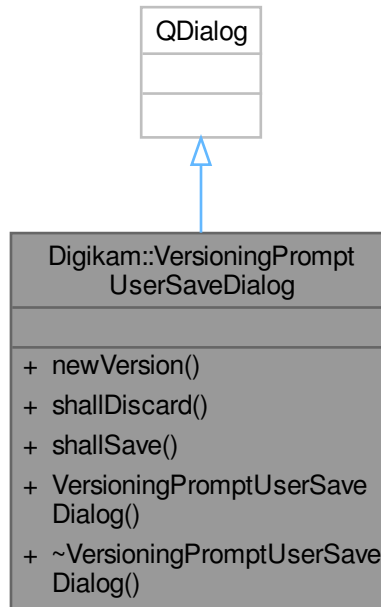
### 6.1523.3.1 allFilePaths()

```
QStringList Digikam::VersionFileOperation::allFilePaths ( ) const
```

Returns a list with all saving locations, for main result or intermediates

## 6.1524 Digikam::VersioningPromptUserSaveDialog Class Reference

Inheritance diagram for Digikam::VersioningPromptUserSaveDialog:



### Public Member Functions

- `bool newVersion () const`
- `bool shallDiscard () const`
- `bool shallSave () const`
- `VersioningPromptUserSaveDialog (QWidget *const parent)`

## 6.1525 Digikam::VersionItemFilterSettings Class Reference

### Public Member Functions

- `bool isExemptedBySettings (const ItemInfo &info) const`
- `bool isFiltering () const`  
*Returns if images will be filtered by these criteria at all.*
- `bool isFilteringByTags () const`  
*Returns if the tag is a filter criteria.*
- `bool isHiddenBySettings (const ItemInfo &info) const`
- `bool matches (const ItemInfo &info) const`
- `bool operator== (const VersionItemFilterSettings &other) const`
- `void setExceptionList (const QList< qulonglong > &idlist, const QString &id)`
- `void setVersionManagerSettings (const VersionManagerSettings &settings)`  
*— Tags filter —*
- `VersionItemFilterSettings (const VersionManagerSettings &settings)`

## Protected Attributes

- QHash< QString, QList< qlonglong > > **m\_exceptionLists**
- int **m\_exceptionTagFilter** = 0
- QList< int > **m\_excludeTagFilter**  
*DatabaseFields::Set watchFlags() const: Would return 0.*
- int **m\_includeTagFilter** = 0

## 6.1525.1 Member Function Documentation

### 6.1525.1.1 matches()

```
bool Digikam::VersionItemFilterSettings::matches (
    const ItemInfo & info ) const
```

Returns true if the given [ItemInfo](#) matches the filter criteria.

### 6.1525.1.2 setExceptionList()

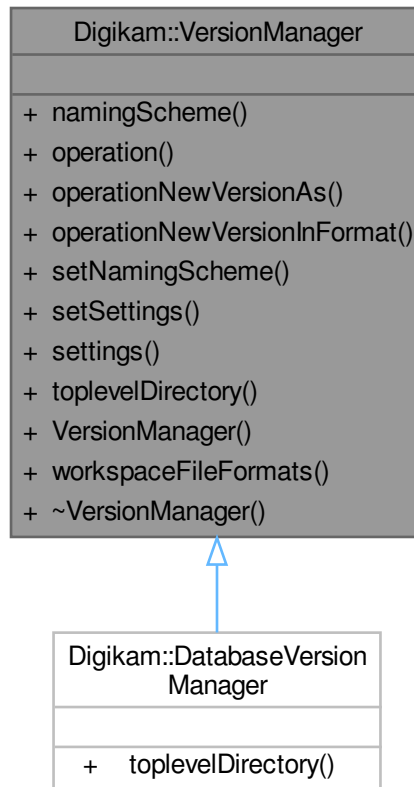
```
void Digikam::VersionItemFilterSettings::setExceptionList (
    const QList< qlonglong > & idlist,
    const QString & id )
```

Add list with exceptions: These images will be exempted from filtering by this filter



## 6.1526 Digikam::VersionManager Class Reference

Inheritance diagram for Digikam::VersionManager:



### Public Types

- enum **FileNameType** { **CurrentVersionName** , **NewVersionName** }

### Public Member Functions

- [VersionNamingScheme](#) \* **namingScheme** () const
- [VersionFileOperation](#) **operation** (FileNameType request, const [VersionFileInfo](#) &loadedFile, const [DImageHistory](#) &initialResolvedHistory, const [DImageHistory](#) &currentHistory)
- [VersionFileOperation](#) **operationNewVersionAs** (const [VersionFileInfo](#) &loadedFile, const [VersionFileInfo](#) &saveLocation, const [DImageHistory](#) &initialResolvedHistory, const [DImageHistory](#) &currentHistory)
- [VersionFileOperation](#) **operationNewVersionInFormat** (const [VersionFileInfo](#) &loadedFile, const QString &format, const [DImageHistory](#) &initialResolvedHistory, const [DImageHistory](#) &currentHistory)
- void **setNamingScheme** ([VersionNamingScheme](#) \*scheme)
- void **setSettings** (const [VersionManagerSettings](#) &settings)
- [VersionManagerSettings](#) **settings** () const
- virtual QString **toplevelDirectory** (const QString &path)
- virtual QStringList **workspaceFileFormats** () const

## 6.1527 Digikam::VersionManagerSettings Class Reference

### Public Types

- enum **EditorClosingMode** { **AlwaysAsk** , **AutoSave** }
- enum **IntermediateSavepoint** { **NoIntermediates** = 0 , **AfterEachSession** = 1 << 0 , **AfterRawConversion** = 1 << 1 , **WhenNotReproducible** = 1 << 2 }
- enum **ShowInViewFlag** { **OnlyShowCurrent** = 0 , **ShowOriginal** = 1 << 0 , **ShowIntermediates** = 1 << 1 }

### Public Member Functions

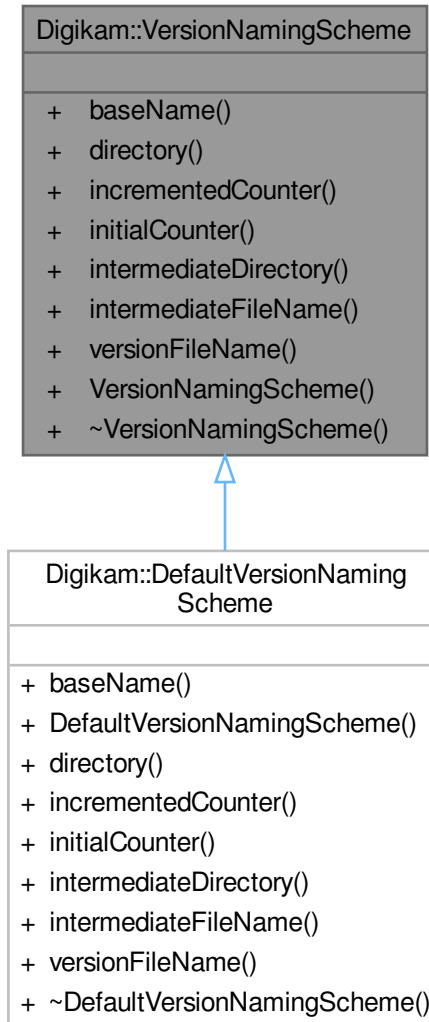
- void **readFromConfig** (const KConfigGroup &group)
- void **writeToConfig** (KConfigGroup &group) const

### Public Attributes

- EditorClosingMode **editorClosingMode** = AlwaysAsk
- bool **enabled** = true
- QString **format** = QLatin1String("JPG")  
*Image format string as defined for database, in upper case.*
- IntermediateBehavior **saveIntermediateVersions** = NoIntermediates
- ShowInViewFlags **showInViewFlags** = ShowOriginal

## 6.1528 Digikam::VersionNamingScheme Class Reference

Inheritance diagram for Digikam::VersionNamingScheme:



### Public Member Functions

- virtual QString [baseName](#) (const QString &path, const QString &filename, QVariant \*counter=nullptr, QVariant \*intermediateCounter=nullptr)=0
- virtual QString [directory](#) (const QString &path, const QString &filename)=0
- virtual QVariant [incrementedCounter](#) (const QVariant &counter)=0
- virtual QVariant [initialCounter](#) ()=0
- virtual QString [intermediateDirectory](#) (const QString &currentPath, const QString &fileName)=0
- virtual QString [intermediateFileName](#) (const QString &path, const QString &filename, const QVariant &version, const QVariant &counter)=0
- virtual QString [versionFileName](#) (const QString &path, const QString &baseName, const QVariant &counter)=0
- [VersionNamingScheme](#) ()=default

## 6.1528.1 Constructor & Destructor Documentation

### 6.1528.1.1 VersionNamingScheme()

```
Digikam::VersionNamingScheme::VersionNamingScheme ( ) [default]
```

Creates and analyzes file names of versioned files.

## 6.1528.2 Member Function Documentation

### 6.1528.2.1 baseName()

```
virtual QString Digikam::VersionNamingScheme::baseName (
    const QString & path,
    const QString & filename,
    QVariant * counter = nullptr,
    QVariant * intermediateCounter = nullptr ) [pure virtual]
```

Analyzes the given file name. Returns the basename in the sense of stripping the file name of all added version information: A scheme that appends a number, like "MyFile-1.jpg", shall return "MyFile". Path is the directory, filename the file name, so path + filename is the file path. If counter is given, and the given file name has a version number, write it to counter. If intermediateCounter is given, and the given file name has an intermediate counter number, write it to counter. If not available, do not touch the given counters. See [initialCounter\(\)](#) for the valid counter formats.

Implemented in [Digikam::DefaultVersionNamingScheme](#).

### 6.1528.2.2 directory()

```
virtual QString Digikam::VersionNamingScheme::directory (
    const QString & path,
    const QString & filename ) [pure virtual]
```

For a loaded file in directory path and with file name filename, returns the directory in which a new version (a new intermediate version, resp.) shall be stored.

Implemented in [Digikam::DefaultVersionNamingScheme](#).

### 6.1528.2.3 incrementedCounter()

```
virtual QVariant Digikam::VersionNamingScheme::incrementedCounter (
    const QVariant & counter ) [pure virtual]
```

Returns the given counter "incremented", that is, changed in a steady, repeatable fashion. You shall never return the given counter.

Implemented in [Digikam::DefaultVersionNamingScheme](#).

#### 6.1528.2.4 initialCounter()

```
virtual QVariant Digikam::VersionNamingScheme::initialCounter ( ) [pure virtual]
```

Returns an initial counter value for version and intermediate number counters. There are two places where you shall generate counters You will receive the given QVariant in [incrementedCounter\(\)](#), [versionFileName\(\)](#) and [baseName\(\)](#), and you shall read a counter value from a generated file name in [baseName\(\)](#).

Implemented in [Digikam::DefaultVersionNamingScheme](#).

#### 6.1528.2.5 intermediateFileName()

```
virtual QString Digikam::VersionNamingScheme::intermediateFileName (
    const QString & path,
    const QString & filename,
    const QVariant & version,
    const QVariant & counter ) [pure virtual]
```

Creates a version file name for an intermediate file in given directory, as previously returned by [directory\(\)](#), given baseName, as previously returned by [baseName\(\)](#), version and intermediate number counter. Do not append a file suffix. You do not need to check if the file exists.

Implemented in [Digikam::DefaultVersionNamingScheme](#).

#### 6.1528.2.6 versionFileName()

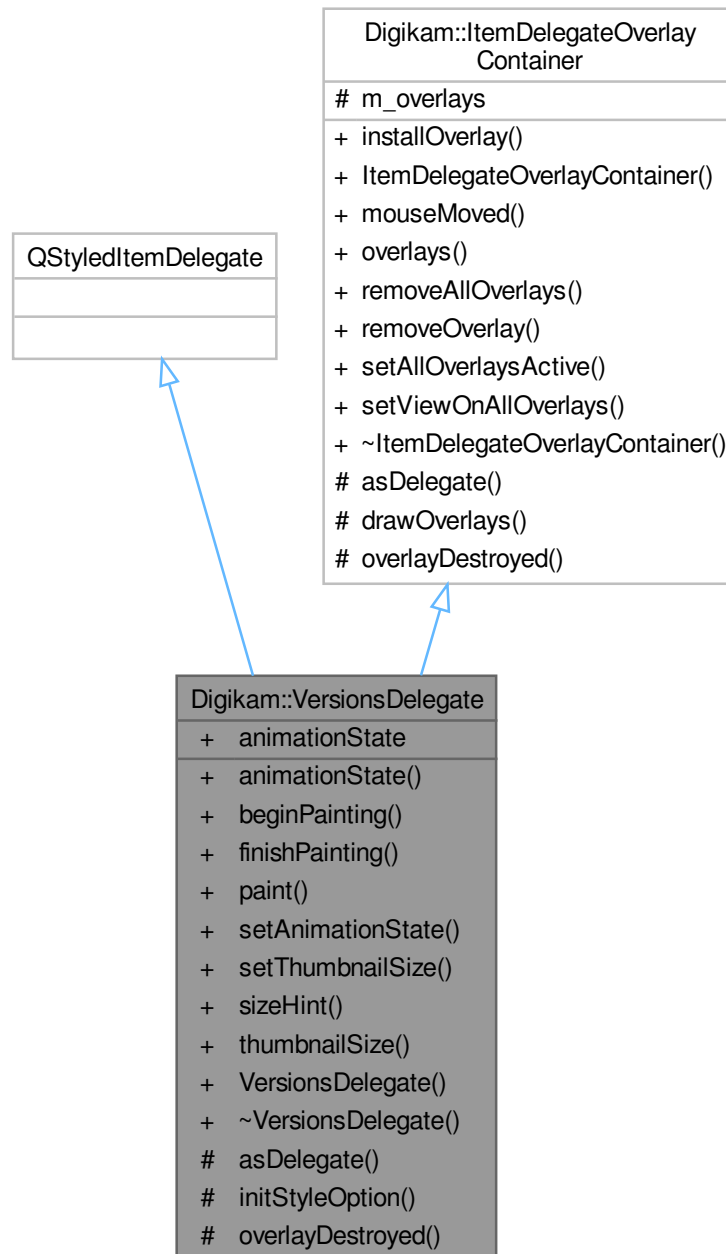
```
virtual QString Digikam::VersionNamingScheme::versionFileName (
    const QString & path,
    const QString & baseName,
    const QVariant & counter ) [pure virtual]
```

Creates a version file name for a file in given directory, as previously returned by [directory\(\)](#), given baseName, as previously returned by [baseName\(\)](#), and version counter. Do not append a file suffix. You do not need to check if the file exists.

Implemented in [Digikam::DefaultVersionNamingScheme](#).

## 6.1529 Digikam::VersionsDelegate Class Reference

Inheritance diagram for Digikam::VersionsDelegate:



### Signals

- void **animationStateChanged** ()
- void **hideNotification** ()

- void **requestNotification** (const QModelIndex &index, const QString &message)  
*NOTE: for [ItemDelegateOverlayContainer](#), unimplemented:*
- void **visualChange** ()

### Public Member Functions

- int **animationState** () const
- void **beginPainting** ()
- void **finishPainting** ()
- void **paint** (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- void **setAnimationState** (int animationState)
- void **setThumbnailSize** (int size) const
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int **thumbnailSize** () const
- **VersionsDelegate** (QObject \*const parent)

### Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

### Protected Slots

- void **overlayDestroyed** (QObject \*o) override

### Protected Member Functions

- QAbstractItemDelegate \* **asDelegate** () override  
*Returns the delegate, typically, the derived class.*
- void **initStyleOption** (QStyleOptionViewItem \*option, const QModelIndex &index) const override

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)  
*Declare as slot in the derived class calling this method.*

### Properties

- int **animationState**

### Additional Inherited Members

### Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- `QList< ItemDelegateOverlay * > m_overlays`

## 6.1529.1 Member Function Documentation

### 6.1529.1.1 `asDelegate()`

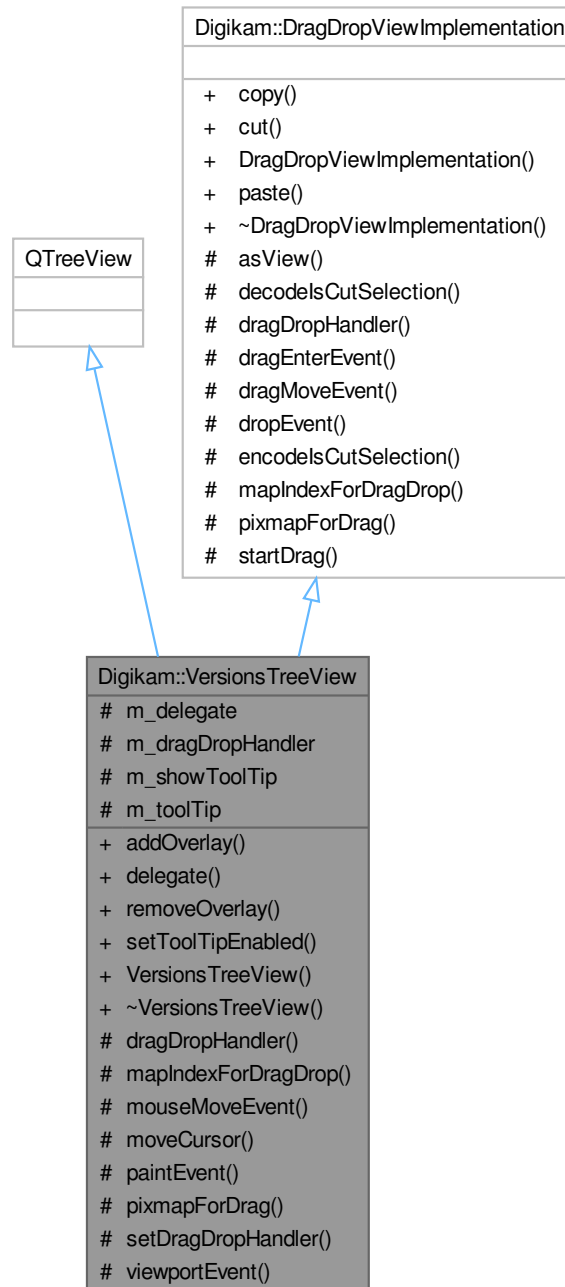
```
QAbstractItemDelegate * Digikam::VersionsDelegate::asDelegate ( ) [inline], [override], [protected],  
[virtual]
```

Implements [Digikam::ItemDelegateOverlayContainer](#).



## 6.1530 Digikam::VersionsTreeView Class Reference

Inheritance diagram for Digikam::VersionsTreeView:



### Public Member Functions

- void **addOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [VersionsDelegate](#) \* **delegate** () const

- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setToolTipEnabled** (bool on)
- **VersionsTreeView** (QWidget \*const parent=nullptr)
- **~VersionsTreeView** () override

*NOTE: All overlay management code in a sophisticated form can be studied in [ItemCategorizedView](#).*

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

## Protected Member Functions

- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*e) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- virtual void **setDragDropHandler** ([AbstractItemDragDropHandler](#) \*handler)
- bool **viewportEvent** (QEvent \*event) override

## Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual [QAbstractItemView](#) \* **asView** ()=0
- bool **decodelsCutSelection** (const [QMimeData](#) \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** ([QMimeData](#) \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

## Protected Attributes

- [VersionsDelegate](#) \* **m\_delegate** = nullptr
- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr
- bool **m\_showToolTip** = false
- [ToolTip](#) \* **m\_toolTip** = nullptr

## 6.1530.1 Member Function Documentation

### 6.1530.1.1 dragDropHandler()

```
AbstractItemDragDropHandler * Digikam::VersionsTreeView::dragDropHandler ( ) const [override],
[protected], [virtual]
```

You need to implement these three methods Returns the drag drop handler.

Implements [Digikam::DragDropViewImplementation](#).

### 6.1530.1.2 mapIndexForDragDrop()

```
QModelIndex Digikam::VersionsTreeView::mapIndexForDragDrop (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Maps the given index of the view's model to an index of the handler's model, which can be a source model of the view's model.

Implements [Digikam::DragDropViewImplementation](#).

### 6.1530.1.3 pixmapForDrag()

```
QPixmap Digikam::VersionsTreeView::pixmapForDrag (
    const QList< QModelIndex > & indexes ) const [override], [protected], [virtual]
```

Creates a pixmap for dragging the given indexes.

Implements [Digikam::DragDropViewImplementation](#).

## 6.1531 Digikam::VersionsWidget Class Reference

Inheritance diagram for Digikam::VersionsWidget:



### Public Slots

- void **setCurrentItem** (const [ItemInfo](#) &info)

### Signals

- void **imageSelected** (const [ItemInfo](#) &info)

### Public Member Functions

- [ActionVersionsOverlay](#) \* **addActionOverlay** (const QIcon &icon, const QString &text, const QString &tip=QString())
- [ShowHideVersionsOverlay](#) \* **addShowHideOverlay** ()
- [VersionsDelegate](#) \* **delegate** () const
- void **readSettings** (const KConfigGroup &group)
- **VersionsWidget** (QWidget \*const parent=nullptr)
- [VersionsTreeView](#) \* **view** () const
- void **writeSettings** (KConfigGroup &group)

### Protected Slots

- void **slotSetupChanged** ()
- void **slotViewCurrentChanged** (const QModelIndex &current, const QModelIndex &previous)
- void **slotViewModeChanged** (int mode)

## 6.1532 Digikam::VideoFrame Class Reference

### Public Member Functions

- **VideoFrame** (int width, int height, int lineSize)

### Public Attributes

- QVector< quint8 > **frameData**
- quint32 **height** = 0
- quint32 **lineSize** = 0
- quint32 **width** = 0

## 6.1533 Digikam::VideoInfoContainer Class Reference

### Public Member Functions

- bool **isEmpty** () const
- bool **isNull** () const
- bool **operator==** (const [VideoInfoContainer](#) &t) const

### Public Attributes

- QString **aspectRatio**
- QString **audioBitRate**
- QString **audioChannelType**
- QString **audioCodec**
- QString **duration**
- QString **frameRate**
- QString **videoCodec**

## 6.1534 Digikam::VideoMetadataContainer Class Reference

### Public Attributes

- bool **allFieldsNull** = true
- QString **aspectRatio**
- QString **audioBitRate**
- QString **audioChannelType**
- QString **audioCodec**
- QString **duration**
- QString **frameRate**
- QString **videoCodec**

## 6.1535 Digikam::VideoStripFilter Class Reference

### Public Member Functions

- void **process** ([VideoFrame](#) &videoFrame)

## 6.1536 Digikam::VideoThumbDecoder Class Reference

### Classes

- class [Private](#)

### Public Member Functions

- bool **decodeVideoFrame** () const
- void **destroy** ()
- QString **getCodec** () const
- int **getDuration** () const
- int **getHeight** () const
- bool **getInitialized** () const
- void **getScaledVideoFrame** (int scaledSize, bool maintainAspectRatio, [VideoFrame](#) &videoFrame)
- int **getWidth** () const
- void **initialize** (const QString &filename)
- void **seek** (int timeInSeconds)
- **VideoThumbDecoder** (const QString &filename)

## 6.1537 Digikam::VideoThumbDecoder::Private Class Reference

### Public Member Functions

- void **convertAndScaleFrame** (AVPixelFormat format, int scaledSize, bool maintainAspectRatio, int &scaledWidth, int &scaledHeight)
- bool **decodeVideoPacket** () const
- void **deleteFilterGraph** ()
- bool **getVideoPacket** ()
- bool **initializeVideo** ()
- bool **processFilterGraph** (AVFrame \*const dst, const AVFrame \*const src, enum AVPixelFormat pixfmt, int width, int height)

### Public Attributes

- bool **allowSeek** = true
- AVFilterContext \* **bufferSinkContext** = nullptr
- AVFilterContext \* **bufferSourceContext** = nullptr
- AVFrame \* **filterFrame** = nullptr
- AVFilterGraph \* **filterGraph** = nullptr
- bool **initialized** = false
- int **lastHeight** = 0
- enum AVPixelFormat **lastPixfmt** = AV\_PIX\_FMT\_NONE
- int **lastWidth** = 0
- AVFormatContext \* **pFormatContext** = nullptr
- AVFrame \* **pFrame** = nullptr
- quint8 \* **pFrameBuffer** = nullptr
- AVPacket \* **pPacket** = nullptr
- AVCodec \* **pVideoCodec** = nullptr
- AVCodecContext \* **pVideoCodecContext** = nullptr
- AVCodecParameters \* **pVideoCodecParameters** = nullptr
- AVStream \* **pVideoStream** = nullptr
- int **videoStream** = -1

## 6.1538 Digikam::VideoThumbnailer Class Reference

### Public Member Functions

- void **addFilter** ([VideoStripFilter](#) \*const filter)
- void **clearFilters** ()
- void **generateThumbnail** (const QString &videoFile, QImage &image)
- void **removeFilter** (const [VideoStripFilter](#) \*const filter)
- void **setMaintainAspectRatio** (bool enabled)
- void **setSeekPercentage** (int percentage)
- void **setSeekTime** (const QString &seekTime)
- void **setSmartFrameSelection** (bool enabled)
- void **setThumbnailSize** (int size)
- void **setWorkAroundIssues** (bool workAround)
- **VideoThumbnailer** (int thumbnailSize, bool workaroundIssues, bool maintainAspectRatio, bool smart↔FrameSelection)

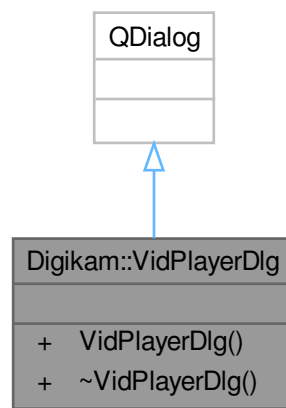
## 6.1539 Digikam::VideoThumbWriter Class Reference

### Public Member Functions

- void **writeFrame** ([VideoFrame](#) &frame, QImage &image)

## 6.1540 Digikam::VidPlayerDlg Class Reference

Inheritance diagram for Digikam::VidPlayerDlg:



### Public Member Functions

- **VidPlayerDlg** (const QString &file, QWidget \*const parent=nullptr)

## 6.1541 Digikam::VidSlideSettings Class Reference

### Public Types

- enum [Selection](#) { **IMAGES** = 0 , **ALBUMS** }
- enum [VidBitRate](#) { [VBR04](#) = 0 , [VBR05](#) , [VBR10](#) , [VBR12](#) , [VBR15](#) , [VBR20](#) , [VBR25](#) , [VBR30](#) , [VBR40](#) , [VBR45](#) , [VBR50](#) , [VBR60](#) , [VBR80](#) }
- enum [VidCodec](#) { [X264](#) = 0 , [MPEG4](#) , [MPEG2](#) , [MJPEG](#) , [FLASH](#) , [WEBMVP8](#) , [THEORA](#) , [WMV7](#) , [WMV8](#) , [WMV9](#) }
- enum [VidFormat](#) { [AVI](#) = 0 , [MKV](#) , [MP4](#) , [MPG](#) }
- enum [VidPlayer](#) { **NOPLAYER** = 0 , **INTERNAL** , **DESKTOP** }

- enum `VidStd` { `PAL = 0` , `NTSC` }
- enum `VidType` {  
`QVGA = 0` , `VCD1` , `VCD2` , `CVD1` ,  
`CVD2` , `HVGA` , `SVCD1` , `SDTV1` ,  
`SDTV2` , `EDTV1` , `SVCD2` , `EGA` ,  
`VGA` , `SDTV3` , `EDTV2` , `DVD1` ,  
`DVD2` , `WVGA` , `SVGA` , `DVGA` ,  
`XVGA` , `HDTV` , `WXGA1` , `WXGA2` ,  
`SXGA` , `SXGAPLUS` , `WSXGA` , `HDPLUS` ,  
`UXGA` , `WSXGAPLUS` , `BLUERAY` , `WUXGA` ,  
`TXGA` , `QXGA` , `UWFHD` , `WQHD` ,  
`WQXGA` , `QSXGA` , `QSXGAPLUS` , `WQXGAPLUS` ,  
`WQSXGA` , `QUXGA` , `UHD4K` , `WQUXGA` ,  
`HXGA` , `UHD5K` , `WHXGA` , `HSXGA` ,  
`UHD6K` , `WHSXGA` , `HUXGA` , `UHD8K` ,  
`WHUXGA` , `UW10K` , `UW16K` }

### Public Member Functions

- `QStringList defaultFFmpegSearchPaths () const`
- void `readSettings (const KConfigGroup &group)`
- int `videoBitRate () const`  
*Return the current video bit rate.*
- `QString videoCodec () const`  
*Return the current video ffmpeg codec name.*
- `QString videoFormat () const`  
*Return the current video format extension.*
- qreal `videoFrameRate () const`  
*Return the current video frame rate.*
- `QSize videoSize () const`  
*Return the current video size.*
- void `writeSettings (KConfigGroup &group)`

### Static Public Member Functions

- static bool `isVideoTVFormat (VidType type)`  
*Return true if type is a video TV format. If false is returned type is computer graphics screen format.*
- static `QMap< VidBitRate, QString > videoBitRateNames ()`
- static `QMap< VidCodec, QString > videoCodecNames ()`
- static `QMap< VidFormat, QString > videoFormatNames ()`
- static `QMap< VidPlayer, QString > videoPlayerNames ()`
- static `QSize videoSizeFromType (VidType type)`  
*Return the current size from a type of video.*
- static `QMap< VidStd, QString > videoStdNames ()`
- static `QMap< VidType, QString > videoTypeNames ()`



**Public Attributes**

- int **abitRate** = 64000  
*Encoded Audio stream bit rate in bit/s.*
- QString **audioTrack**  
*Soundtrack stream.*
- FileSaveConflictBox::ConflictRule **conflictRule** = FileSaveConflictBox::OVERWRITE
- bool **equalize** = false  
*Equalize filter to applying while encoding video from frames.*
- QMap< QString, QString > **ffmpegCodecs**  
*Map of FFmpeg codec names and features.*
- QMap< QString, QString > **ffmpegFormats**  
*Map of FFmpeg format names and features.*
- QString **ffmpegPath**  
*Path to FFmpeg binary.*
- QString **filesList**  
*Path to list of frame files to encode.*
- DInfoInterface \* **iface** = nullptr  
*Plugin host interface to handle item properties.*
- int **imgFrames** = 125  
*Amount of frames by image to encode in video (ex: 125 frames = 5 s at 25 img/s).*
- QList< QUrl > **inputImages**  
*Images stream.*
- FrameOsdSettings **osdSettings**  
*On Screen Display parameters.*
- QString **outputDir** = QStandardPaths::writableLocation(QStandardPaths::MoviesLocation)
- QString **outputFile**  
*Path to encoded video.*
- VidPlayer **outputPlayer** = INTERNAL  
*Open video stream in player at end.*
- QString **outputVideo**  
*Target video file encoded at end.*
- Selection **selMode** = IMAGES  
*Items selection mode.*
- QTime **soundtrackLength**  
*Duration of the soundtrack.*
- int **strength** = 5  
*Equalization strength factor.*
- QString **tempDir**  
*To store temporary frames.*
- TransitionMngr::TransType **transition** = TransitionMngr::None  
*Transition type between images.*
- VidBitRate **vbitRate** = VBR12  
*Encoded Video stream bit rate in bit/s.*
- VidCodec **vCodec** = X264  
*Encoded video codec.*
- EffectMngr::EffectType **vEffect** = EffectMngr::None  
*Encoded video effect while displaying images.*
- VidFormat **vFormat** = MP4  
*Encoded video container format.*
- VidStd **vStandard** = PAL  
*Encoded Video standard => frame rate in img/s.*
- VidType **vType** = BLUERAY  
*Encoded video type.*

## 6.1541.1 Member Enumeration Documentation

### 6.1541.1.1 Selection

enum `Digikam::VidSlideSettings::Selection`

Images selection mode

### 6.1541.1.2 VidBitRate

enum `Digikam::VidSlideSettings::VidBitRate`

Video rates in bits per seconds.

Enumerator

VBR04	400000
VBR05	500000
VBR10	1000000
VBR12	1200000
VBR15	1500000
VBR20	2000000
VBR25	2500000
VBR30	3000000
VBR40	4000000
VBR45	4500000
VBR50	5000000
VBR60	6000000
VBR80	8000000

### 6.1541.1.3 VidCodec

enum `Digikam::VidSlideSettings::VidCodec`

Video Codecs

Enumerator

X264	<a href="https://en.wikipedia.org/wiki/X264">https://en.wikipedia.org/wiki/X264</a>
MPEG4	<a href="https://en.wikipedia.org/wiki/MPEG-4">https://en.wikipedia.org/wiki/MPEG-4</a>
MPEG2	<a href="https://en.wikipedia.org/wiki/MPEG-2">https://en.wikipedia.org/wiki/MPEG-2</a>
MJPEG	<a href="https://en.wikipedia.org/wiki/Motion_JPEG">https://en.wikipedia.org/wiki/Motion_JPEG</a>
FLASH	<a href="https://en.wikipedia.org/wiki/Adobe_Flash">https://en.wikipedia.org/wiki/Adobe_Flash</a>
WEBMVP8	<a href="https://en.wikipedia.org/wiki/VP8">https://en.wikipedia.org/wiki/VP8</a>
THEORA	<a href="https://en.wikipedia.org/wiki/Theora">https://en.wikipedia.org/wiki/Theora</a>
WMV7	<a href="https://en.wikipedia.org/wiki/Windows_Media_Video">https://en.wikipedia.org/wiki/Windows_Media_Video</a>
WMV8	<a href="https://en.wikipedia.org/wiki/Windows_Media_Video">https://en.wikipedia.org/wiki/Windows_Media_Video</a>
WMV9	<a href="https://en.wikipedia.org/wiki/Windows_Media_Video">https://en.wikipedia.org/wiki/Windows_Media_Video</a>

### 6.1541.1.4 VidFormat

enum `Digikam::VidSlideSettings::VidFormat`

Video Container Formats

Enumerator

AVI	<a href="https://en.wikipedia.org/wiki/Audio_Video_Interleave">https://en.wikipedia.org/wiki/Audio_Video_Interleave</a>
MKV	<a href="https://en.wikipedia.org/wiki/Matroska">https://en.wikipedia.org/wiki/Matroska</a>
MP4	<a href="https://en.wikipedia.org/wiki/MPEG-4_Part_14">https://en.wikipedia.org/wiki/MPEG-4_Part_14</a>
MPG	<a href="https://en.wikipedia.org/wiki/MPEG-2">https://en.wikipedia.org/wiki/MPEG-2</a>

### 6.1541.1.5 VidPlayer

enum `Digikam::VidSlideSettings::VidPlayer`

Video player to use

### 6.1541.1.6 VidStd

enum `Digikam::VidSlideSettings::VidStd`

Video Standards

Enumerator

PAL	25 FPS
NTSC	29.97 FPS

### 6.1541.1.7 VidType

enum `Digikam::VidSlideSettings::VidType`

Video types (size of target screen) See [https://en.wikipedia.org/wiki/List\\_of\\_common\\_resolutions#Digital\\_TV\\_standards](https://en.wikipedia.org/wiki/List_of_common_resolutions#Digital_TV_standards) [https://en.wikipedia.org/wiki/Aspect\\_ratio\\_\(image\)](https://en.wikipedia.org/wiki/Aspect_ratio_(image))

Enumerator

QVGA	320 x 180 - 16:9 - Computer Graphics
VCD1	352 x 240 - 7:5 - Digital TV
VCD2	352 x 288 - 6:5 - Digital TV
CVD1	352 x 480 - 11:15 - Digital TV
CVD2	352 x 576 - 11:18 - Digital TV
HVGA	480 x 270 - 16:9 - Computer Graphics
SVCD1	480 x 480 - 1:1 - Digital TV
SDTV1	528 x 480 - 11:10 - Digital TV

## Enumerator

SDTV2	544 x 480 - 17:15 - Digital TV
EDTV1	544 x 576 - 17;18 - Digital TV
SVCD2	480 x 576 - 5:6 - Digital TV
EGA	640 x 350 - 16:9 - Computer Graphics
VGA	640 x 480 - 4:3 - Computer Graphics
SDTV3	704 x 480 - 22:15 - Digital TV
EDTV2	704 x 576 - 11;9 - Digital TV
DVD1	720 x 480 - 3:2 - Digital TV
DVD2	720 x 576 - 5:4 - Digital TV
WVGA	800 x 450 - 16:9 - Computer Graphics
SVGA	800 x 600 - 4:3 - Computer Graphics
DVGA	960 x 640 - 3:2 - Computer Graphics
XVGA	1024 x 576 - 16:9 - Computer Graphics
HDTV	1280 x 720 - 16:9 - Digital TV
WXGA1	1280 x 768 - 5:3 - Computer Graphics
WXGA2	1280 x 800 - 8:5 - Computer Graphics
SXGA	1280 x 1024 - 5:4 - Computer Graphics
SXGAPLUS	1400 x 1050 - 4:3 - Computer Graphics
WSXGA	1440 x 900 - 8:5 - Computer Graphics
HDPLUS	1600 x 900 - 16:9 - Digital TV
UXGA	1600 x 1200 - 4:3 - Computer Graphics
WSXGAPLUS	1680 x 1050 - 8:5 - Computer Graphics
BLUERAY	1920 x 1080 - 19:9 - Digital TV
WUXGA	1920 x 1200 - 8:5 - Computer Graphics
TXGA	1920 x 1440 - 7:5 - Computer Graphics
QXGA	2048 x 1536 - 4:3 - Computer Graphics
UWFHD	2560 < 1080 - 21:9 - Computer Graphics
WQHD	2560 x 1440 - 16:9 - Computer Graphics
WQXGA	2560 x 1600 - 8:5 - Computer Graphics
QSXGA	2560 x 2048 - 5:4 - Computer Graphics
QSXGAPLUS	2800 x 2100 - 4:3 - Computer Graphics
WQXGAPLUS	3200 x 1800 - 16:9 - Computer Graphics
WQSXGA	3200 x 2048 - 25:16 - Computer Graphics
QUXGA	3200 x 2400 - 4:3 - Computer Graphics
UHD4K	3840 x 2160 - 19:9 - Digital TV
WQUXGA	3840 x 2400 - 8:5 - Computer Graphics
HXGA	4096 x 3072 - 4:3 - Computer Graphics
UHD5K	5120 x 2880 - 16:9 - Computer Graphics
WHXGA	5120 x 3200 - 8:5 - Computer Graphics
HSXGA	5120 x 4096 - 5:4 - Computer Graphics
UHD6K	6016 x 3384 - 16:9 - Computer Graphics
WHSXGA	6400 x 4096 - 25:16 - Computer Graphics
HUXGA	6400 x 4800 - 4:3 - Computer Graphics
UHD8K	7680 x 4320 - 16:9 - Digital TV
WHUXGA	7680 x 4800 - 8:5 - Computer Graphics
UW10K	10240 x 4320 - 21:9 - Computer Graphics

## Enumerator

UW16K	15360 x 8640 - 16:9 - Computer Graphics
-------	---

## 6.1541.2 Member Function Documentation

### 6.1541.2.1 readSettings()

```
void Digikam::VidSlideSettings::readSettings (
    const KConfigGroup & group )
```

Read and write settings in config file between sessions.

### 6.1541.2.2 videoTypeNames()

```
QMap< VidSlideSettings::VidType, QString > Digikam::VidSlideSettings::videoTypeNames ( )
[static]
```

Helper methods to fill combobox from GUI.

## 6.1541.3 Member Data Documentation

### 6.1541.3.1 conflictRule

```
FileSaveConflictBox::ConflictRule Digikam::VidSlideSettings::conflictRule = FileSaveConflict↔
Box::OVERWRITE
```

[Rule](#) to follow if video file already exists.

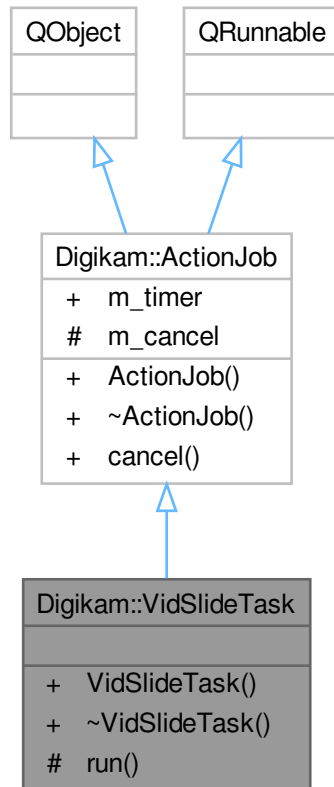
### 6.1541.3.2 outputDir

```
QString Digikam::VidSlideSettings::outputDir = QStandardPaths::writableLocation(QStandard↔
Paths::MoviesLocation)
```

Encoded video stream directory.

## 6.1542 Digikam::VidSlideTask Class Reference

Inheritance diagram for Digikam::VidSlideTask:



### Signals

- void **signalDone** (bool)
- void **signalMessage** (const QString &, bool)

### Signals inherited from [Digikam::ActionJob](#)

- void [signalDone](#) ()
- void [signalProgress](#) (int)
- void [signalStarted](#) ()

### Public Member Functions

- **VidSlideTask** ([VidSlideSettings](#) \*const settings)

## Public Member Functions inherited from [Digikam::ActionJob](#)

- [ActionJob](#) (QObject \*const parent=nullptr)
- [~ActionJob](#) () override

## Protected Member Functions

- void [run](#) () override

## Additional Inherited Members

## Public Slots inherited from [Digikam::ActionJob](#)

- void [cancel](#) ()

## Public Attributes inherited from [Digikam::ActionJob](#)

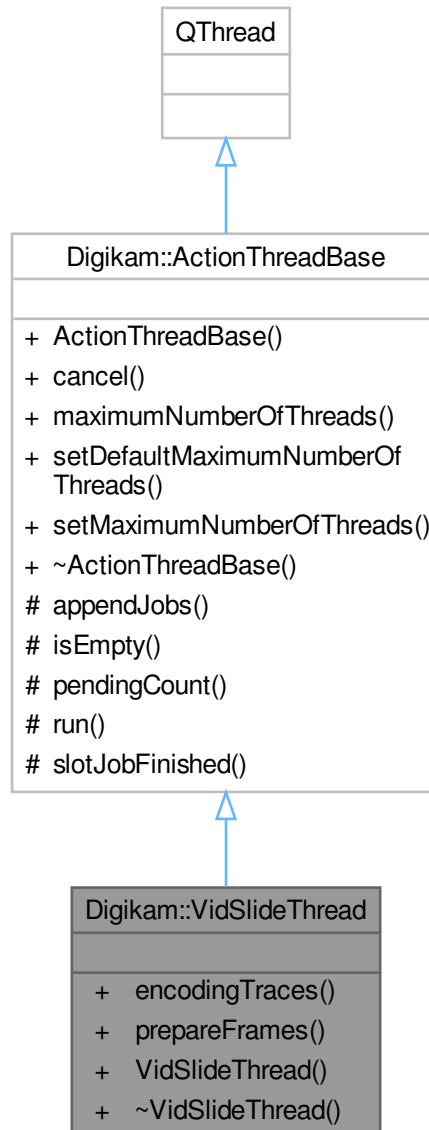
- QElapsedTimer [m\\_timer](#)

## Protected Attributes inherited from [Digikam::ActionJob](#)

- bool [m\\_cancel](#) = false

## 6.1543 Digikam::VidSlideThread Class Reference

Inheritance diagram for Digikam::VidSlideThread:



### Signals

- void **signalDone** (bool)
- void **signalMessage** (const QString &, bool)
- void **signalProgress** (int)



## Public Member Functions

- QString **encodingTraces** () const
- void **prepareFrames** (VidSlideSettings \*const settings)
- **VidSlideThread** (QObject \*const parent)

## Public Member Functions inherited from Digikam::ActionThreadBase

- **ActionThreadBase** (QObject \*const parent=nullptr)
- void **cancel** (bool isCancel=true)
- int **maximumNumberOfThreads** () const
- void **setDefaultMaximumNumberOfThreads** ()
- void **setMaximumNumberOfThreads** (int n)

## Additional Inherited Members

## Protected Slots inherited from Digikam::ActionThreadBase

- void **slotJobFinished** ()

## Protected Member Functions inherited from Digikam::ActionThreadBase

- void **appendJobs** (const ActionJobCollection &jobs)
- bool **isEmpty** () const
- int **pendingCount** () const
- void **run** () override

## 6.1543.1 Member Function Documentation

### 6.1543.1.1 prepareFrames()

```
void Digikam::VidSlideThread::prepareFrames (  
    VidSlideSettings *const settings )
```

Stage 1: prepare frames in temporary directory.

## 6.1544 Digikam::VisibilityController Class Reference

Inheritance diagram for Digikam::VisibilityController:



### Public Types

- enum **Status** {  
**Unknown** , **Hidden** , **Showing** , **Shown** ,  
**Hiding** }

### Public Slots

- void **hide** ()
- void **setVisible** (bool visible)  
*Shows/hides all added objects.*
- void **show** ()
- void **triggerVisibility** ()  
*Shows if hidden and hides if visible.*

## Public Member Functions

- void [addObject](#) ([VisibilityObject](#) \*const object)
- void [addWidget](#) (QWidget \*const widget)
- bool [isVisible](#) () const
- void [setContainerWidget](#) (QWidget \*const widget)
- **VisibilityController** (QObject \*const parent)

## Protected Member Functions

- void [allSteps](#) ()
- virtual void [beginStatusChange](#) ()
- void [step](#) ()

## 6.1544.1 Member Function Documentation

### 6.1544.1.1 addObject()

```
void Digikam::VisibilityController::addObject (
    VisibilityObject *const object )
```

Add an object implementing the [VisibilityObject](#) interface. You can use this if you have your widgets grouped in intermediate objects.

### 6.1544.1.2 addWidget()

```
void Digikam::VisibilityController::addWidget (
    QWidget *const widget )
```

Add a widget to this controller.

### 6.1544.1.3 isVisible()

```
bool Digikam::VisibilityController::isVisible ( ) const
```

Returns true if the contained objects are visible or becoming visible.

### 6.1544.1.4 setContainerWidget()

```
void Digikam::VisibilityController::setContainerWidget (
    QWidget *const widget )
```

Set the widget containing the widgets added to this controller



## 6.1546.1 Member Data Documentation

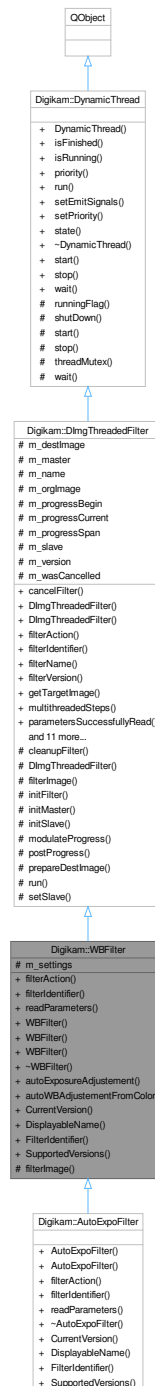
### 6.1546.1.1 black

```
double Digikam::WBContainer::black = 0.0
```

Neutral color temperature settings.

## 6.1547 Digikam::WBFilter Class Reference

Inheritance diagram for Digikam::WBFilter:



### Public Member Functions

- [FilterAction filterAction \(\)](#) override
- [QString filterIdentifier \(\)](#) const override

- void [readParameters](#) (const [FilterAction](#) &action) override
- **WBFilter** (const [WBContainer](#) &settings, [DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100)
- **WBFilter** ([DImg](#) \*const orgImage, QObject \*const parent=nullptr, const [WBContainer](#) &settings=[WBContainer](#)())
- **WBFilter** (QObject \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cancelFilter](#) ()
- [DImgThreadedFilter](#) ([DImg](#) \*const orgImage, QObject \*const parent, const QString &name=QString())
- [DImgThreadedFilter](#) (QObject \*const parent=nullptr, const QString &name=QString())
- const QString & **filterName** ()
- int **filterVersion** () const
- [DImg](#) **getTargetImage** ()
- QList< int > **multithreadedSteps** (int stop, int start=0) const
- virtual bool **parametersSuccessfullyRead** () const
- virtual QString **readParametersError** (const [FilterAction](#) &actionThatFailed) const
- void **setFilterName** (const QString &name)
- void **setFilterVersion** (int version)
- void **setOriginalImage** (const [DImg](#) &orgImage)
- void **setupAndStartDirectly** (const [DImg](#) &orgImage, [DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- void **setupFilter** (const [DImg](#) &orgImage)
- virtual void **startFilter** ()
- virtual void **startFilterDirectly** ()
- virtual QList< int > **supportedVersions** () const

### Public Member Functions inherited from [Digikam::DynamicThread](#)

- [DynamicThread](#) (QObject \*const parent=nullptr)
- bool **isFinished** () const
- bool **isRunning** () const
- QThread::Priority **priority** () const
- void **setEmitSignals** (bool emitThem)
- void **setPriority** (QThread::Priority priority)
- State **state** () const
- [~DynamicThread](#) () override

### Static Public Member Functions

- static void **autoExposureAdjustment** (const [DImg](#) \*const img, double &black, double &expo)
- static void **autoWBAdjustmentFromColor** (const QColor &tc, double &temperature, double &green)
- static int **CurrentVersion** ()
- static QString **DisplayableName** ()
- static QString **FilterIdentifier** ()
- static QList< int > **SupportedVersions** ()

### Protected Member Functions

- void **filterImage** () override

## Protected Member Functions inherited from [Digikam::DImgThreadedFilter](#)

- virtual void [cleanupFilter](#) ()
- [DImgThreadedFilter](#) ([DImgThreadedFilter](#) \*const master, const [DImg](#) &orgImage, const [DImg](#) &destImage, int progressBegin=0, int progressEnd=100, const QString &name=QString())
- virtual void [initFilter](#) ()
- void [initMaster](#) ()
- void [initSlave](#) ([DImgThreadedFilter](#) \*const master, int progressBegin=0, int progressEnd=100)
- virtual int [modulateProgress](#) (int progress)
- void [postProgress](#) (int progress)
- virtual void [prepareDestImage](#) ()
- void [run](#) () override
- void [setSlave](#) ([DImgThreadedFilter](#) \*const slave)

## Protected Member Functions inherited from [Digikam::DynamicThread](#)

- bool [runningFlag](#) () const volatile
- void [shutDown](#) ()
- void [start](#) (QMutexLocker< QMutex > &locker)
- void [stop](#) (const QMutexLocker< QMutex > &locker)
- QMutex \* [threadMutex](#) () const
- void [wait](#) (QMutexLocker< QMutex > &locker)

## Protected Attributes

- [WBContainer](#) [m\\_settings](#)

## Protected Attributes inherited from [Digikam::DImgThreadedFilter](#)

- [DImg](#) [m\\_destImage](#)
- [DImgThreadedFilter](#) \* [m\\_master](#) = nullptr
- QString [m\\_name](#)
- [DImg](#) [m\\_orgImage](#)
- int [m\\_progressBegin](#) = 0
- int [m\\_progressCurrent](#) = 0
  - To prevent signals bombarding with progress indicator value in [postProgress\(\)](#).*
- int [m\\_progressSpan](#) = 0
- [DImgThreadedFilter](#) \* [m\\_slave](#) = nullptr
- int [m\\_version](#) = 1
- bool [m\\_wasCancelled](#) = false

## Additional Inherited Members

## Public Types inherited from [Digikam::DynamicThread](#)

- enum [State](#) { [Inactive](#) , [Scheduled](#) , [Running](#) , [Deactivating](#) }



## Public Slots inherited from [Digikam::DynamicThread](#)

- void **start** ()
- void **stop** ()
- void **wait** ()

## Signals inherited from [Digikam::DImgThreadedFilter](#)

- void **finished** (bool success)
- void **progress** (int progress)
- void **started** ()

## Signals inherited from [Digikam::DynamicThread](#)

- void **finished** ()
- void **starting** ()

## 6.1547.1 Member Function Documentation

### 6.1547.1.1 `autoWBAdjustementFromColor()`

```
void Digikam::WBFilter::autoWBAdjustementFromColor (
    const QColor & tc,
    double & temperature,
    double & green ) [static]
```

This is a dichotomic search based on Blue and Red layers ratio to find the matching temperature adapted from `ufraw (0.12.1) RGB_to_Temperature`

### 6.1547.1.2 `filterAction()`

```
FilterAction Digikam::WBFilter::filterAction ( ) [override], [virtual]
```

Returns the action description corresponding to currently set options.

Implements [Digikam::DImgThreadedFilter](#).

### 6.1547.1.3 `filterIdentifier()`

```
QString Digikam::WBFilter::filterIdentifier ( ) const [inline], [override], [virtual]
```

Return the identifier for this filter in the image history.

Implements [Digikam::DImgThreadedFilter](#).

#### 6.1547.1.4 filterImage()

```
void Digikam::WBFilter::filterImage ( ) [override], [protected], [virtual]
```

Main image filter method. Override in subclass.

Implements [Digikam::DImgThreadedFilter](#).

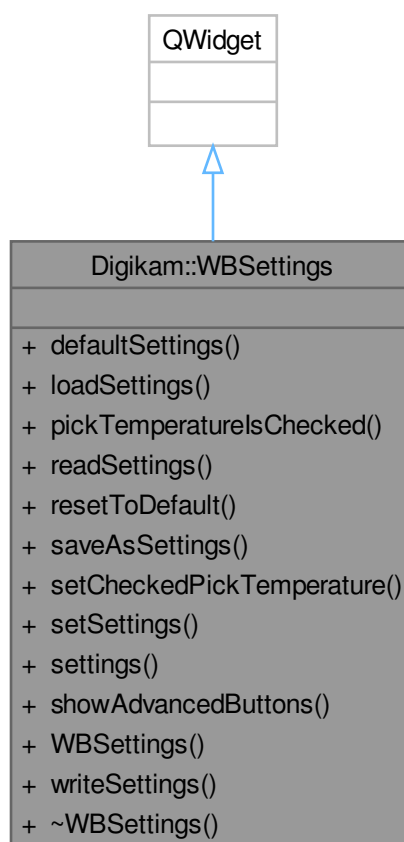
#### 6.1547.1.5 readParameters()

```
void Digikam::WBFilter::readParameters (
    const FilterAction & action ) [override], [virtual]
```

Implements [Digikam::DImgThreadedFilter](#).

## 6.1548 Digikam::WBSettings Class Reference

Inheritance diagram for Digikam::WBSettings:



## Signals

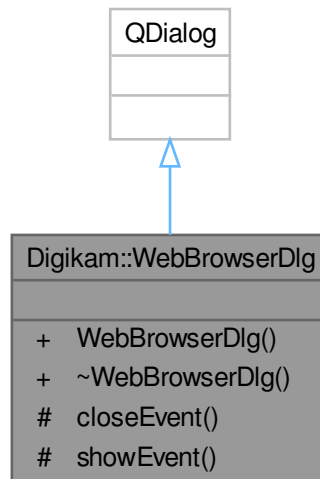
- void **signalAutoAdjustExposure** ()
- void **signalPickerColorButtonActivated** ()
- void **signalSettingsChanged** ()

## Public Member Functions

- [WBContainer](#) **defaultSettings** () const
- void **loadSettings** ()
- bool **pickTemperatureIsChecked** ()
- void **readSettings** (const KConfigGroup &group)
- void **resetToDefault** ()
- void **saveAsSettings** ()
- void **setCheckedPickTemperature** (bool b)
- void **setSettings** (const [WBContainer](#) &settings)
- [WBContainer](#) **settings** () const
- void **showAdvancedButtons** (bool b)
- **WBSettings** (QWidget \*const parent)
- void **writeSettings** (KConfigGroup &group)

## 6.1549 Digikam::WebBrowserDlg Class Reference

Inheritance diagram for Digikam::WebBrowserDlg:



## Signals

- void **closeView** (bool val)
- void **urlChanged** (const QUrl &url)

### Public Member Functions

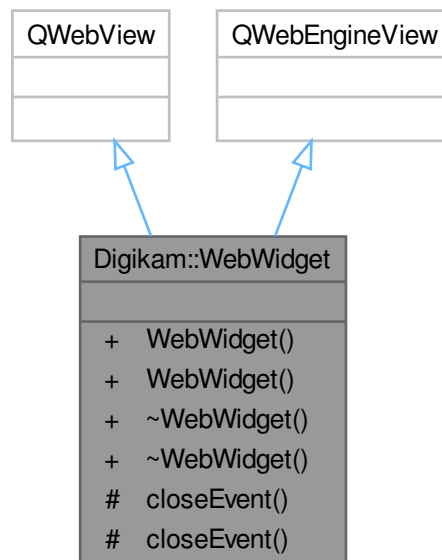
- **WebBrowserDlg** (const QUrl &url, QWidget \*const parent, bool hideDeskBrowser=false)

### Protected Member Functions

- void **closeEvent** (QCloseEvent \*) override
- void **showEvent** (QShowEvent \*) override

## 6.1550 Digikam::WebWidget Class Reference

Inheritance diagram for Digikam::WebWidget:



### Signals

- void **closeView** (bool val)
- void **closeView** (bool val)

### Public Member Functions

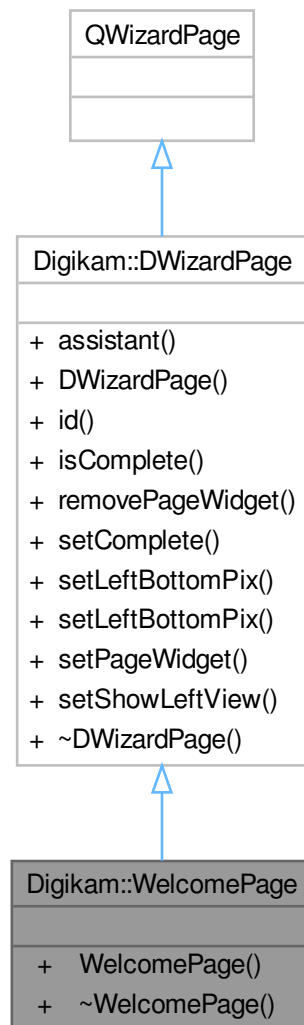
- **WebWidget** (QWidget \*const parent=nullptr)
- **WebWidget** (QWidget \*const parent=nullptr)

**Protected Member Functions**

- void **closeEvent** (QCloseEvent \*event) override
- void **closeEvent** (QCloseEvent \*event) override

**6.1551 Digikam::WelcomePage Class Reference**

Inheritance diagram for Digikam::WelcomePage:

**Public Member Functions**

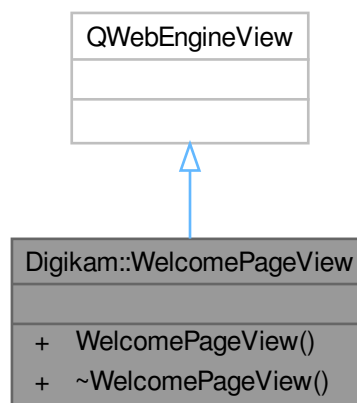
- **WelcomePage** (QWizard \*const dlg)

## Public Member Functions inherited from [Digikam::DWizardPage](#)

- `QWizard * assistant () const`
- `DWizardPage (QWizard *const dlg, const QString &title)`
- `int id () const`
- `bool isComplete () const` override
- `void removePageWidget (QWidget *const w)`
- `void setComplete (bool b)`
- `void setLeftBottomPix (const QIcon &icon)`
- `void setLeftBottomPix (const QPixmap &pix)`
- `void setPageWidget (QWidget *const w)`
- `void setShowLeftView (bool v)`

## 6.1552 Digikam::WelcomePageView Class Reference

Inheritance diagram for Digikam::WelcomePageView:

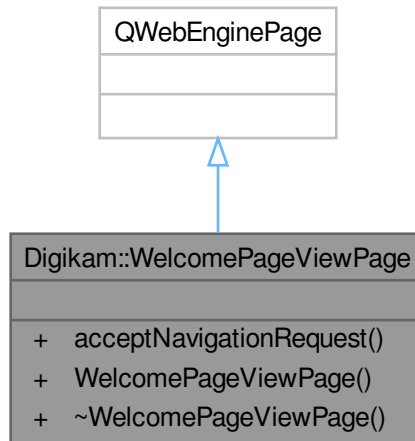


### Public Member Functions

- `WelcomePageView (QWidget *const parent)`

## 6.1553 Digikam::WelcomePageViewPage Class Reference

Inheritance diagram for Digikam::WelcomePageViewPage:



### Signals

- void **linkClicked** (const QUrl &)

### Public Member Functions

- bool **acceptNavigationRequest** (const QUrl &, QWebEnginePage::NavigationType, bool) override
- **WelcomePageViewPage** (QObject \*const parent=nullptr)





### Static Public Member Functions

- static bool **connectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method, Qt::ConnectionType type=Qt::AutoConnection)
- static bool **disconnectAndSchedule** (const QObject \*sender, const char \*signal, const [WorkerObject](#) \*receiver, const char \*method)

### Protected Member Functions

- virtual void [aboutToDeactivate](#) ()
- virtual void [aboutToQuitLoop](#) ()
- void **addRunnable** (WorkerObjectRunnable \*loop)
- bool **event** (QEvent \*e) override
- void **removeRunnable** (WorkerObjectRunnable \*loop)
- void **run** ()
- void **setEventLoop** (QEventLoop \*loop)
- void [shutDown](#) ()
- void **transitionToInactive** ()
- bool **transitionToRunning** ()

### Friends

- class **ThreadManager**
- class **WorkerObjectRunnable**

## 6.1554.1 Member Enumeration Documentation

### 6.1554.1.1 DeactivatingMode

```
enum Digikam::WorkerObject::DeactivatingMode
```

#### Enumerator

FlushSignals	Already sent signals are cleared.
KeepSignals	The thread is stopped, but already sent signals remain in the queue.
PhaseOut	The thread is stopped when all signals emitted until now have been processed.

## 6.1554.2 Constructor & Destructor Documentation

### 6.1554.2.1 WorkerObject()

```
Digikam::WorkerObject::WorkerObject ( ) [explicit]
```

Deriving from a worker object allows you to execute your slots in a thread. Implement any slots and connect signals just as usual. Call [schedule\(\)](#) before or when signals are emitted. The object will have moved to a thread when the signals are received by the slots. Call [deactivate\(\)](#) to stop computation. Note that without calling [schedule\(\)](#), no signal will ever be processed. You can use the connectAndSchedule convenience connection to avoid having to call [schedule\(\)](#) directly. Note that you cannot make this QObject the child of another QObject. Please check if you need to call shutDown from your destructor (see below).

## 6.1554.3 Member Function Documentation

### 6.1554.3.1 aboutToDeactivate()

```
void Digikam::WorkerObject::aboutToDeactivate ( ) [protected], [virtual]
```

Called from [deactivate\(\)](#), typically from a different thread than the worker thread, possibly the UI thread. You can stop any extra controlled threads here. Immediately afterwards, an event will be sent to the working thread which will cause the event loop to quit. ([aboutToQuitLoop\(\)](#))

Reimplemented in [Digikam::RecognitionWorker](#), and [Digikam::TrainerWorker](#).

### 6.1554.3.2 aboutToQuitLoop()

```
void Digikam::WorkerObject::aboutToQuitLoop ( ) [protected], [virtual]
```

Called from within thread's event loop to quit processing. Quit any blocking operation. Immediately afterwards, the event loop will be quit.

### 6.1554.3.3 connectAndSchedule()

```
bool Digikam::WorkerObject::connectAndSchedule (
    const QObject * sender,
    const char * signal,
    const char * method,
    Qt::ConnectionType type = Qt::AutoConnection ) const
```

You must normally call [schedule\(\)](#) to ensure that the object is active when you send a signal with work data. Instead, you can use these [connect\(\)](#) methods when connecting your signal to this object, the signal that carries work data. Then the object will be scheduled each time you emit the signal.

### 6.1554.3.4 deactivate

```
void Digikam::WorkerObject::deactivate (
    DeactivatingMode mode = FlushSignals ) [slot]
```

Quits execution of this worker object. If mode is `FlushSignals`, all already emitted signals will be cleared. If mode is `KeepSignals`, already emitted signals are not cleared and will be kept in the event queue until destruction or [schedule\(\)](#) is called. If mode is `PhaseOut`, already emitted signals will be processed and the thread quit immediately afterwards.

### 6.1554.3.5 schedule

```
void Digikam::WorkerObject::schedule ( ) [slot]
```

Starts execution of this worker object: The object is moved to a thread and an event loop started, so that queued signals will be received.

### 6.1554.3.6 setPriority()

```
void Digikam::WorkerObject::setPriority (
    QThread::Priority priority )
```

Sets the priority for this dynamic thread. Can be set anytime. If the thread is currently not running, the priority will be set when it is run next time. When you set QThread::InheritPriority (default), the priority is not changed but inherited from the thread pool.

### 6.1554.3.7 shutDown()

```
void Digikam::WorkerObject::shutDown ( ) [protected]
```

If you are deleting data in your destructor which is accessed from the thread, do one of the following from your destructor to guarantee a safe shutdown: 1) Call this method 2) Call stop() and wait(), knowing that nothing will call start() anymore after this 3) Be sure the thread will never be running at destruction. Note: This irrevocably stops this object. Note: It is not sufficient that your parent class does this. Calling this method, or providing one of the above mentioned equivalent guarantees, must be done by every single last class in the hierarchy with an implemented destructor deleting data. (the base class destructor is always called after the derived class)

## 6.1555 Digikam::Workflow Class Reference

### Public Attributes

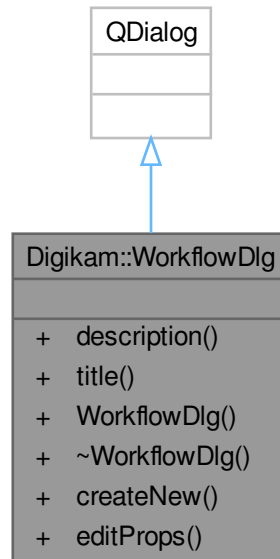
- [BatchSetList](#) **aTools**
- QString **desc**
- [QueueSettings](#) **qSettings**
- QString **title**

### 6.1555.1 Detailed Description

This container group all queue common settings plus all assigned batch tools.

## 6.1556 Digikam::WorkflowDlg Class Reference

Inheritance diagram for Digikam::WorkflowDlg:



### Public Member Functions

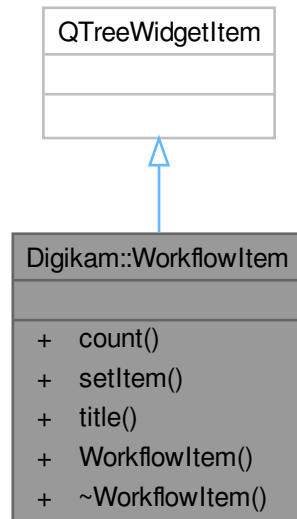
- `QString` **description** () const
- `QString` **title** () const
- **WorkflowDlg** (const [Workflow](#) &wf, bool create=false)

### Static Public Member Functions

- static bool **createNew** ([Workflow](#) &wf)
- static bool **editProps** ([Workflow](#) &wf)

## 6.1557 Digikam::WorkflowItem Class Reference

Inheritance diagram for Digikam::WorkflowItem:

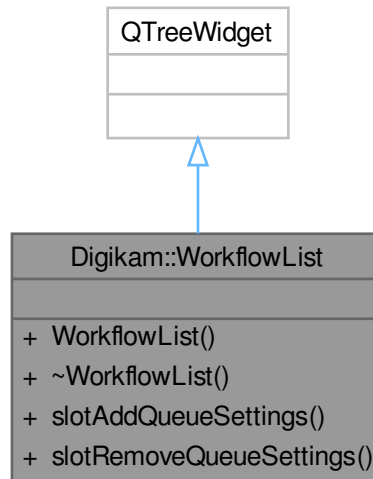


### Public Member Functions

- `int count () const`
- `void setItem (const QString &title=QString(), const QString &desc=QString(), int count=0)`
- `QString title () const`
- `WorkflowItem (WorkflowList *const parent, const QString &name)`

## 6.1558 Digikam::WorkflowList Class Reference

Inheritance diagram for Digikam::WorkflowList:



### Public Slots

- void **slotAddQueueSettings** (const QString &title)
- void **slotRemoveQueueSettings** (const QString &title)

### Signals

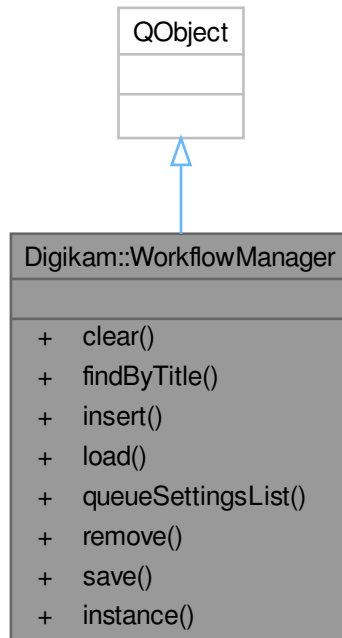
- void **signalAssignQueueSettings** (const QString &)
- void **signalUpdateQueueSettings** (const QString &)

### Public Member Functions

- **WorkflowList** (QWidget \*const parent)

## 6.1559 Digikam::WorkflowManager Class Reference

Inheritance diagram for Digikam::WorkflowManager:



### Signals

- void **signalQueueSettingsAdded** (const QString &)
- void **signalQueueSettingsRemoved** (const QString &)

### Public Member Functions

- void **clear** ()
- [Workflow](#) **findByTitle** (const QString &title) const
- void **insert** (const [Workflow](#) &q)
- bool **load** (QStringList &failed)
- QList< [Workflow](#) > **queueSettingsList** () const
- void **remove** (const [Workflow](#) &q)
- bool **save** ()

### Static Public Member Functions

- static [WorkflowManager](#) \* **instance** ()

**Friends**

- class **WorkflowManagerCreator**

**6.1559.1 Member Function Documentation****6.1559.1.1 load()**

```
bool Digikam::WorkflowManager::load (
    QStringList & failed )
```

Load all [Workflow](#) from XML settings file. Fill 'failed' list with incompatible [Workflow](#) title/description not loaded.

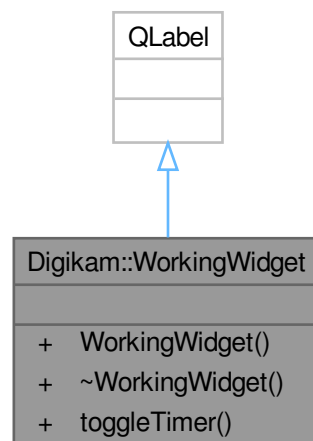
**6.1559.1.2 save()**

```
bool Digikam::WorkflowManager::save ( )
```

Save all [Workflow](#) to XML settings file.

**6.1560 Digikam::WorkingWidget Class Reference**

Inheritance diagram for Digikam::WorkingWidget:

**Public Slots**

- void **toggleTimer** (bool turnOn=false)



## Signals

- void **animationStep** ()

## Public Member Functions

- **WorkingWidget** (QWidget \*const parent=nullptr)

## 6.1561 Digikam::WSAlbum Class Reference

### Public Member Functions

- void **setBaseAlbum** (const [WSAlbum](#) &album)

### Public Attributes

- QString **description**
- QString **id**
- bool **isRoot** = true
- QString **location**
- QString **parentID**
- QString **title**
- bool **uploadable** = true
- QString **url**

### 6.1561.1 Member Function Documentation

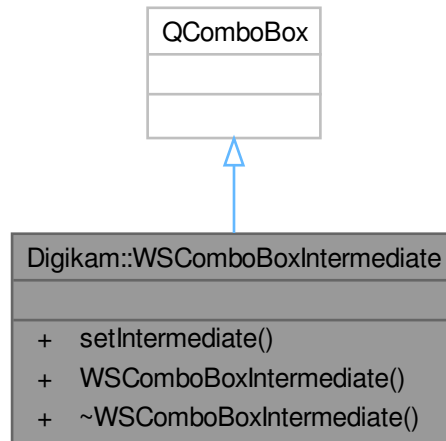
#### 6.1561.1.1 setBaseAlbum()

```
void Digikam::WSAlbum::setBaseAlbum (  
    const WSAlbum & album ) [inline]
```

This method is used by derived class of [WSAlbum](#), to set the attributes inherited from [WSAlbum](#), knowing a [WSAlbum](#).

## 6.1562 Digikam::WSComboBoxIntermediate Class Reference

Inheritance diagram for Digikam::WSComboBoxIntermediate:



### Public Member Functions

- void [setIntermediate](#) (bool)
- [WSComboBoxIntermediate](#) (QWidget \*const = nullptr, const QString &=QString())

### 6.1562.1 Constructor & Destructor Documentation

#### 6.1562.1.1 WSComboBoxIntermediate()

```

Digikam::WSComboBoxIntermediate::WSComboBoxIntermediate (
    QWidget * const parent = nullptr,
    const QString & text = QString() ) [explicit]
  
```

Initialize the combobox with a parent and a string to indicate the intermediate state.

### 6.1562.2 Member Function Documentation

#### 6.1562.2.1 setIntermediate()

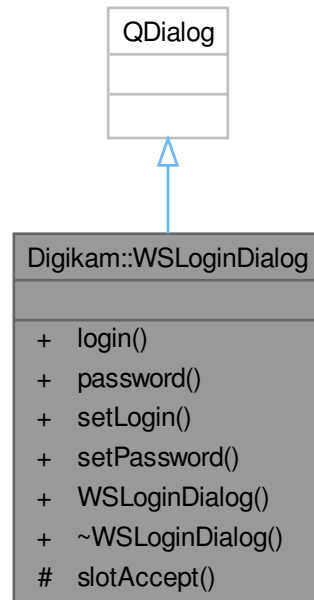
```

void Digikam::WSComboBoxIntermediate::setIntermediate (
    bool state )
  
```

Set the state of the combobox to intermediate. The intermediate state is 'unset' when another index is selected.

## 6.1563 Digikam::WSLoginDialog Class Reference

Inheritance diagram for Digikam::WSLoginDialog:



### Public Member Functions

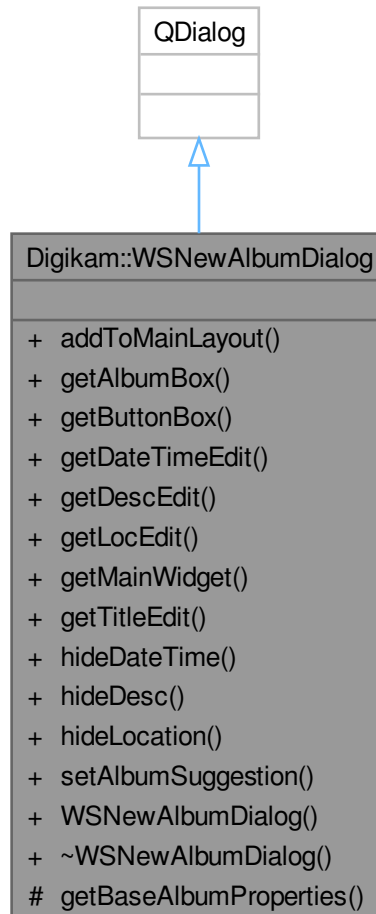
- `QString login () const`
- `QString password () const`
- `void setLogin (const QString &)`
- `void setPassword (const QString &)`
- `WSLoginDialog (QWidget *const parent, const QString &prompt, const QString &header=QString(), const QString &passwd=QString())`
- `~WSLoginDialog ()`

### Protected Slots

- `void slotAccept ()`

## 6.1564 Digikam::WSNewAlbumDialog Class Reference

Inheritance diagram for Digikam::WSNewAlbumDialog:



### Public Member Functions

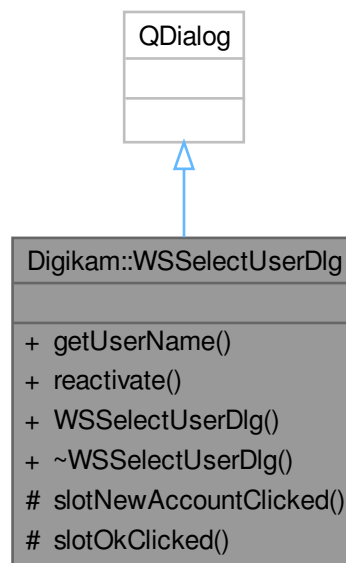
- void **addToMainLayout** (QWidget \*const widget)
- QGroupBox \* **getAlbumBox** () const
- QDialogButtonBox \* **getButtonBox** () const
- QDateTimeEdit \* **getDateTimeEdit** () const
- [DTextEdit](#) \* **getDescEdit** () const
- [DTextEdit](#) \* **getLocEdit** () const
- QWidget \* **getMainWidget** () const
- [DTextEdit](#) \* **getTitleEdit** () const
- void **hideDateTime** ()
- void **hideDesc** ()
- void **hideLocation** ()
- void **setAlbumSuggestion** (const QString &title)
- **WSNewAlbumDialog** (QWidget \*const parent, const QString &toolName)

### Protected Member Functions

- void **getBaseAlbumProperties** ([WSAlbum](#) &baseAlbum)

## 6.1565 Digikam::WSSelectUserDlg Class Reference

Inheritance diagram for Digikam::WSSelectUserDlg:



### Public Member Functions

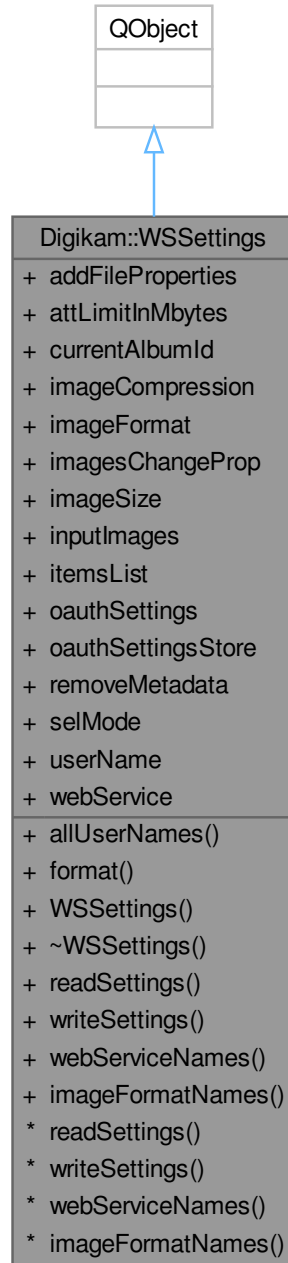
- `QString` **getUserName** () const
- void **reactivate** ()
- **WSSelectUserDlg** (`QWidget *const parent, const QString &serviceName`)

### Protected Slots

- void **slotNewAccountClicked** ()
- void **slotOkClicked** ()

## 6.1566 Digikam::WSSettings Class Reference

Inheritance diagram for Digikam::WSSettings:



### Public Types

- enum `ImageFormat` { `JPEG = 0` , `PNG` }
- enum `Selection` { `EXPORT = 0` , `IMPORT` }

*Images selection mode.*

- enum **WebService** {  
**FLICKR** = 0 , **DROPBOX** , **IMGUR** , **FACEBOOK** ,  
**SMUGMUG** , **GDRIVE** , **GPHOTO** }

### Public Member Functions

- QStringList **allUserNames** (const QString &serviceName)  
*Helper method to list all user accounts (of all web service) that user logged in before.*
- QString **format** () const
- **WSSettings** (QObject \*const parent=nullptr)
  
- void **readSettings** (const KConfigGroup &group)  
*Read and write settings in config file between sessions.*
- void **writeSettings** (KConfigGroup &group)

### Static Public Member Functions

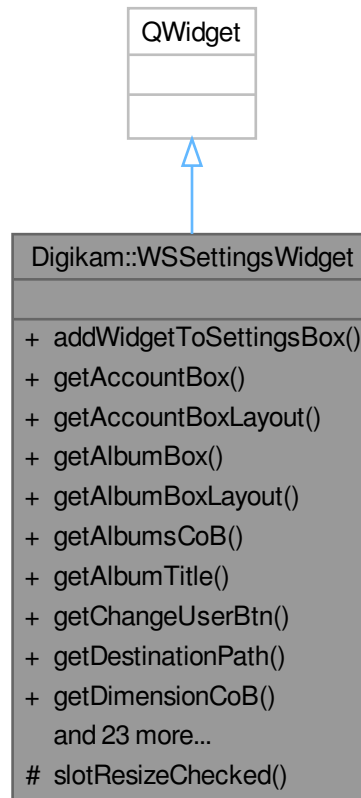
- static QMap< WebService, QString > **webServiceNames** ()  
*Helper methods to fill settings from GUI.*
- static QMap< ImageFormat, QString > **imageFormatNames** ()

### Public Attributes

- bool **addFileProperties** = false
- qint64 **attLimitInMbytes** = 17
- QString **currentAlbumId**  
*Selected album to upload to.*
- int **imageCompression** = 75
- ImageFormat **imageFormat** = JPEG
- bool **imagesChangeProp** = false
- int **imageSize** = 1024
- QList< QUrl > **inputImages**  
*Selected items to upload.*
- QMap< QUrl, QUrl > **itemsList**  
*Map of original item and attached item (can be resized).*
- QSettings \* **oauthSettings** = nullptr
- O0SettingsStore \* **oauthSettingsStore** = nullptr
- bool **removeMetadata** = false
- Selection **selMode** = EXPORT  
*Items selection mode.*
- QString **userName**
- WebService **webService** = FLICKR

## 6.1567 Digikam::WSSettingsWidget Class Reference

Inheritance diagram for Digikam::WSSettingsWidget:



### Public Member Functions

- void **addWidgetToSettingsBox** (QWidget \*const widget)
- QGroupBox \* **getAccountBox** () const
- QGridLayout \* **getAccountBoxLayout** () const
- QGroupBox \* **getAlbumBox** () const
- QGridLayout \* **getAlbumBoxLayout** () const
- QComboBox \* **getAlbumsCoB** () const
- QString **getAlbumTitle** () const
- QPushButton \* **getChangeUserBtn** () const
- QString **getDestinationPath** () const
- QComboBox \* **getDimensionCoB** () const
- QSpinBox \* **getDimensionSpB** () const
- QLabel \* **getHeaderLbl** () const
- QSpinBox \* **getImgQualitySpB** () const
- QPushButton \* **getNewAlbmBtn** () const
- QGroupBox \* **getOptionsBox** () const
- QGridLayout \* **getOptionsBoxLayout** () const



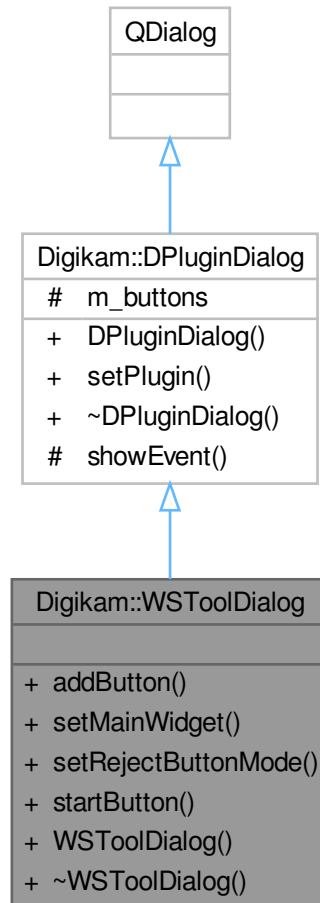
- QCheckBox \* **getOriginalCheckBox** () const
- QCheckBox \* **getPhotoldCheckBox** () const
- QPushButton \* **getReloadBtn** () const
- QCheckBox \* **getResizeCheckBox** () const
- QWidget \* **getSettingsBox** () const
- QVBoxLayout \* **getSettingsBoxLayout** () const
- QGroupBox \* **getSizeBox** () const
- QVBoxLayout \* **getSizeBoxLayout** () const
- QGroupBox \* **getUploadBox** () const
- QVBoxLayout \* **getUploadBoxLayout** () const
- QLabel \* **getUserNameLabel** () const
- [DItemsList](#) \* **imagesList** () const
- [DProgressWdg](#) \* **progressBar** () const
- void **replacelmageList** (QWidget \*const widget)
- virtual void **updateLabels** (const QString &name=QString(), const QString &url=QString())=0
- **WSSettingsWidget** (QWidget \*const parent, [DInfoInterface](#) \*const iface, const QString &toolName)

### Protected Slots

- void **slotResizeChecked** ()

## 6.1568 Digikam::WSToolDialog Class Reference

Inheritance diagram for Digikam::WSToolDialog:



### Signals

- void **cancelClicked** ()

### Public Member Functions

- void **addButton** (QAbstractButton \*button, QDialogButtonBox::ButtonRole role)
- void **setMainWidget** (QWidget \*const widget)
- void **setRejectButtonMode** (QDialogButtonBox::StandardButton button)
- QPushButton \* **startButton** () const
- **WSToolDialog** (QWidget \*const parent, const QString &objName)

## Public Member Functions inherited from [Digikam::DPluginDialog](#)

- **DPluginDialog** (QWidget \*const parent, const QString &objName)
- void **setPlugin** ([DPlugin](#) \*const tool)

## Additional Inherited Members

## Protected Member Functions inherited from [Digikam::DPluginDialog](#)

- void **showEvent** (QShowEvent \*) override

## Protected Attributes inherited from [Digikam::DPluginDialog](#)

- QDialogButtonBox \* **m\_buttons** = nullptr

# 6.1569 Digikam::WSToolUtils Class Reference

## Static Public Member Functions

- static void **clearToken** (const QString &name)
- static QString **decodeKey** (const QString &key)
- static QSettings \* **getOauthSettings** (QObject \*const parent)
- static QDir **makeTemporaryDir** (const char \*prefix)
- static QString **randomString** (const int &length)
- static QString **readToken** (const QString &name)
- static void **removeTemporaryDir** (const char \*prefix)
- static void **saveToken** (const QString &name, const QString &token)

## 6.1569.1 Member Function Documentation

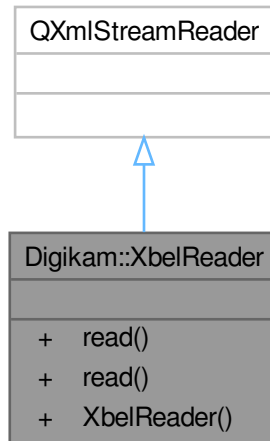
### 6.1569.1.1 randomString()

```
QString Digikam::WSToolUtils::randomString (
    const int & length ) [static]
```

Generates random string.

## 6.1570 Digikam::XbelReader Class Reference

Inheritance diagram for Digikam::XbelReader:



### Public Member Functions

- [BookmarkNode](#) \* **read** (const QString &fileName)
- [BookmarkNode](#) \* **read** (QIODevice \*const device, bool addRootFolder=false)

## 6.1571 Digikam::XbelWriter Class Reference

Inheritance diagram for Digikam::XbelWriter:

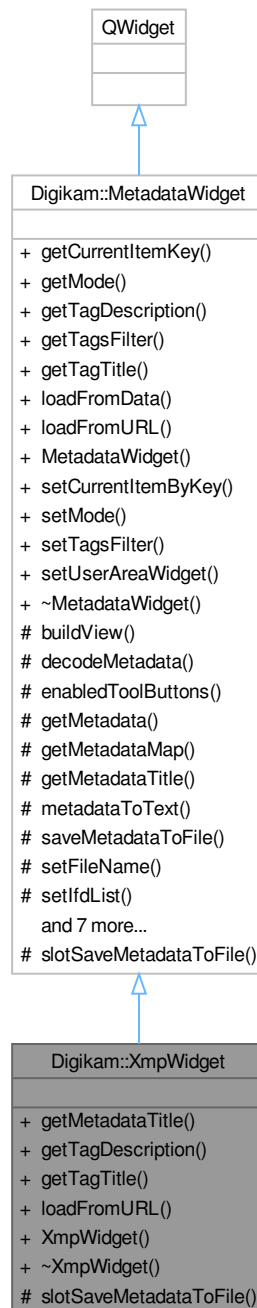


## Public Member Functions

- bool **write** (const QString &fileName, const [BookmarkNode](#) \*const root)
- bool **write** (QIODevice \*const device, const [BookmarkNode](#) \*const root)

## 6.1572 Digikam::XmpWidget Class Reference

Inheritance diagram for Digikam::XmpWidget:



**Public Member Functions**

- QString [getMetadataTitle](#) () const override
- QString [getTagDescription](#) (const QString &key) override
- QString [getTagTitle](#) (const QString &key) override
- bool [loadFromURL](#) (const QUrl &url) override
- **XmpWidget** (QWidget \*const parent, const QString &name=QString())

**Public Member Functions inherited from [Digikam::MetadataWidget](#)**

- QString [getCurrentItemKey](#) () const
- int [getMode](#) () const
- QStringList [getTagsFilter](#) () const
- virtual bool [loadFromData](#) (const QString &fileName, const [DMetadata](#) &data=[DMetadata](#)())
- **MetadataWidget** (QWidget \*const parent, const QString &name=QString())
- void [setCurrentItemByKey](#) (const QString &itemKey)
- void [setMode](#) (int mode)
- void [setTagsFilter](#) (const QStringList &list)
- void [setUserAreaWidget](#) (QWidget \*const w)

**Protected Slots**

- void [slotSaveMetadataToFile](#) () override

**Protected Slots inherited from [Digikam::MetadataWidget](#)**

- virtual void [slotSaveMetadataToFile](#) ()=0

**Additional Inherited Members****Public Types inherited from [Digikam::MetadataWidget](#)**

- enum [TagFilters](#) { **NONE** = 0 , **PHOTO** , **CUSTOM** }

**Signals inherited from [Digikam::MetadataWidget](#)**

- void [signalSetupMetadataFilters](#) ()

**Protected Member Functions inherited from [Digikam::MetadataWidget](#)**

- void [enabledToolButtons](#) (bool)
- [DMetadata](#) \* [getMetadata](#) () const
- const [DMetadata::MetaDataMap](#) & [getMetadataMap](#) ()
- QString [metadataToText](#) () const
- QUrl [saveMetadataToFile](#) (const QString &caption, const QString &fileFilter)
- void [setFileName](#) (const QString &fileName)
- void [setIfdList](#) (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &keysFilter, const QStringList &tagsFilter)
- void [setIfdList](#) (const [DMetadata::MetaDataMap](#) &ifds, const QStringList &tagsFilter=QStringList())
- bool [setMetadata](#) (const [DMetadata](#) &data=[DMetadata](#)())
- virtual void [setMetadataEmpty](#) ()
- void [setMetadataMap](#) (const [DMetadata::MetaDataMap](#) &data=[DMetadata::MetaDataMap](#)())
- void [setup](#) ()
- bool [storeMetadataToFile](#) (const QUrl &url, const QByteArray &metaData)
- [MetadataListView](#) \* [view](#) () const

## 6.1572.1 Member Function Documentation

### 6.1572.1.1 getMetadataTitle()

```
QString Digikam::XmpWidget::getMetadataTitle ( ) const [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

### 6.1572.1.2 getTagDescription()

```
QString Digikam::XmpWidget::getTagDescription (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

### 6.1572.1.3 getTagTitle()

```
QString Digikam::XmpWidget::getTagTitle (
    const QString & key ) [override], [virtual]
```

Reimplemented from [Digikam::MetadataWidget](#).

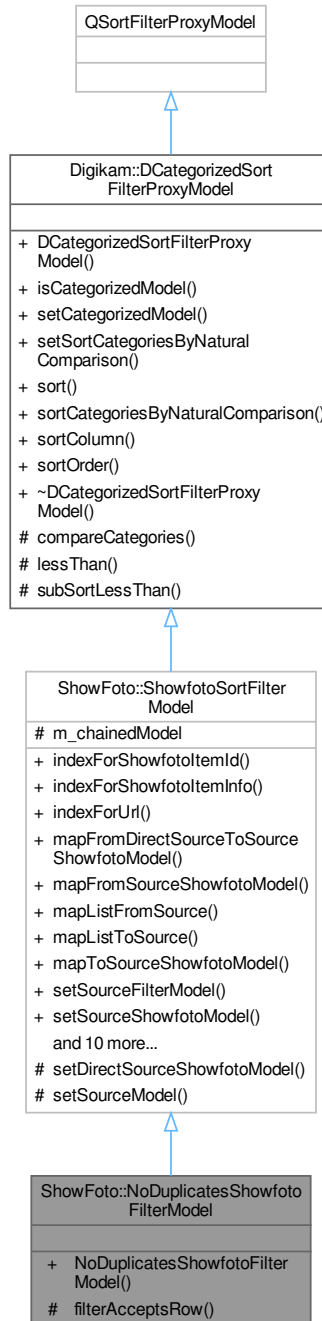
### 6.1572.1.4 loadFromURL()

```
bool Digikam::XmpWidget::loadFromURL (
    const QUrl & url ) [override], [virtual]
```

Implements [Digikam::MetadataWidget](#).

## 6.1573 ShowFoto::NoDuplicatesShowfotoFilterModel Class Reference

Inheritance diagram for ShowFoto::NoDuplicatesShowfotoFilterModel:



### Public Member Functions

- **NoDuplicatesShowfotoFilterModel** (QObject \*const parent=nullptr)



## Public Member Functions inherited from ShowFoto::ShowfotoSortFilterModel

- QModelIndex **indexForShowfotoItemId** (qulonglong id) const
- QModelIndex **indexForShowfotoItemInfo** (const ShowfotoItemInfo &info) const
- QModelIndex **indexForUrl** (const QUrl &fileUrl) const
- QModelIndex **mapFromDirectSourceToSourceShowfotoModel** (const QModelIndex &sourceModelIndex) const
- QModelIndex **mapFromSourceShowfotoModel** (const QModelIndex &showfotoModelIndex) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceShowfotoModel** (const QModelIndex &proxyIndex) const
- void **setSourceFilterModel** (ShowfotoSortFilterModel \*const sourceModel)
- void **setSourceShowfotoModel** (ShowfotoItemModel \*const sourceModel)
- virtual ShowfotoFilterModel \* **showfotoFilterModel** () const  
*Returns this, any chained ShowfotoFilterModel, or 0.*
- qulonglong **showfotoItemId** (const QModelIndex &index) const
- QList< qulonglong > **showfotoItemIds** (const QList< QModelIndex > &indexes) const
- ShowfotoItemInfo **showfotoItemInfo** (const QModelIndex &index) const
- QList< ShowfotoItemInfo > **showfotoItemInfos** (const QList< QModelIndex > &indexes) const
- QList< ShowfotoItemInfo > **showfotoItemInfosSorted** () const
- ShowfotoSortFilterModel (QObject \*const parent=nullptr)
- ShowfotoSortFilterModel \* **sourceFilterModel** () const
- ShowfotoItemModel \* **sourceShowfotoModel** () const

## Public Member Functions inherited from Digikam::DCategorizedSortFilterProxyModel

- DCategorizedSortFilterProxyModel (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void **setCategorizedModel** (bool categorizedModel)
- void **setSortCategoriesByNaturalComparison** (bool sortCategoriesByNaturalComparison)
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool **sortCategoriesByNaturalComparison** () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

## Protected Member Functions

- bool **filterAcceptsRow** (int source\_row, const QModelIndex &source\_parent) const override

## Protected Member Functions inherited from ShowFoto::ShowfotoSortFilterModel

- virtual void **setDirectSourceShowfotoModel** (ShowfotoItemModel \*const sourceModel)  
*Reimplement if needed. Called only when model shall be set as (direct) sourceModel.*
- void **setSourceModel** (QAbstractItemModel \*sourceModel) override

## Protected Member Functions inherited from Digikam::DCategorizedSortFilterProxyModel

- virtual int **compareCategories** (const QModelIndex &left, const QModelIndex &right) const
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool **subSortLessThan** (const QModelIndex &left, const QModelIndex &right) const

### Additional Inherited Members

### Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

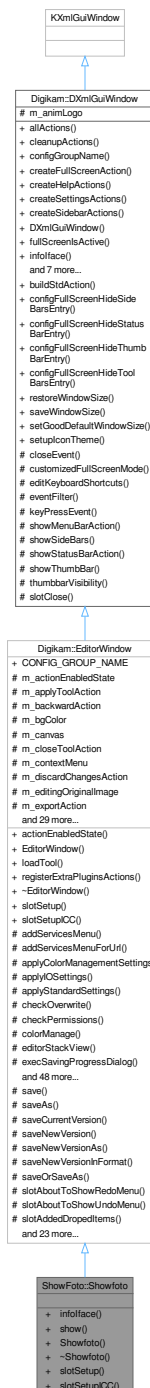
- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

### Protected Attributes inherited from [ShowFoto::ShowfotoSortFilterModel](#)

- [ShowfotoSortFilterModel](#) \* [m\\_chainedModel](#) = nullptr

## 6.1574 ShowFoto::Showfoto Class Reference

Inheritance diagram for ShowFoto::Showfoto:



### Classes

- class [Private](#)

**Public Slots**

- void **slotSetup** () override
- void **slotSetupICC** () override

**Public Slots inherited from [Digikam::EditorWindow](#)**

- void **slotSetup** () override=0
- virtual void **slotSetupICC** ()=0

**Signals**

- void **signalInfoList** (const ShowfotoItemInfoList &)
- void **signalLoadCurrentItem** (const QList< QUrl > &urlList)
- void **signalOpenFile** (const QList< QUrl > &urls)
- void **signalOpenFolder** (const QUrl &)

**Signals inherited from [Digikam::EditorWindow](#)**

- void **signalNoCurrentItem** ()
- void **signalPreviewModeChanged** (int)
- void **signalSelectionChanged** (const QRect &)
- void **signalToolApplied** ()

**Public Member Functions**

- [DInfoInterface](#) \* **infoface** ([DPluginAction](#) \*const ac) override
- virtual void **show** ()
- **Showfoto** (const QList< QUrl > &urlList, QWidget \*const parent=nullptr)

**Public Member Functions inherited from [Digikam::EditorWindow](#)**

- bool **actionEnabledState** () const
- **EditorWindow** (const QString &name, QWidget \*const parent=nullptr)
- void **loadTool** ([EditorTool](#) \*const tool)
- void **registerExtraPluginsActions** (QString &dom) override

**Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)**

- QList< QAction \* > **allActions** () const
- void **cleanupActions** ()
- QString **configGroupName** () const
- void **createFullscreenAction** (const QString &name)
- void **createHelpActions** (const QString &handbookSection, bool coreOptions=true)
- void **createSettingsActions** ()
- void **createSidebarActions** ()
- **DXmlGuiWindow** (QWidget \*const parent=nullptr, Qt::WindowFlags f=Qt::WindowFlags())
- bool **fullScreensActive** () const
- void **readFullscreenSettings** (const KConfigGroup &group)
- void **registerPluginsActions** ()
- void **setConfigGroupName** (const QString &name)
- void **setFullscreenOptions** (int options)
- void **unminimizeAndActivateWindow** ()

## Additional Inherited Members

### Public Types inherited from [Digikam::EditorWindow](#)

- enum **TransformType** { **RotateLeft** , **RotateRight** , **FlipHorizontal** , **FlipVertical** }

### Static Public Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- static QAction \* **buildStdAction** (StdActionType type, const QObject \*const recvr, const char \*const slot, QObject \*const parent)
- static QString **configFullScreenHideSideBarsEntry** ()
- static QString **configFullScreenHideStatusBarEntry** ()
- static QString **configFullScreenHideThumbBarEntry** ()
- static QString **configFullScreenHideToolBarsEntry** ()
- static void **restoreWindowSize** (QWindow \*const win, const KConfigGroup &group)
- static void **saveWindowSize** (QWindow \*const win, KConfigGroup &group)
- static void **setGoodDefaultWindowSize** (QWindow \*const win)
- static void **setupIconTheme** ()

### Static Public Attributes inherited from [Digikam::EditorWindow](#)

- static const QString **CONFIG\_GROUP\_NAME**

### Protected Types inherited from [Digikam::EditorWindow](#)

- enum **SaveAskMode** { **AskIfNeeded** , **OverwriteWithoutAsking** , **AlwaysSaveAs** , **SaveVersionWithoutAsking** = Overwrite↔ WithoutAsking , **AlwaysNewVersion** = AlwaysSaveAs }

### Protected Slots inherited from [Digikam::EditorWindow](#)

- virtual bool **saveOrSaveAs** ()
- void **slotAboutToShowRedoMenu** ()
- void **slotAboutToShowUndoMenu** ()
- virtual void **slotAddedDroppedItems** (QDropEvent \*e)=0
- virtual void **slotBackward** ()=0
- virtual void **slotChanged** ()=0
- void **slotComponentsInfo** () override
- virtual void **slotContextMenu** ()=0
- virtual void **slotDeleteCurrentItem** ()=0
- virtual void **slotDiscardChanges** ()
- virtual void **slotFileOriginChanged** (const QString &filePath)
- virtual void **slotFileWithDefaultApplication** ()=0
- virtual void **slotFirst** ()=0
- virtual void **slotForward** ()=0
- virtual void **slotLast** ()=0
- virtual void **slotLoadingFinished** (const QString &filename, bool success)
- void **slotLoadingProgress** (const QString &filePath, float progress)
- virtual void **slotLoadingStarted** (const QString &filename)

- void **slotNameLabelCancelButtonPressed** ()
- virtual void **slotOpenOriginal** ()
- virtual void **slotOpenWith** (QAction \*action=nullptr)=0
- virtual void **slotPrepareToLoad** ()
- virtual void **slotRevert** ()=0
- void **slotSavingProgress** (const QString &filePath, float progress)
- virtual void **slotSavingStarted** (const QString &filename)
- void **slotSelected** (bool)
- virtual void **slotUpdateItemInfo** ()=0

### Protected Slots inherited from [Digikam::DXmlGuiWindow](#)

- bool **slotClose** ()

### Protected Member Functions inherited from [Digikam::EditorWindow](#)

- void **addServicesMenuForUrl** (const QUrl &url)
- void **applyColorManagementSettings** ()
- void **applyIOSettings** ()
- void **applyStandardSettings** ()
- bool **checkOverwrite** (const QUrl &url)
- bool **checkPermissions** (const QUrl &url)
- void **colorManage** ()
- [EditorStackView](#) \* **editorStackView** () const
- void **execSavingProgressDialog** ()
- [ExposureSettingsContainer](#) \* **exposureSettings** () const
- virtual bool **hasOriginalToRestore** ()
- bool **moveLocalFile** (const QString &src, const QString &dest)
- void **movingSaveFileFinished** (bool successful)
- void **openWith** (const QUrl &url, QAction \*action)
- bool **promptForOverWrite** ()
- bool **promptUserDelete** (const QUrl &url)
- bool **promptUserSave** (const QUrl &url, SaveAskMode mode=AskIfNeeded, bool allowCancel=true)
- void **readStandardSettings** ()
- void **resetOrigin** ()
- void **resetOriginSwitchFile** ()
- virtual [DImageHistory](#) **resolvedImageHistory** (const [DImageHistory](#) &history)
- [VersionFileOperation](#) **saveAsVersionFileOperation** (const QUrl &url, const QUrl &saveLocation, const QString &format)
- [VersionFileOperation](#) **saveInFormatVersionFileOperation** (const QUrl &url, const QString &format)
- void **saveStandardSettings** ()
- [VersionFileOperation](#) **saveVersionFileOperation** (const QUrl &url, bool fork)
- void **setupContextMenu** ()
- void **setupSelectToolsAction** ()
- void **setupStandardActions** ()
- void **setupStandardConnections** ()
- void **setupStatusBar** ()
- [SidebarSplitter](#) \* **sidebarSplitter** () const
- void **startingSave** (const QUrl &url)
- bool **startingSaveAs** (const QUrl &url)
- bool **startingSaveCurrentVersion** (const QUrl &url)
- bool **startingSaveNewVersion** (const QUrl &url)
- bool **startingSaveNewVersionAs** (const QUrl &url)

- bool **startingSaveNewVersionInFormat** (const QUrl &url, const QString &format)
- void **toggleNonDestructiveActions** ()
- void **toggleStandardActions** (bool val)
- void **toggleToolActions** ([EditorTool](#) \*tool=nullptr)
- void **toggleZoomActions** (bool val)
- virtual [VersionManager](#) \* **versionManager** () const
- bool **waitForSavingToComplete** ()

### Protected Member Functions inherited from [Digikam::DXmlGuiWindow](#)

- void **closeEvent** (QCloseEvent \*e) override
- void **editKeyboardShortcuts** (KActionCollection \*const extraac=nullptr, const QString &actitle=QString())
- bool **eventFilter** (QObject \*obj, QEvent \*ev) override
- void **keyPressEvent** (QKeyEvent \*e) override
- QAction \* **showMenuBarAction** () const
- QAction \* **showStatusBarAction** () const

### Protected Attributes inherited from [Digikam::EditorWindow](#)

- bool **m\_actionEnabledState** = false
- QAction \* **m\_applyToolAction** = nullptr
- QAction \* **m\_backwardAction** = nullptr
- QColor **m\_bgColor**
- [Canvas](#) \* **m\_canvas** = nullptr
- QAction \* **m\_closeToolAction** = nullptr
- QMenu \* **m\_contextMenu** = nullptr
- QAction \* **m\_discardChangesAction** = nullptr
- bool **m\_editingOriginalImage** = true
- QAction \* **m\_exportAction** = nullptr
- QAction \* **m\_fileDeleteAction** = nullptr
- QAction \* **m\_firstAction** = nullptr
- QString **m\_formatForRAWVersioning**
- QString **m\_formatForSubversions**
- QAction \* **m\_forwardAction** = nullptr
- [IOFileSettings](#) \* **m\_IOFileSettings** = nullptr
- QAction \* **m\_lastAction** = nullptr
- [StatusProgressBar](#) \* **m\_nameLabel** = nullptr
- bool **m\_nonDestructive** = true
- QAction \* **m\_openVersionAction** = nullptr
- [KToolBarPopupAction](#) \* **m\_redoAction** = nullptr
- [DAdjustableLabel](#) \* **m\_resLabel** = nullptr
- QAction \* **m\_revertAction** = nullptr
- QAction \* **m\_saveAction** = nullptr
- QAction \* **m\_saveAsAction** = nullptr
- QAction \* **m\_saveCurrentVersionAction** = nullptr
- [KToolBarPopupAction](#) \* **m\_saveNewVersionAction** = nullptr
- QAction \* **m\_saveNewVersionAsAction** = nullptr
- QMenu \* **m\_saveNewVersionInFormatAction** = nullptr
- [SavingContext](#) **m\_savingContext**
- [QPointer](#)< [QProgressDialog](#) > **m\_savingProgressDialog** = nullptr
- QAction \* **m\_serviceAction** = nullptr
- QMenu \* **m\_servicesMenu** = nullptr
- bool **m\_setExifOrientationTag** = true

- QAction \* **m\_showBarAction** = nullptr
- SidebarSplitter \* **m\_splitter** = nullptr
- EditorStackView \* **m\_stackView** = nullptr
- QVector< TransformType > **m\_transformQue**  
*NOTE: using QVector to store transforms.*
- KToolBarPopupAction \* **m\_undoAction** = nullptr

## Protected Attributes inherited from [Digikam::DXmlGuiWindow](#)

- DLogoAction \* **m\_animLogo** = nullptr

## 6.1574.1 Member Function Documentation

### 6.1574.1.1 infolface()

```
DInfoInterface * ShowFoto::Showfoto::infoIface (
    DPluginAction *const ac ) [override], [virtual]
```

Return the interface instance to access to items information.

Implements [Digikam::DXmlGuiWindow](#).

## 6.1575 ShowFoto::Showfoto::Private Class Reference

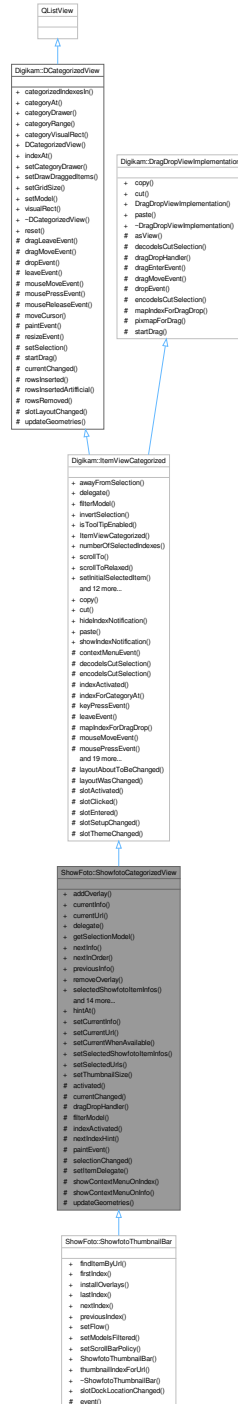
### Public Attributes

- QUrl **currentLoadedUrl**
- ShowfotoDragDropHandler \* **dDHandler** = nullptr
- QAction \* **fileOpenAction** = nullptr
- ShowfotoFilterModel \* **filterModel** = nullptr
- QAction \* **first** = nullptr
- ShowfotoFolderViewSideBar \* **folderView** = nullptr
- ShowfotoItemInfoList **infoList**
- int **itemsNb** = 0
- QUrl **lastOpenedDirectory**
- Digikam::Sidebar \* **leftSideBar** = nullptr
- QAction \* **mediaServerAction** = nullptr
- ShowfotoThumbnailModel \* **model** = nullptr
- ShowfotoNormalDelegate \* **normalDelegate** = nullptr
- QAction \* **openFilesInFolderAction** = nullptr
- Digikam::ItemPropertiesSideBar \* **rightSideBar** = nullptr
- ShowfotoSettings \* **settings** = nullptr
- Digikam::DSplashScreen \* **splash** = nullptr
- ShowfotoStackViewSideBar \* **stackView** = nullptr
- ShowfotoThumbnailBar \* **thumbBar** = nullptr
- Digikam::ThumbBarDock \* **thumbBarDock** = nullptr
- Digikam::ThumbnailLoadThread \* **thumbLoadThread** = nullptr
- bool **validIccPath** = true
- QSplitter \* **vSplitter** = nullptr



## 6.1576 ShowFoto::ShowfotoCategorizedView Class Reference

Inheritance diagram for ShowFoto::ShowfotoCategorizedView:



### Public Slots

- void `hintAt` (const `ShowfotoItemInfo` &info)
- void `setCurrentInfo` (const `ShowfotoItemInfo` &info)

- void [setCurrentUrl](#) (const QUrl &url)
- void [setCurrentWhenAvailable](#) (qulonglong ShowfotoItemId)
- void [setSelectedShowfotoItemInfos](#) (const QList< [ShowfotoItemInfo](#) > &infos)
- void [setSelectedUrls](#) (const QList< QUrl > &urlList)
- void [setThumbnailSize](#) (int size)

### Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void [copy](#) () override
- void [cut](#) () override
- void [hideIndexNotification](#) ()
- void [paste](#) () override
- void [showIndexNotification](#) (const QModelIndex &index, const QString &message)

### Public Slots inherited from [Digikam::DCategorizedView](#)

- void [reset](#) () override

### Signals

- void [currentChanged](#) (const [ShowfotoItemInfo](#) &info)
- void [deselected](#) (const QList< [ShowfotoItemInfo](#) > &nowDeselectedInfos)
- void [modelChanged](#) ()
- void [selected](#) (const QList< [ShowfotoItemInfo](#) > &newSelectedInfos)
- void [showfotoItemInfoActivated](#) (const [ShowfotoItemInfo](#) &info)

### Signals inherited from [Digikam::ItemViewCategorized](#)

- void [clicked](#) (const QMouseEvent \*e, const QModelIndex &index)
- void [entered](#) (const QMouseEvent \*e, const QModelIndex &index)
- void [keyPressed](#) (QKeyEvent \*e)
- void [selectionChanged](#) ()
- void [selectionCleared](#) ()
- void [viewportClicked](#) (const QMouseEvent \*e)
- void [zoomInStep](#) ()
- void [zoomOutStep](#) ()

### Public Member Functions

- void [addOverlay](#) ([ItemDelegateOverlay](#) \*overlay, [ShowfotoDelegate](#) \*delegate=nullptr)
- [ShowfotoItemInfo](#) [currentInfo](#) () const
- QUrl [currentUrl](#) () const
- [ShowfotoDelegate](#) \* [delegate](#) () const
- QListSelectionModel \* [getSelectionModel](#) () const
- [ShowfotoItemInfo](#) [nextInfo](#) (const [ShowfotoItemInfo](#) &info)
- [ShowfotoItemInfo](#) [nextInOrder](#) (const [ShowfotoItemInfo](#) &startingPoint, int nth)
- [ShowfotoItemInfo](#) [previousInfo](#) (const [ShowfotoItemInfo](#) &info)
- void [removeOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- QList< [ShowfotoItemInfo](#) > [selectedShowfotoItemInfos](#) () const
- QList< [ShowfotoItemInfo](#) > [selectedShowfotoItemInfosCurrentFirst](#) () const

- `QList< QUrl > selectedUrls ()` const
- void `setModels (ShowfotoItemModel *model, ShowfotoSortFilterModel *filterModel)`
- virtual void `setThumbnailSize (const ThumbnailSize &size)`
- `ShowfotoCategorizedView (QWidget *const parent=nullptr)`
- `ShowfotoFilterModel * showfotoFilterModel ()` const
- `QList< ShowfotoItemInfo > showfotoItemInfos ()` const
- `ShowfotoItemModel * showfotoItemModel ()` const
- `ShowfotoSortFilterModel * showfotoSortFilterModel ()` const
- `ShowfotoThumbnailModel * showfotoThumbnailModel ()` const
- `ThumbnailSize thumbnailSize ()` const
- void `toIndex (const QUrl &url)`
- `QList< QUrl > urls ()` const

### Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void `awayFromSelection ()`
- `DItemDelegate * delegate ()` const
- void `invertSelection ()`
- bool `isToolTipEnabled ()` const
- `ItemViewCategorized (QWidget *const parent=nullptr)`
- int `numberOfSelectedIndexes ()` const
- void `scrollTo (const QModelIndex &index, ScrollHint hint=EnsureVisible)` override
- void `scrollToRelaxed (const QModelIndex &index, ScrollHint hint=EnsureVisible)`
- void `setInitialSelectedItem (bool enabled)`
- void `setScrollCurrentToCenter (bool enabled)`
- void `setScrollStepGranularity (int factor)`
- void `setSelectedIndexes (const QList< QModelIndex > &indexes)`
- void `setSpacing (int spacing)`
- void `setToolTipEnabled (bool enabled)`
- void `setUsePointingHandCursor (bool useCursor)`
- void `toFirstIndex ()`
- void `toIndex (const QModelIndex &index)`
- void `toLastIndex ()`
- void `toNextIndex ()`
- void `toPreviousIndex ()`

### Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual `QModelIndexList categorizedIndexesIn (const QRect &rect)` const
- virtual `QModelIndex categoryAt (const QPoint &point)` const
- `DCategoryDrawer * categoryDrawer ()` const
- virtual `QItemSelectionRange categoryRange (const QModelIndex &index)` const
- virtual `QRect categoryVisualRect (const QModelIndex &index)` const
- `DCategorizedView (QWidget *const parent=nullptr)`
- `QModelIndex indexAt (const QPoint &point)` const override
- void `setCategoryDrawer (DCategoryDrawer *categoryDrawer)`
- void `setDrawDraggedItems (bool drawDraggedItems)`
- void `setGridSize (const QSize &size)`
- void `setModel (QAbstractItemModel *model)` override
- `QRect visualRect (const QModelIndex &index)` const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

## Protected Member Functions

- virtual void **activated** (const [ShowfotoItemInfo](#) &info, Qt::KeyboardModifiers modifiers)  
*Reimplement these in a subclass.*
- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override  
*reimplemented from parent class*
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **paintEvent** (QPaintEvent \*e) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([ShowfotoDelegate](#) \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override  
*Reimplement these in a subclass.*
- virtual void **showContextMenuOnInfo** (QContextMenuEvent \*event, const [ShowfotoItemInfo](#) &info)
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([DItemDelegate](#) \*delegate)
- void **setToolTip** ([ItemViewToolTip](#) \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

## Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

## Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

## Additional Inherited Members

## Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void **layoutAboutToBeChanged** ()
- void **layoutWasChanged** ()
- void **slotActivated** (const QModelIndex &index)
- void **slotClicked** (const QModelIndex &index)
- void **slotEntered** (const QModelIndex &index)
- virtual void **slotSetupChanged** ()
- virtual void **slotThemeChanged** ()

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## 6.1576.1 Member Function Documentation

### 6.1576.1.1 addOverlay()

```
void ShowFoto::ShowfotoCategorizedView::addOverlay (
    ItemDelegateOverlay * overlay,
    ShowfotoDelegate * delegate = nullptr )
```

Add and remove an overlay. It will as well be removed automatically when destroyed. Unless you pass a different delegate, the current delegate will be used.

### 6.1576.1.2 deselected

```
void ShowFoto::ShowfotoCategorizedView::deselected (
    const QList< ShowfotoItemInfo > & nowDeselectedInfos ) [signal]
```

Emitted when items are deselected. There may be other selected infos left. This signal is not emitted when the model is reset; then only selectionCleared is emitted.

### 6.1576.1.3 dragDropHandler()

```
AbstractItemDragDropHandler * ShowFoto::ShowfotoCategorizedView::dragDropHandler ( ) const
[override], [protected], [virtual]
```

You need to implement these three methods Returns the drag drop handler.

Implements [Digikam::DragDropViewImplementation](#).

### 6.1576.1.4 filterModel()

```
QSortFilterProxyModel * ShowFoto::ShowfotoCategorizedView::filterModel ( ) const [override],
[protected], [virtual]
```

Implements [Digikam::ItemViewCategorized](#).

### 6.1576.1.5 hintAt

```
void ShowFoto::ShowfotoCategorizedView::hintAt (
    const ShowfotoItemInfo & info ) [slot]
```

Does something to gain attention for info, but not changing current selection

### 6.1576.1.6 indexActivated()

```
void ShowFoto::ShowfotoCategorizedView::indexActivated (
    const QModelIndex & index,
    Qt::KeyboardModifiers modifiers ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

### 6.1576.1.7 modelChanged

```
void ShowFoto::ShowfotoCategorizedView::modelChanged ( ) [signal]
```

Emitted when a new model is set.

### 6.1576.1.8 nextIndexHint()

```
QModelIndex ShowFoto::ShowfotoCategorizedView::nextIndexHint (
    const QModelIndex & indexToAnchor,
    const QItemSelectionRange & removed ) const [override], [protected], [virtual]
```

Assuming the given indexes would be removed (hypothetically!), return the index to be selected instead, starting from anchor. The default implementation returns the next remaining sibling.

Reimplemented from [Digikam::ItemViewCategorized](#).

### 6.1576.1.9 nextInOrder()

```
ShowfotoItemInfo ShowFoto::ShowfotoCategorizedView::nextInOrder (
    const ShowfotoItemInfo & startingPoint,
    int nth )
```

Returns the n-th info after the given one. Specifically, return the previous info for  $n = -1$  and the next info for  $n = 1$ . Returns a null info if either *startingPoint* or the *nth* info are not contained in the model

### 6.1576.1.10 selected

```
void ShowFoto::ShowfotoCategorizedView::selected (
    const QList< ShowfotoItemInfo > & newSelectedInfos ) [signal]
```

Emitted when new items are selected. The parameter includes only the newly selected infos, there may be other already selected infos.

### 6.1576.1.11 setCurrentInfo

```
void ShowFoto::ShowfotoCategorizedView::setCurrentInfo (
    const ShowfotoItemInfo & info ) [slot]
```

Set as current item the item identified by the [ShowfotoItemInfo](#)

### 6.1576.1.12 setCurrentUrl

```
void ShowFoto::ShowfotoCategorizedView::setCurrentUrl (
    const QUrl & url ) [slot]
```

Set as current item the item identified by its file url

### 6.1576.1.13 setCurrentWhenAvailable

```
void ShowFoto::ShowfotoCategorizedView::setCurrentWhenAvailable (
    qlonglong ShowfotoItemId ) [slot]
```

Scroll the view to the given item when it becomes available

### 6.1576.1.14 setSelectedShowfotoItemInfos

```
void ShowFoto::ShowfotoCategorizedView::setSelectedShowfotoItemInfos (
    const QList< ShowfotoItemInfo > & infos ) [slot]
```

Set selected items

### 6.1576.1.15 setSelectedUrls

```
void ShowFoto::ShowfotoCategorizedView::setSelectedUrls (
    const QList< QUrl > & urlList ) [slot]
```

Set selected items identified by their file urls

### 6.1576.1.16 showContextMenuOnIndex()

```
void ShowFoto::ShowfotoCategorizedView::showContextMenuOnIndex (
    QContextMenuEvent * event,
    const QModelIndex & index ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::ItemViewCategorized](#).

### 6.1576.1.17 showfotoFilterModel()

```
ShowfotoFilterModel * ShowFoto::ShowfotoCategorizedView::showfotoFilterModel ( ) const
```

Returns any [ShowfotoFilterModel](#) in chain. May not be sourceModel()

### 6.1576.1.18 showfotoItemInfoActivated

```
void ShowFoto::ShowfotoCategorizedView::showfotoItemInfoActivated (
    const ShowfotoItemInfo & info ) [signal]
```

Emitted when the given [ShowfotoItemInfo](#) is activated. Info is never null.

### 6.1576.1.19 showfotoThumbnailModel()

```
ShowfotoThumbnailModel * ShowFoto::ShowfotoCategorizedView::showfotoThumbnailModel ( ) const
```

Returns 0 if the [ShowfotoItemModel](#) is not an [ShowfotoThumbnailModel](#)



## 6.1576.1.20 toIndex()

```
void ShowFoto::ShowfotoCategorizedView::toIndex (
    const QUrl & url )
```

Selects the index as current and scrolls to it

## 6.1577 ShowFoto::ShowfotoCoordinatesOverlay Class Reference

Inheritance diagram for ShowFoto::ShowfotoCoordinatesOverlay:



### Public Member Functions

- [ShowfotoCoordinatesOverlayWidget](#) \* **buttonWidget** () const
- **ShowfotoCoordinatesOverlay** (QObject \*const parent)

### Public Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- [AbstractWidgetDelegateOverlay](#) (QObject \*const parent)

### Public Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- virtual bool **acceptsDelegate** (QAbstractItemDelegate \*) const
- QAbstractItemDelegate \* **delegate** () const
- **ItemDelegateOverlay** (QObject \*const parent=nullptr)
- virtual void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- virtual void **paint** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index)
- void **setDelegate** (QAbstractItemDelegate \*delegate)
- void **setView** (QAbstractItemView \*view)
- QAbstractItemView \* **view** () const

### Protected Member Functions

- bool **checkIndex** (const QModelIndex &index) const override
- QWidget \* **createWidget** () override
- void **setActive** (bool active) override
- void **slotEntered** (const QModelIndex &index) override
- void **updatePosition** ()
- void **visualChange** () override

### Protected Member Functions inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- bool **checkIndexOnEnter** (const QModelIndex &index) const
- bool **eventFilter** (QObject \*obj, QEvent \*event) override
- virtual void **hide** ()
- virtual QString **notifyMultipleMessage** (const QModelIndex &, int number)
- QWidget \* **parentWidget** () const
- virtual void **viewportLeaveEvent** (QObject \*obj, QEvent \*event)
- virtual void **widgetEnterEvent** ()
- void **widgetEnterNotifyMultiple** (const QModelIndex &index)
- virtual void **widgetLeaveEvent** ()
- void **widgetLeaveNotifyMultiple** ()

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlay](#)

- QList< QModelIndex > **affectedIndexes** (const QModelIndex &index) const
- bool **affectsMultiple** (const QModelIndex &index) const
- int **numberOfAffectedIndexes** (const QModelIndex &index) const
- bool **viewHasMultiSelection** () const

### Protected Attributes

- `QPersistentModelIndex m_index`

### Protected Attributes inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- `bool m_mouseButtonPressedOnWidget = false`
- `QWidget * m_widget = nullptr`

### Protected Attributes inherited from [Digikam::ItemDelegateOverlay](#)

- `QAbstractItemDelegate * m_delegate = nullptr`
- `QAbstractItemView * m_view = nullptr`

### Additional Inherited Members

### Signals inherited from [Digikam::ItemDelegateOverlay](#)

- `void hideNotification ()`
- `void requestNotification (const QModelIndex &index, const QString &message)`
- `void update (const QModelIndex &index)`

### Protected Slots inherited from [Digikam::AbstractWidgetDelegateOverlay](#)

- `virtual void slotLayoutChanged ()`
- `virtual void slotReset ()`
- `virtual void slotRowsRemoved (const QModelIndex &parent, int start, int end)`
- `virtual void slotViewportEntered ()`

### Protected Slots inherited from [Digikam::ItemDelegateOverlay](#)

## 6.1577.1 Member Function Documentation

### 6.1577.1.1 `checkIndex()`

```
bool ShowFoto::ShowfotoCoordinatesOverlay::checkIndex (
    const QModelIndex & index ) const [override], [protected], [virtual]
```

Return true here if you want to show the overlay for the given index. The default implementation returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

### 6.1577.1.2 `createWidget()`

```
QWidget * ShowFoto::ShowfotoCoordinatesOverlay::createWidget ( ) [override], [protected],
[virtual]
```

Create your widget here. When creating the object, pass `parentWidget()` as parent widget. Ownership of the object is passed. It will be deleted in `setActive(false)`.

Implements [Digikam::AbstractWidgetDelegateOverlay](#).

**6.1577.1.3 setActive()**

```
void ShowFoto::ShowfotoCoordinatesOverlay::setActive (
    bool active ) [override], [protected], [virtual]
```

If active is true, this will call [createWidget\(\)](#), initialize the widget for use, and setup connections for the virtual slots. If active is false, this will delete the widget and disconnect all signal from model and view to this object (!)

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

**6.1577.1.4 slotEntered()**

```
void ShowFoto::ShowfotoCoordinatesOverlay::slotEntered (
    const QModelIndex & index ) [override], [protected], [virtual]
```

Default implementation shows the widget iff the index is valid and checkIndex returns true.

Reimplemented from [Digikam::AbstractWidgetDelegateOverlay](#).

**6.1577.1.5 visualChange()**

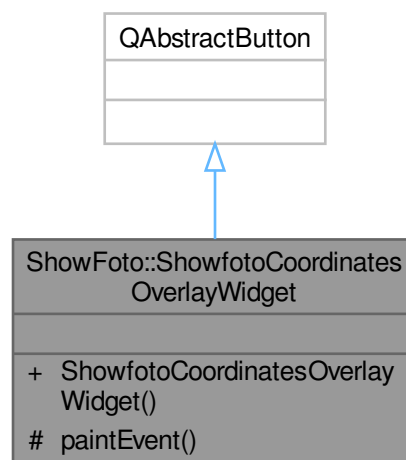
```
void ShowFoto::ShowfotoCoordinatesOverlay::visualChange ( ) [override], [protected], [virtual]
```

Called when any change from the delegate occurs - when the overlay is installed, when size hints, styles or fonts change

Reimplemented from [Digikam::ItemDelegateOverlay](#).

**6.1578 ShowFoto::ShowfotoCoordinatesOverlayWidget Class Reference**

Inheritance diagram for ShowFoto::ShowfotoCoordinatesOverlayWidget:



**Public Member Functions**

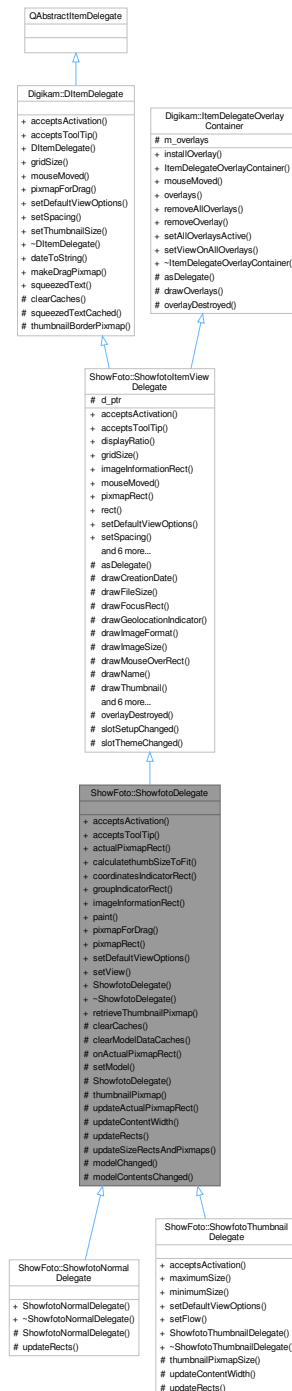
- **ShowfotoCoordinatesOverlayWidget** (QWidget \*const parent=nullptr)

**Protected Member Functions**

- void **paintEvent** (QPaintEvent \*) override

## 6.1579 ShowFoto::ShowfotoDelegate Class Reference

Inheritance diagram for ShowFoto::ShowfotoDelegate:



### Classes

- class [ShowfotoDelegatePrivate](#)

## Public Member Functions

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect [actualPixmapRect](#) (const QModelIndex &index) const
- int [calculatethumbSizeToFit](#) (int ws)
- QRect [coordinatesIndicatorRect](#) () const
- QRect [groupIndicatorRect](#) () const
- QRect [imageInformationRect](#) () const override
- void [paint](#) (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setView](#) ([ShowfotoThumbnailBar](#) \*view)
- **ShowfotoDelegate** (QWidget \*const parent)

## Public Member Functions inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double [displayRatio](#) () const
- QSize [gridSize](#) () const override
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- QRect [rect](#) () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from [DItemDelegate](#)*
- **ShowfotoItemViewDelegate** (QWidget \*const parent)
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int [spacing](#) () const
- [ThumbnailSize](#) [thumbnailSize](#) () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- **DItemDelegate** (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void [installOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > [overlays](#) () const
- void [removeAllOverlays](#) ()
- void [removeOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- void [setAllOverlaysActive](#) (bool active)
- void [setViewOnAllOverlays](#) (QAbstractItemView \*view)

### Static Public Member Functions

- static QPixmap [retrieveThumbnailPixmap](#) (const QModelIndex &index, int thumbnailSize)

### Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString [dateToString](#) (const QDateTime &datetime)
- static QPixmap [makeDragPixmap](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString [squeezedText](#) (const QFontMetrics &fm, int width, const QString &text)

### Protected Slots

- void [modelChanged](#) ()
- void [modelContentsChanged](#) ()

### Protected Slots inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- void [overlayDestroyed](#) (QObject \*o) override
- void [slotSetupChanged](#) ()
- void [slotThemeChanged](#) ()

### Protected Member Functions

- void [clearCaches](#) () override
- virtual void [clearModelDataCaches](#) ()
- bool [onActualPixmapRect](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void [setModel](#) (QAbstractItemModel \*model)
- [ShowfotoDelegate](#) ([ShowfotoDelegate::ShowfotoDelegatePrivate](#) &dd, QWidget \*const parent)
- virtual QPixmap [thumbnailPixmap](#) (const QModelIndex &index) const
- void [updateActualPixmapRect](#) (const QModelIndex &index, const QRect &rect)
- virtual void [updateContentWidth](#) ()
- virtual void [updateRects](#) ()=0
- void [updateSizeRectsAndPixmaps](#) () override

### Protected Member Functions inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- QAbstractItemDelegate \* [asDelegate](#) () override  
*Returns the delegate, typically, the derived class.*
- void [drawCreationDate](#) (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void [drawFileSize](#) (QPainter \*p, const QRect &r, qlonglong bytes) const
- void [drawFocusRect](#) (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void [drawGeolocationIndicator](#) (QPainter \*p, const QRect &r) const
- void [drawImageFormat](#) (QPainter \*p, const QRect &dimsRect, const QString &mime) const
- void [drawImageSize](#) (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void [drawMouseOverRect](#) (QPainter \*p, const QStyleOptionViewItem &option) const
- void [drawName](#) (QPainter \*p, const QRect &nameRect, const QString &name) const
- QRect [drawThumbnail](#) (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail) const  
*Use the tool methods for painting in subclasses.*
- virtual void [invalidatePaintingCache](#) ()  
*reimplement these in subclasses*
- void [prepareBackground](#) ()
- void [prepareFonts](#) ()
- void [prepareMetrics](#) (int maxWidth)
- [ShowfotoItemViewDelegate](#) ([ShowfotoItemViewDelegatePrivate](#) &dd, QWidget \*const parent)



## Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)

*Declare as slot in the derived class calling this method.*

## Additional Inherited Members

## Signals inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

## Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Protected Attributes inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- [ShowfotoItemViewDelegatePrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > **m\_overlays**

## 6.1579.1 Member Function Documentation

### 6.1579.1.1 acceptsActivation()

```
bool ShowFoto::ShowfotoDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect = nullptr ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

**6.1579.1.2 acceptsToolTip()**

```
bool ShowFoto::ShowfotoDelegate::acceptsToolTip (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * tooltipRect = nullptr ) const [override], [virtual]
```

These methods take four parameters: The position on viewport, the rect on viewport, the index, and optionally a parameter into which, if the return value is true, a rectangle can be written for which the return value will be true as well.

Implements [Digikam::DItemDelegate](#).

**6.1579.1.3 clearCaches()**

```
void ShowFoto::ShowfotoDelegate::clearCaches ( ) [override], [protected], [virtual]
```

Reimplemented from [Digikam::DItemDelegate](#).

**6.1579.1.4 clearModelDataCaches()**

```
void ShowFoto::ShowfotoDelegate::clearModelDataCaches ( ) [protected], [virtual]
```

Reimplement to clear caches based on model indexes (hash on row number etc.) Change signals are listened to this is called whenever such properties become invalid.

**6.1579.1.5 imageInformationRect()**

```
QRect ShowFoto::ShowfotoDelegate::imageInformationRect ( ) const [override], [virtual]
```

Returns the area where the image information is drawn, or null if empty / not supported. The image information is textual or graphical information, but not the pixmap. The ratingRect() will e.g. typically be contained in this area.

Reimplemented from [ShowFoto::ShowfotoItemViewDelegate](#).

**6.1579.1.6 pixmapForDrag()**

```
QPixmap ShowFoto::ShowfotoDelegate::pixmapForDrag (
    const QStyleOptionViewItem & option,
    const QList< QModelIndex > & indexes ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

**6.1579.1.7 pixmapRect()**

```
QRect ShowFoto::ShowfotoDelegate::pixmapRect ( ) const [override], [virtual]
```

Returns the area where the pixmap is drawn, or null if not supported

Reimplemented from [ShowFoto::ShowfotoItemViewDelegate](#).

### 6.1579.1.8 retrieveThumbnailPixmap()

```
QPixmap ShowFoto::ShowfotoDelegate::retrieveThumbnailPixmap (
    const QModelIndex & index,
    int thumbnailSize ) [static]
```

Retrieve the thumbnail pixmap in given size for the [ShowfotoItemModel::ThumbnailRole](#) for the given index from the given index, which must adhere to [ShowfotoThumbnailModel](#) semantics.

### 6.1579.1.9 setDefaultViewOptions()

```
void ShowFoto::ShowfotoDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. option.rect shall be the viewport rectangle. Call on resize, font change.

Implements [Digikam::DItemDelegate](#).

Reimplemented in [ShowFoto::ShowfotoThumbnailDelegate](#).

### 6.1579.1.10 updateContentWidth()

```
void ShowFoto::ShowfotoDelegate::updateContentWidth ( ) [protected], [virtual]
```

Reimplement this to set contentWidth. This is the maximum width of all content rectangles, typically excluding margins on both sides.

Reimplemented in [ShowFoto::ShowfotoThumbnailDelegate](#).

### 6.1579.1.11 updateRects()

```
virtual void ShowFoto::ShowfotoDelegate::updateRects ( ) [protected], [pure virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The paint() method operates depending on these rects.

Implemented in [ShowFoto::ShowfotoThumbnailDelegate](#), and [ShowFoto::ShowfotoNormalDelegate](#).

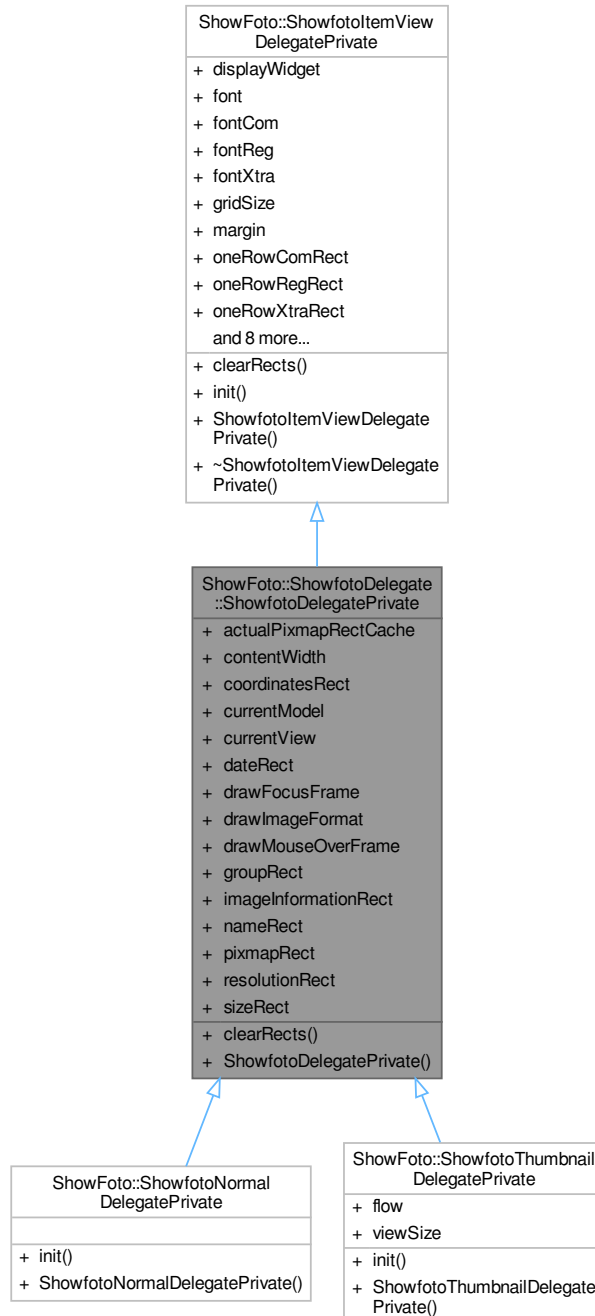
### 6.1579.1.12 updateSizeRectsAndPixmaps()

```
void ShowFoto::ShowfotoDelegate::updateSizeRectsAndPixmaps ( ) [override], [protected], [virtual]
```

Implements [ShowFoto::ShowfotoItemViewDelegate](#).

## 6.1580 ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate Class Reference

Inheritance diagram for ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate:



### Public Member Functions

- void `clearRects` () override  
*Resets cached rects. Remember to reimplement in subclass for added rects.*

## Public Member Functions inherited from ShowFoto::ShowfotoItemViewDelegatePrivate

- void **init** ([ShowfotoItemViewDelegate](#) \*const \_q, QWidget \*const \_widget)

### Public Attributes

- QCache< int, QRect > **actualPixmapRectCache**
- int **contentWidth** = 0
- QRect **coordinatesRect**
- QAbstractItemModel \* **currentModel** = nullptr
- [ShowfotoThumbnailBar](#) \* **currentView** = nullptr
- QRect **dateRect**
- bool **drawFocusFrame** = true
- bool **drawImageFormat** = true
- bool **drawMouseOverFrame** = true
- QRect **groupRect**
- QRect **imageInformationRect**
- QRect **nameRect**
- QRect **pixmapRect**
- QRect **resolutionRect**
- QRect **sizeRect**

## Public Attributes inherited from ShowFoto::ShowfotoItemViewDelegatePrivate

- QWidget \* **displayWidget** = nullptr
- QFont **font**
- QFont **fontCom**
- QFont **fontReg**
- QFont **fontXtra**
- QSize **gridSize**
- int **margin** = 5
- QRect **oneRowComRect**
- QRect **oneRowRegRect**
- QRect **oneRowXtraRect**
- [ShowfotoItemViewDelegate](#) \* **q** = nullptr
- int **radius** = 3
- *constant values for drawing*
- QVector< QPixmap > **ratingPixmaps**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

## 6.1580.1 Member Function Documentation

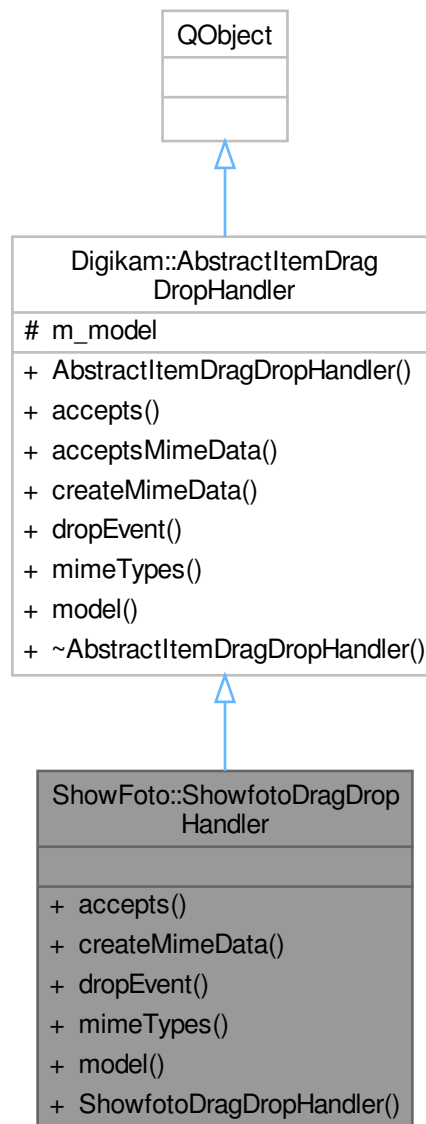
### 6.1580.1.1 clearRects()

```
void ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate::clearRects ( ) [override], [virtual]
```

Reimplemented from [ShowFoto::ShowfotoItemViewDelegatePrivate](#).

## 6.1581 ShowFoto::ShowfotoDragDropHandler Class Reference

Inheritance diagram for ShowFoto::ShowfotoDragDropHandler:



### Signals

- void **signalDroppedUrls** (const QList< QUrl > &droppedUrls, bool dropped, const QUrl &current)

### Public Member Functions

- Qt::DropAction **accepts** (const QDropEvent \*e, const QModelIndex &dropIndex) override

- `QMimeData * createMimeData` (const QList< QModelIndex > &) override
- bool `dropEvent` (QAbstractItemView \*view, const QDropEvent \*e, const QModelIndex &droppedOn) override
- QStringList `mimeTypes` () const override
- `ShowfotoItemModel * model` () const
- `ShowfotoDragDropHandler` (`ShowfotoItemModel *const model`)

## Public Member Functions inherited from [Digikam::AbstractItemDragDropHandler](#)

- `AbstractItemDragDropHandler` (QAbstractItemModel \*const model)
- virtual bool `acceptsMimeData` (const QMimeData \*data)
- QAbstractItemModel \* `model` () const

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::AbstractItemDragDropHandler](#)

- QAbstractItemModel \* `m_model` = nullptr

## 6.1581.1 Member Function Documentation

### 6.1581.1.1 `accepts()`

```
Qt::DropAction ShowFoto::ShowfotoDragDropHandler::accepts (
    const QDropEvent * e,
    const QModelIndex & dropIndex ) [override], [virtual]
```

Returns if the given mime data is accepted for drop on dropIndex. Returns the proposed action, or Qt::IgnoreAction if not accepted.

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.1581.1.2 `createMimeData()`

```
QMimeData * ShowFoto::ShowfotoDragDropHandler::createMimeData (
    const QList< QModelIndex > & ) [override], [virtual]
```

Create a mime data object for starting a drag from the given Albums

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

### 6.1581.1.3 `dropEvent()`

```
bool ShowFoto::ShowfotoDragDropHandler::dropEvent (
    QAbstractItemView * view,
    const QDropEvent * e,
    const QModelIndex & droppedOn ) [override], [virtual]
```

Gives the view and the occurring drop event. The index is the index where the drop was dropped on. It may be invalid (dropped on decoration, viewport) Returns true if the event is to be accepted.

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).

#### 6.1581.1.4 mimeTypees()

```
QStringList ShowFoto::ShowfotoDragDropHandler::mimeTypees ( ) const [override], [virtual]
```

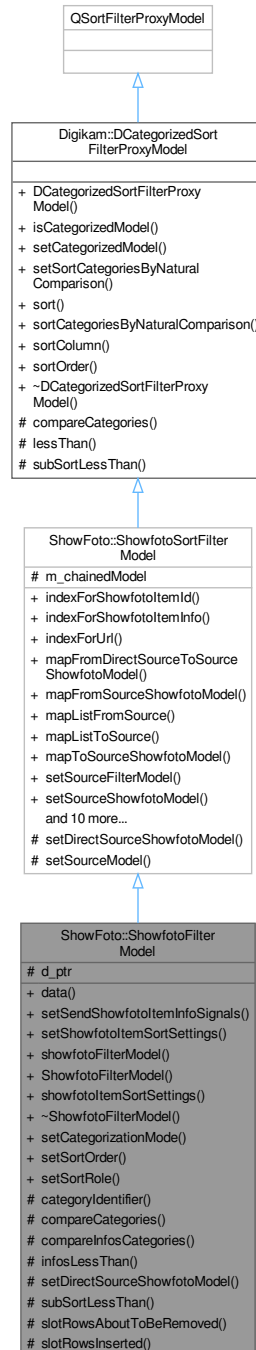
Returns the supported mime types. Called by the default implementation of model's [mimeTypees\(\)](#).

Reimplemented from [Digikam::AbstractItemDragDropHandler](#).



## 6.1582 ShowFoto::ShowfotoFilterModel Class Reference

Inheritance diagram for ShowFoto::ShowfotoFilterModel:



### Public Types

- enum `ShowfotoFilterModelRoles` { `CategorizationModeRole` = `ShowfotoItemModel::FilterModelRoles` + 1 , `SortOrderRole` = `ShowfotoItemModel::FilterModelRoles` + 2 , `CategoryFormatRole` = `ShowfotoItemModel::FilterModelRoles` + 3 , `ShowfotoFilterModelPointerRole` = `ShowfotoItemModel::FilterModelRoles` + 50 }

## Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## Public Slots

- void **setCategorizationMode** ([ShowfotoItemSortSettings::CategorizationMode](#) mode)
- void **setSortOrder** ([ShowfotoItemSortSettings::SortOrder](#) order)
- void **setSortRole** ([ShowfotoItemSortSettings::SortRole](#) role)

## Signals

- void **showfotoItemInfosAboutToBeRemoved** (const QList< [ShowfotoItemInfo](#) > &infos)
- void [showfotoItemInfosAdded](#) (const QList< [ShowfotoItemInfo](#) > &infos)

## Public Member Functions

- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- void **setSendShowfotoItemInfoSignals** (bool sendSignals)
  - *Enables sending ShowfotoItemInfosAdded and ShowfotoItemInfosAboutToBeRemoved.*
- void **setShowfotoItemSortSettings** (const [ShowfotoItemSortSettings](#) &sorter)
- [ShowfotoFilterModel](#) \* **showfotoFilterModel** () const override
  - *Returns this, any chained ShowfotoFilterModel, or 0.*
- [ShowfotoFilterModel](#) ([QObject](#) \*const parent=nullptr)
- [ShowfotoItemSortSettings](#) **showfotoItemSortSettings** () const

## Public Member Functions inherited from [ShowFoto::ShowfotoSortFilterModel](#)

- QModelIndex **indexForShowfotoItemId** (qulonglong id) const
- QModelIndex **indexForShowfotoItemInfo** (const [ShowfotoItemInfo](#) &info) const
- QModelIndex **indexForUrl** (const [QUrl](#) &fileUrl) const
- QModelIndex **mapFromDirectSourceToSourceShowfotoModel** (const QModelIndex &sourceModelIndex) const
- QModelIndex **mapFromSourceShowfotoModel** (const QModelIndex &showfotoModelIndex) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceShowfotoModel** (const QModelIndex &proxyIndex) const
- void **setSourceFilterModel** ([ShowfotoSortFilterModel](#) \*const sourceModel)
- void **setSourceShowfotoModel** ([ShowfotoItemModel](#) \*const sourceModel)
- qulonglong **showfotoItemId** (const QModelIndex &index) const
- QList< qulonglong > **showfotoItemIds** (const QList< QModelIndex > &indexes) const
- [ShowfotoItemInfo](#) **showfotoItemInfo** (const QModelIndex &index) const
- QList< [ShowfotoItemInfo](#) > **showfotoItemInfos** (const QList< QModelIndex > &indexes) const
- QList< [ShowfotoItemInfo](#) > **showfotoItemInfosSorted** () const
- [ShowfotoSortFilterModel](#) ([QObject](#) \*const parent=nullptr)
- [ShowfotoSortFilterModel](#) \* **sourceFilterModel** () const
- [ShowfotoItemModel](#) \* **sourceShowfotoModel** () const

## Public Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- **DCategorizedSortFilterProxyModel** (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void **setCategorizedModel** (bool categorizedModel)
- void **setSortCategoriesByNaturalComparison** (bool [sortCategoriesByNaturalComparison](#))
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool **sortCategoriesByNaturalComparison** () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

## Protected Slots

- void **slotRowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end)
- void **slotRowsInserted** (const QModelIndex &parent, int start, int end)

## Protected Member Functions

- virtual QString **categoryIdentifier** (const [ShowfotoItemInfo](#) &info) const
- int **compareCategories** (const QModelIndex &left, const QModelIndex &right) const override
- virtual int **compareInfosCategories** (const [ShowfotoItemInfo](#) &left, const [ShowfotoItemInfo](#) &right) const
- virtual bool **infosLessThan** (const [ShowfotoItemInfo](#) &left, const [ShowfotoItemInfo](#) &right) const
- void **setDirectSourceShowfotoModel** ([ShowfotoItemModel](#) \*const sourceModel) override  
*Reimplement if needed. Called only when model shall be set as (direct) sourceModel.*
- bool **subSortLessThan** (const QModelIndex &left, const QModelIndex &right) const override

## Protected Member Functions inherited from [ShowFoto::ShowfotoSortFilterModel](#)

- void **setSourceModel** (QAbstractItemModel \*sourceModel) override

## Protected Member Functions inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override

## Protected Attributes

- [ShowfotoFilterModelPrivate](#) \*const **d\_ptr** = nullptr

## Protected Attributes inherited from [ShowFoto::ShowfotoSortFilterModel](#)

- [ShowfotoSortFilterModel](#) \* **m\_chainedModel** = nullptr

## 6.1582.1 Member Enumeration Documentation

### 6.1582.1.1 ShowfotoFilterModelRoles

```
enum ShowFoto::ShowfotoFilterModel::ShowfotoFilterModelRoles
```

## Enumerator

CategorizationModeRole	Returns the current categorization mode.
SortOrderRole	Returns the current sort order.
CategoryFormatRole	Returns the format of the index which is used for category.
ShowfotoFilterModelPointerRole	Returns true if the given showfoto item is a group leader, and the group is opened. TODO: GroupsOpenRole = ShowfotoItemModel::FilterModelRoles + 4

## 6.1582.2 Member Function Documentation

### 6.1582.2.1 categoryIdentifier()

```
QString ShowFoto::ShowfotoFilterModel::categoryIdentifier (
    const ShowfotoItemInfo & info ) const [protected], [virtual]
```

Returns a unique identifier for the category if info. The string need not be for user display.

### 6.1582.2.2 compareCategories()

```
int ShowFoto::ShowfotoFilterModel::compareCategories (
    const QModelIndex & left,
    const QModelIndex & right ) const [override], [protected], [virtual]
```

This method compares the category of the `left` index with the category of the `right` index.

Internally and if not reimplemented, this method will ask for `left` and `right` models for role `CategorySortRole`. In order to correctly sort categories, the `data()` method of the model should return a `qulonglong` (or numeric) value, or a `QString` object. `QString` objects will be sorted with `QString::localeAwareCompare` if `sortCategoriesByNaturalComparison()` is true.

#### Note

Please have present that: `QString(QChar(QChar::ObjectReplacementCharacter)) > QString(QChar(QChar::ReplacementCharacter)) > [ all possible strings ] > QString();`

This means that `QString()` will be sorted the first one, while `QString(QChar(QChar::ObjectReplacementCharacter))` and `QString(QChar(QChar::ReplacementCharacter))` will be sorted in last position.

#### Warning

Please note that `data()` method of the model should return always information of the same type. If you return a `QString` for an index, you should return always `QStrings` for all indexes for role `CategorySortRole` in order to correctly sort categories. You can't mix by returning a `QString` for one index, and a `qulonglong` for other.

#### Note

If you need a more complex layout, you will have to reimplement this method.

#### Returns

A negative value if the category of `left` should be placed before the category of `right`. 0 if `left` and `right` are on the same category, and a positive value if the category of `left` should be placed after the category of `right`.

Reimplemented from [Digikam::DCategorizedSortFilterProxyModel](#).

### 6.1582.2.3 compareInfosCategories()

```
int ShowFoto::ShowfotoFilterModel::compareInfosCategories (
    const ShowfotoItemInfo & left,
    const ShowfotoItemInfo & right ) const [protected], [virtual]
```

Reimplement to customize category sorting. Return negative if category of left < category right, Return 0 if left and right are in the same category, else return positive.

### 6.1582.2.4 infosLessThan()

```
bool ShowFoto::ShowfotoFilterModel::infosLessThan (
    const ShowfotoItemInfo & left,
    const ShowfotoItemInfo & right ) const [protected], [virtual]
```

Reimplement to customize sorting. Do not take categories into account here.

### 6.1582.2.5 setDirectSourceShowfotoModel()

```
void ShowFoto::ShowfotoFilterModel::setDirectSourceShowfotoModel (
    ShowfotoItemModel *const sourceModel ) [override], [protected], [virtual]
```

Reimplemented from [ShowFoto::ShowfotoSortFilterModel](#).

### 6.1582.2.6 showfotoFilterModel()

```
ShowfotoFilterModel * ShowFoto::ShowfotoFilterModel::showfotoFilterModel ( ) const [override],
[virtual]
```

Reimplemented from [ShowFoto::ShowfotoSortFilterModel](#).

### 6.1582.2.7 showfotoItemInfosAdded

```
void ShowFoto::ShowfotoFilterModel::showfotoItemInfosAdded (
    const QList< ShowfotoItemInfo > & infos ) [signal]
```

These signals need to be explicitly enabled with [setSendItemInfoSignals\(\)](#).

### 6.1582.2.8 subSortLessThan()

```
bool ShowFoto::ShowfotoFilterModel::subSortLessThan (
    const QModelIndex & left,
    const QModelIndex & right ) const [override], [protected], [virtual]
```

This method has a similar purpose as [lessThan\(\)](#) has on [QSortFilterProxyModel](#). It is used for sorting items that are in the same category.

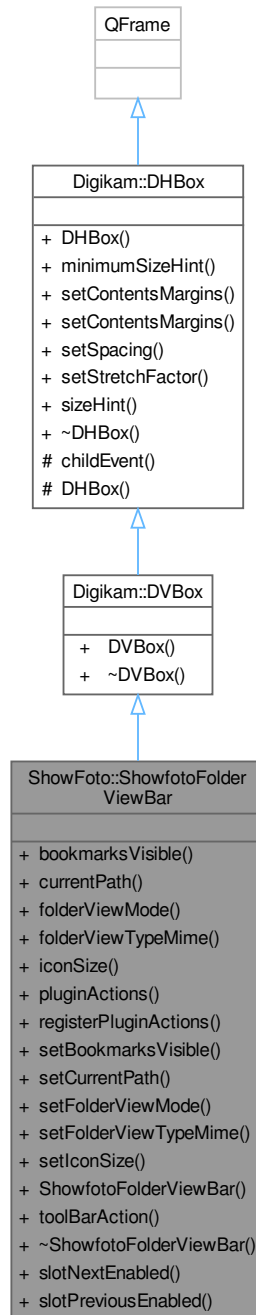
#### Returns

Returns true if the item `left` is less than the item `right` when sorting.

Reimplemented from [Digikam::DCategorizedSortFilterProxyModel](#).

## 6.1583 ShowFoto::ShowfotoFolderViewBar Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewBar:



### Public Types

- enum **FolderViewTypeMime** {  
**TYPE\_MIME\_JPEG** = 0, **TYPE\_MIME\_TIFF**, **TYPE\_MIME\_PNG**, **TYPE\_MIME\_PGF**,  
**TYPE\_MIME\_HEIF**, **TYPE\_MIME\_AVIF**, **TYPE\_MIME\_JXL**, **TYPE\_MIME\_WEBP**,  
**TYPE\_MIME\_DNG**, **TYPE\_MIME\_RAW**, **TYPE\_MIME\_NORAW**, **TYPE\_MIME\_ALL** }

## Public Slots

- void **slotNextEnabled** (bool)
- void **slotPreviousEnabled** (bool)

## Signals

- void **signalAppendContents** ()
- void **signalCustomPathChanged** (const QString &)
- void **signalGoHome** ()
- void **signalGoNext** ()
- void **signalGoPrevious** ()
- void **signalGoUp** ()
- void **signalIconSizeChanged** (int)
- void **signalLoadContents** ()
- void **signalPluginActionTriggered** (QAction \*)
- void **signalSetup** ()
- void **signalShowBookmarks** (bool)
- void **signalTypeMimesChanged** (const QString &)
- void **signalViewModeChanged** (int)

## Public Member Functions

- bool **bookmarksVisible** () const
- QString **currentPath** () const
- int **folderViewMode** () const
- int **folderViewTypeMime** () const
- int **iconSize** () const
- QList< QAction \* > **pluginActions** () const
- void **registerPluginActions** (const QList< [DPluginAction](#) \* > &actions)
- void **setBookmarksVisible** (bool b)
- void **setCurrentPath** (const QString &path)
- void **setFolderViewMode** (int mode)
- void **setFolderViewTypeMime** (int mime)
- void **setIconSize** (int size)
- [ShowfotoFolderViewBar](#) ([ShowfotoFolderViewSideBar](#) \*const parent)
- QAction \* **toolBarAction** (const QString &name) const

## Public Member Functions inherited from [Digikam::DVBox](#)

- [DVBox](#) (QWidget \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::DHBox](#)

- [DHBox](#) (QWidget \*const parent=nullptr)
- QSize **minimumSizeHint** () const override
- void **setContentsMargins** (const QMargins &margins)
- void **setContentsMargins** (int left, int top, int right, int bottom)
- void **setSpacing** (int space)
- void **setStretchFactor** (QWidget \*const widget, int stretch)
- QSize **sizeHint** () const override

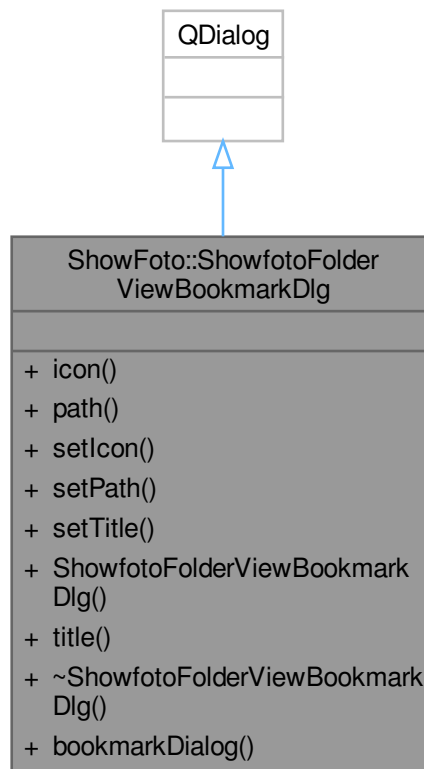
### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DHBox](#)

- void **childEvent** (QChildEvent \*e) override
- **DHBox** (bool vertical, QWidget \*const parent)

## 6.1584 ShowFoto::ShowfotoFolderViewBookmarkDlg Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewBookmarkDlg:



### Public Member Functions

- QString **icon** () const
- QString **path** () const
- void **setIcon** (const QString &icon)
- void **setPath** (const QString &path)
- void **setTitle** (const QString &title)
- **ShowfotoFolderViewBookmarkDlg** ([ShowfotoFolderViewBookmarkList](#) \*const parent, bool create=false)
- QString **title** () const

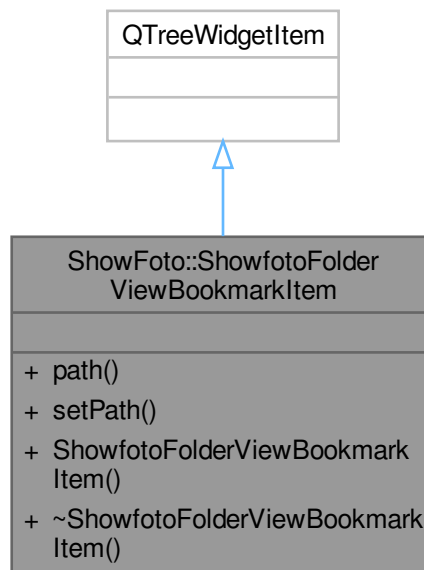


### Static Public Member Functions

- static bool **bookmarkDialog** ([ShowfotoFolderViewItemBookmarkList](#) \*const parent, QString &title, QString &icon, QString &path, bool create=false)

## 6.1585 ShowFoto::ShowfotoFolderViewItem Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewItem:

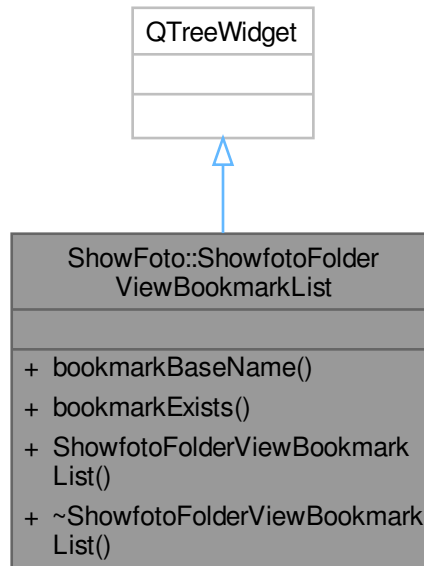


### Public Member Functions

- QString **path** () const
- void **setPath** (const QString &)
- **ShowfotoFolderViewItemBookmarkItem** (QTreeWidgetItem \*const parent)

## 6.1586 ShowFoto::ShowfotoFolderViewBookmarkList Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewBookmarkList:



### Signals

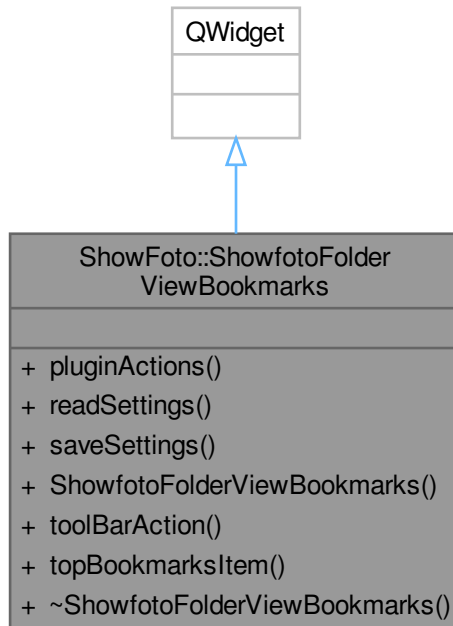
- void **signalAddBookmark** (const QString &path)
- void **signalLoadContents** (const QString &path)

### Public Member Functions

- QString **bookmarkBaseName** (const QString &path) const
- [ShowfotoFolderViewBookmarkItem](#) \* **bookmarkExists** (const QString &path) const
- **ShowfotoFolderViewBookmarkList** ([ShowfotoFolderViewBookmarks](#) \*const parent)

## 6.1587 ShowFoto::ShowfotoFolderViewBookmarks Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewBookmarks:



### Signals

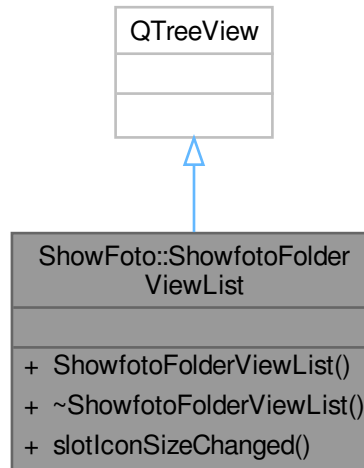
- void **signalLoadContents** ()

### Public Member Functions

- `QList< QAction * >` **pluginActions** () const
- void **readSettings** (const `KConfigGroup &`)
- void **saveSettings** (`KConfigGroup &`)
- **ShowfotoFolderViewBookmarks** (`ShowfotoFolderViewSideBar *const sidebar`)
- `QAction *` **toolBarAction** (const `QString &name`) const
- `QTreeWidgetItem *` **topBookmarksItem** () const

## 6.1588 ShowFoto::ShowfotoFolderViewList Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewList:



### Public Types

- enum `FolderViewMode` { `ShortView = 0` , `DetailedView` }
- enum `FolderViewRole` { `FileName = 0` , `FileSize` , `FileType` , `FileDate` }

### Public Slots

- void `slotIconSizeChanged` (int)

### Signals

- void `signalAddBookmark` ()

### Public Member Functions

- `ShowfotoFolderViewList` (`ShowfotoFolderViewSideBar` \*const view, `ShowfotoFolderViewBar` \*const bar)

## 6.1588.1 Member Enumeration Documentation

### 6.1588.1.1 FolderViewRole

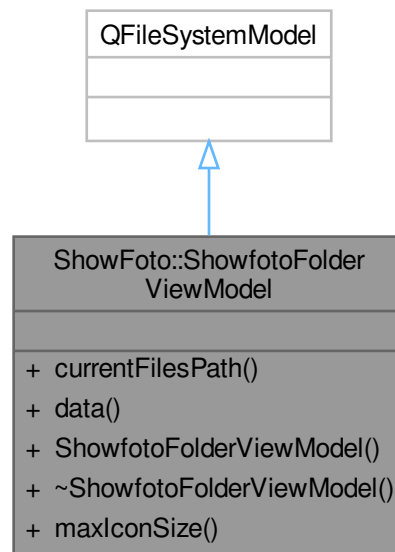
```
enum ShowFoto::ShowfotoFolderViewList::FolderViewRole
```

Enumerator

FileDate	Modifier date.
----------	----------------

## 6.1589 ShowFoto::ShowfotoFolderViewModel Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewModel:



### Public Member Functions

- `QStringList currentFilePath ()` const
- `QVariant data (const QModelIndex &index, int role)` const override
- `ShowfotoFolderViewModel (ShowfotoFolderViewList *const view)`

### Static Public Member Functions

- static int `maxIconSize ()`

## 6.1589.1 Member Function Documentation

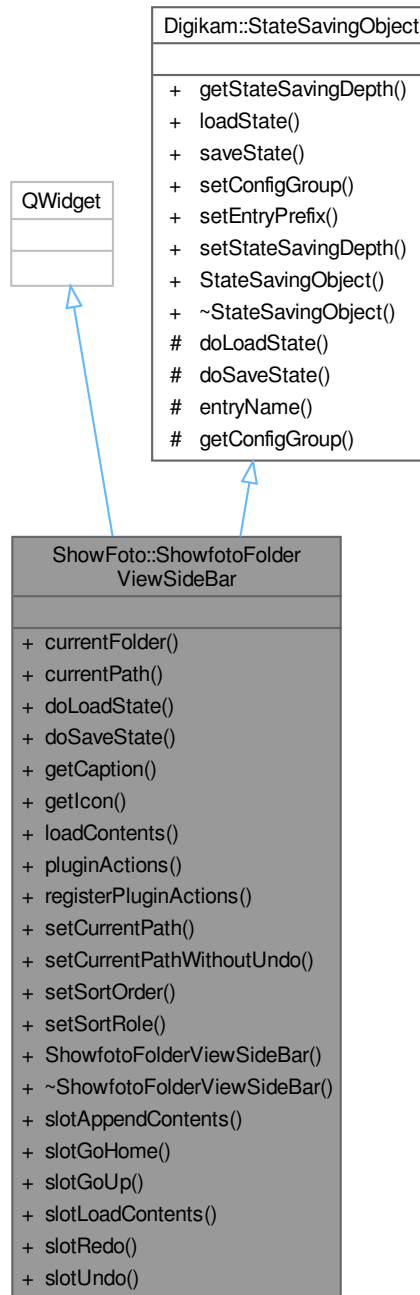
### 6.1589.1.1 currentFilePath()

```
QStringList ShowFoto::ShowfotoFolderViewModel::currentFilePath ( ) const
```

List all file paths from the current model root index selected in the view.

## 6.1590 ShowFoto::ShowfotoFolderViewSideBar Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewSideBar:



### Public Slots

- void **slotAppendContents** ()
- void **slotGoHome** ()

- void **slotGoUp** ()
- void **slotLoadContents** ()
- void **slotRedo** ()
- void **slotUndo** ()

## Signals

- void **signalAddBookmark** ()
- void **signalAppendContentsFromFiles** (const QStringList &files, const QString &current)
- void **signalLoadContentsFromFiles** (const QStringList &files, const QString &current)
- void **signalLoadContentsFromPath** (const QString &path)
- void **signalSetup** ()

## Public Member Functions

- QString **currentFolder** () const
- QString **currentPath** () const
- void **doLoadState** () override
- void **doSaveState** () override
- const QString **getCaption** ()
- const QIcon **getIcon** ()
- void **loadContents** (const QModelIndex &index, bool append=false)
- QList< QAction \* > **pluginActions** () const
- void **registerPluginActions** (const QList< DPluginAction \* > &actions)
- void **setCurrentPath** (const QString &newPathNative)
- void **setCurrentPathWithoutUndo** (const QString &newPath)
- void **setSortOrder** (int order)
- void **setSortRole** (int role)
- **ShowfotoFolderViewSideBar** ([Showfoto](#) \*const parent)

## Public Member Functions inherited from [Digikam::StateSavingObject](#)

- [StateSavingDepth](#) **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const KConfigGroup &group)
- virtual void **setEntryPrefix** (const QString &prefix)
- void **setStateSavingDepth** (const [StateSavingDepth](#) depth)
- [StateSavingObject](#) (QObject \*const host)
- virtual [~StateSavingObject](#) ()

## Additional Inherited Members

## Public Types inherited from [Digikam::StateSavingObject](#)

- enum [StateSavingDepth](#) { [INSTANCE](#) , [DIRECT\\_CHILDREN](#) , [RECURSIVE](#) }

## Protected Member Functions inherited from [Digikam::StateSavingObject](#)

- QString **entryName** (const QString &base) const
- KConfigGroup **getConfigGroup** () const

## 6.1590.1 Member Function Documentation

### 6.1590.1.1 doLoadState()

```
void ShowFoto::ShowfotoFolderViewSideBar::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1590.1.2 doSaveState()

```
void ShowFoto::ShowfotoFolderViewSideBar::doSaveState ( ) [override], [virtual]
```

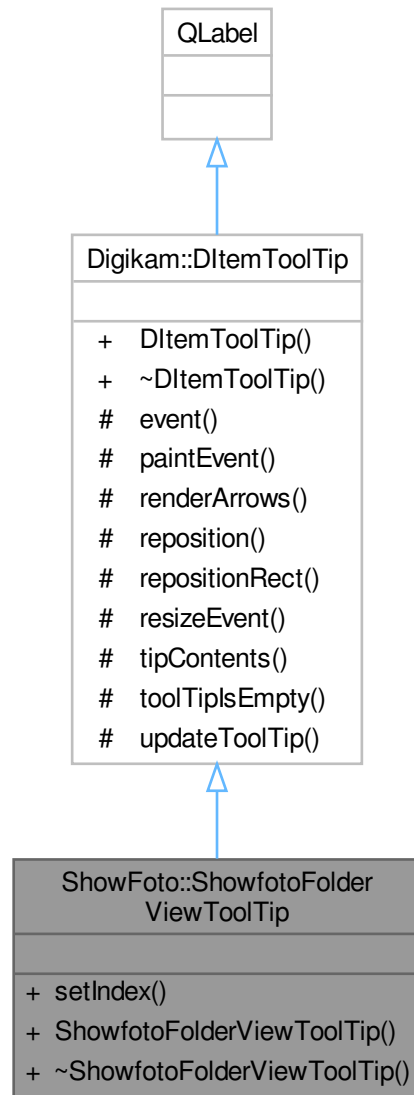
Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).



## 6.1591 ShowFoto::ShowfotoFolderViewToolTip Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewToolTip:



### Public Member Functions

- void **setIndex** (const QModelIndex &index)
- **ShowfotoFolderViewToolTip** (ShowfotoFolderViewList \*const view)

### Public Member Functions inherited from Digikam::DItemToolTip

- **DItemToolTip** (QWidget \*const parent=nullptr)

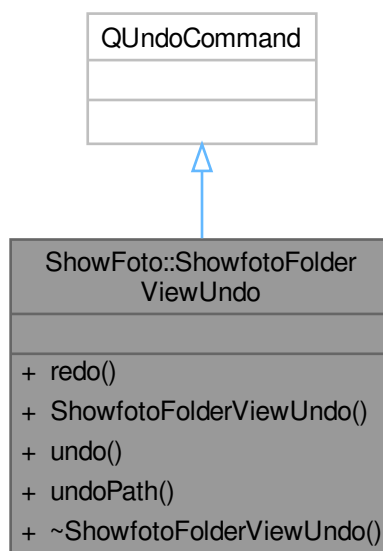
### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DItemToolTip](#)

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- void **resizeEvent** (QResizeEvent \*) override
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()

## 6.1592 ShowFoto::ShowfotoFolderViewUndo Class Reference

Inheritance diagram for ShowFoto::ShowfotoFolderViewUndo:

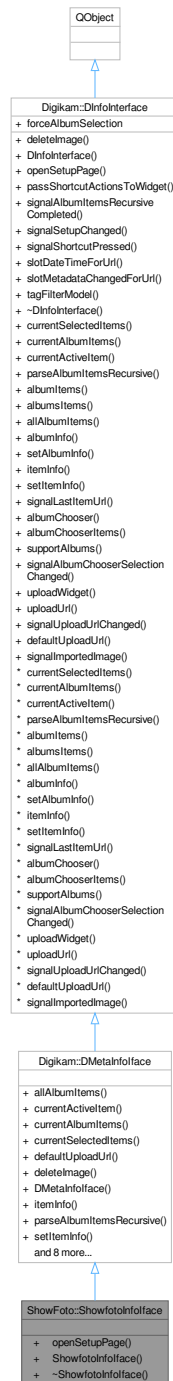


### Public Member Functions

- void **redo** ()
- **ShowfotoFolderViewUndo** ([ShowfotoFolderViewSideBar](#) \*const view, const QString &newPath)
- void **undo** ()
- QString **undoPath** () const

## 6.1593 ShowFoto::ShowfotoInfolface Class Reference

Inheritance diagram for ShowFoto::ShowfotoInfolface:



### Public Member Functions

- void [openSetupPage](#) (SetupPage page) override  
*Open configuration dialog page.*
- **ShowfotoInfolface** (QObject \*const parent, const QList< QUrl > &lst, const QUrl &currentActive)

## Public Member Functions inherited from [Digikam::DMetalInfoface](#)

- [QList< QUrl > allAlbumItems \(\)](#) const override
- [QUrl currentActiveItem \(\)](#) const override
- [QList< QUrl > currentAlbumItems \(\)](#) const override
- [QList< QUrl > currentSelectedItems \(\)](#) const override
  - Low level items and albums methods.*
- [QUrl defaultUploadUrl \(\)](#) const override
  - Url to upload new items without to use album selector.*
- void [deleteImage \(const QUrl &url\)](#) override
  - Manipulate with item.*
- **DMetalInfoface** (QObject \*const parent, const QList< QUrl > &lst, const QUrl &currentActive)
- [DInfoMap itemInfo \(const QUrl &\)](#) const override
- void [parseAlbumItemsRecursive \(\)](#) override
- void [setItemInfo \(const QUrl &, const DInfoMap &\)](#) override
- Q\_SIGNAL void **signalItemChanged** (const QUrl &url)
- Q\_SIGNAL void **signalRemoveImageFromAlbum** (const QUrl &)
- Q\_SLOT void [slotDateTimeForUrl \(const QUrl &url, const QDateTime &dt, bool updModDate\)](#) override
  - Slot to call when date time stamp from item is changed.*
- Q\_SLOT void [slotMetadataChangedForUrl \(const QUrl &url\)](#) override
  - Slot to call when something in metadata from item is changed.*
- bool [supportAlbums \(\)](#) const override
- [QUrl uploadUrl \(\)](#) const override
- QWidget \* [uploadWidget \(QWidget \\*const parent\)](#) const override
  - Album selector view methods (to upload items from an external place).*

## Public Member Functions inherited from [Digikam::DInfoInterface](#)

- **DInfoInterface** (QObject \*const parent)
- virtual QMap< QString, QString > [passShortcutActionsToWidget \(QWidget \\*const\)](#) const
  - Pass extra shortcut actions to widget and return prefixes of shortcuts.*
- Q\_SIGNAL void **signalAlbumItemsRecursiveCompleted** (const QList< QUrl > &imageList)
- Q\_SIGNAL void **signalSetupChanged** ()
- Q\_SIGNAL void **signalShortcutPressed** (const QString &shortcut, int val)
- virtual QAbstractItemModel \* [tagFilterModel \(\)](#)
  - Return an instance of tag filter model if host application support this feature, else null pointer.*
  
- virtual QList< QUrl > **albumItems (int)** const
- virtual QList< QUrl > **albumsItems (const DAlbumIDs &)** const
- virtual [DInfoMap albumInfo \(int\)](#) const
- virtual void **setAlbumInfo (int, const DInfoMap &)** const
- Q\_SIGNAL void **signalLastItemUrl** (const QUrl &)
  
- virtual QWidget \* [albumChooser \(QWidget \\*const parent\)](#) const
  - Albums chooser view methods (to use items from albums before to process).*
- virtual [DAlbumIDs albumChooserItems \(\)](#) const
- Q\_SIGNAL void **signalAlbumChooserSelectionChanged** ()
  
- Q\_SIGNAL void **signalUploadUrlChanged** ()
- Q\_SIGNAL void **signalImportedImage** (const QUrl &)

## Additional Inherited Members

### Public Types inherited from [Digikam::DInfoInterface](#)

- typedef `QList< int >` **DAlbumIDs**  
*List of [Album](#) ids.*
- typedef `QMap< QString, QVariant >` **DInfoMap**  
*Map of properties name and value.*
- enum **SetupPage** { `ExifToolPage = 0` , `ImageQualityPage` }

### Public Attributes inherited from [Digikam::DInfoInterface](#)

- bool **forceAlbumSelection** = false

## 6.1593.1 Member Function Documentation

### 6.1593.1.1 openSetupPage()

```
void ShowFoto::ShowfotoInfoIface::openSetupPage (  
    SetupPage page ) [override], [virtual]
```

Reimplemented from [Digikam::DInfoInterface](#).

## 6.1594 ShowFoto::ShowfotoItemInfo Class Reference

### Public Member Functions

- bool **isNull** () const
- bool **operator!=** (const [ShowfotoItemInfo](#) &info) const
- bool **operator==** (const [ShowfotoItemInfo](#) &info) const

### Static Public Member Functions

- static [ShowfotoItemInfo](#) **itemInfoFromFile** (const `QFileInfo` &inf)

## Public Attributes

- QDateTime **ctime**  
*camera date stamp*
- QDateTime **dtime**  
*creation time on disk*
- QString **folder**  
*Folder path to access to file.*
- int **height** = 0  
*Image height in pixels.*
- qlonglong **id** = -1  
*Unique image id.*
- QString **mime**  
*Type mime of file.*
- QString **name**  
*File name in file-system.*
- [PhotoInfoContainer](#) **photoInfo**
- qint64 **size** = -1  
*Static values.*
- QUrl **url**  
*file Url*
- int **width** = 0  
*Image width in pixels.*

## 6.1594.1 Member Function Documentation

### 6.1594.1.1 isNull()

```
bool ShowFoto::ShowfotoItemInfo::isNull ( ) const
```

Return true if all member in this container are null.

### 6.1594.1.2 operator==( )

```
bool ShowFoto::ShowfotoItemInfo::operator== (
    const ShowfotoItemInfo & info ) const
```

Compare for information equality and un-equality, not including variable values.

## 6.1594.2 Member Data Documentation

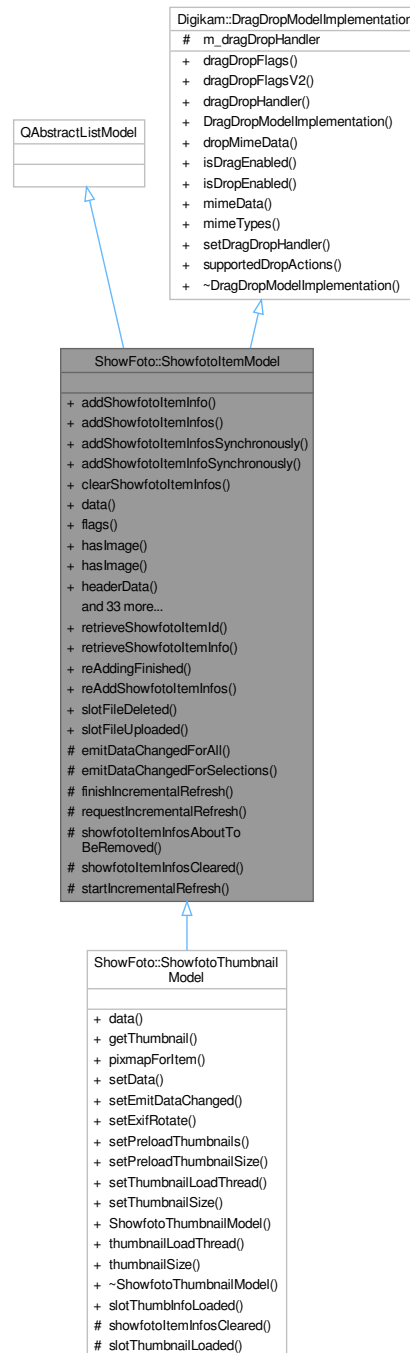
### 6.1594.2.1 size

```
qint64 ShowFoto::ShowfotoItemInfo::size = -1
```

file size in bytes.

## 6.1595 ShowFoto::ShowfotoItemModel Class Reference

Inheritance diagram for ShowFoto::ShowfotoItemModel:



### Public Types

- enum [ShowfotoItemModelRoles](#) {  
[ShowfotoItemModelPointerRole](#) = Qt::UserRole , [ShowfotoItemModelInternalId](#) = Qt::UserRole + 1 ,  
[ThumbnailRole](#) = Qt::UserRole + 2 , [ExtraDataRole](#) = Qt::UserRole + 3 ,  
[ExtraDataDuplicateCount](#) = Qt::UserRole + 6 , [FilterModelRoles](#) = Qt::UserRole + 100 }

## Public Slots

- void **reAddingFinished** ()
- void **reAddShowfotoItemInfos** (const ShowfotoItemInfoList &infos)
- void **slotFileDeleted** (const QString &folder, const QString &file, bool status)
- void **slotFileUploaded** (const ShowfotoItemInfo &info)

## Signals

- void **allRefreshingFinished** ()
- void **itemInfosAboutToBeAdded** (const QList< ShowfotoItemInfo > &infos)
- void **itemInfosAboutToBeRemoved** (const QList< ShowfotoItemInfo > &infos)
- void **itemInfosAdded** (const QList< ShowfotoItemInfo > &infos)
- void **itemInfosRemoved** (const QList< ShowfotoItemInfo > &infos)
- void **preprocess** (const QList< ShowfotoItemInfo > &infos)
- void **processAdded** (const QList< ShowfotoItemInfo > &infos)
- void **readyForIncrementalRefresh** ()

## Public Member Functions

- void **addShowfotoItemInfo** (const ShowfotoItemInfo &info)
- void **addShowfotoItemInfos** (const QList< ShowfotoItemInfo > &infos)
- void **addShowfotoItemInfosSynchronously** (const QList< ShowfotoItemInfo > &infos)
- void **addShowfotoItemInfoSynchronously** (const ShowfotoItemInfo &info)
- void **clearShowfotoItemInfos** ()
- QVariant **data** (const QModelIndex &index, int role) const override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **hasImage** (const ShowfotoItemInfo &info) const
- bool **hasImage** (qulonglong id) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role) const override
- QModelIndex **index** (int row, int column, const QModelIndex &parent) const override
- QList< QModelIndex > **indexesForShowfotoItemId** (qulonglong id) const
- QList< QModelIndex > **indexesForShowfotoItemInfo** (const ShowfotoItemInfo &info) const
- QList< QModelIndex > **indexesForUrl** (const QUrl &fileUrl) const
- QModelIndex **indexForShowfotoItemId** (qulonglong id) const
- QModelIndex **indexForShowfotoItemInfo** (const ShowfotoItemInfo &info) const
- QModelIndex **indexForUrl** (const QUrl &fileUrl) const
- bool **isEmpty** () const
- int **numberOfIndexesForShowfotoItemId** (qulonglong id) const
- int **numberOfIndexesForShowfotoItemInfo** (const ShowfotoItemInfo &info) const
- void **removeIndex** (const QModelIndex &index)
- void **removeIndexes** (const QList< QModelIndex > &indexes)
- void **removeShowfotoItemInfo** (const ShowfotoItemInfo &info)
- void **removeShowfotoItemInfos** (const QList< ShowfotoItemInfo > &infos)
- int **rowCount** (const QModelIndex &parent) const override
- void **setKeepsFileUrlCache** (bool keepCache)
- DECLARE\_MODEL\_DRAG\_DROP\_METHODS void **setSendRemovalSignals** (bool send)
- void **setShowfotoItemInfos** (const QList< ShowfotoItemInfo > &infos)
- qulonglong **showfotoItemId** (const QModelIndex &index) const
- qulonglong **showfotoItemId** (int row) const
- QList< qulonglong > **showfotoItemIds** () const
- QList< qulonglong > **showfotoItemIds** (const QList< QModelIndex > &indexes) const
- ShowfotoItemInfo **showfotoItemInfo** (const QModelIndex &index) const



- [ShowfotoItemInfo](#) **showfotoItemInfo** (const QUrl &fileUrl) const
- [ShowfotoItemInfo](#) **showfotoItemInfo** (int row) const
- [ShowfotoItemInfo](#) & **showfotoItemInfoRef** (const QModelIndex &index) const
- [ShowfotoItemInfo](#) & **showfotoItemInfoRef** (int row) const
- QList< [ShowfotoItemInfo](#) > **showfotoItemInfos** () const
- ShowfotoItemInfoList **showfotoItemInfos** (const QList< QModelIndex > &indexes) const
- QList< [ShowfotoItemInfo](#) > **showfotoItemInfos** (const QUrl &fileUrl) const
- **ShowfotoItemModel** (QObject \*const parent)
- QList< [ShowfotoItemInfo](#) > **uniqueShowfotoItemInfos** () const

## Public Member Functions inherited from [Digikam::DragDropModelImplementation](#)

- virtual Qt::ItemFlags **dragDropFlags** (const QModelIndex &index) const
- Qt::ItemFlags **dragDropFlagsV2** (const QModelIndex &index) const
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const
- [DragDropModelImplementation](#) ()=default
- bool **dropMimeData** (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool **isDragEnabled** (const QModelIndex &index) const
- virtual bool **isDropEnabled** (const QModelIndex &index) const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const
- QStringList **mimeTypes** () const
- void **setDragDropHandler** ([AbstractItemDragDropHandler](#) \*handler)
- Qt::DropActions **supportedDropActions** () const

## Static Public Member Functions

- static qlonglong **retrieveShowfotoItemId** (const QModelIndex &index)
- static [ShowfotoItemInfo](#) **retrieveShowfotoItemInfo** (const QModelIndex &index)

## Protected Member Functions

- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelections** (const QItemSelection &selection)
- void **finishIncrementalRefresh** ()
- void **requestIncrementalRefresh** ()
- virtual void **showfotoItemInfosAboutToBeRemoved** (int, int)
- virtual void **showfotoItemInfosCleared** ()
- void **startIncrementalRefresh** ()

## Additional Inherited Members

## Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr

## 6.1595.1 Member Enumeration Documentation

### 6.1595.1.1 ShowfotoItemModelRoles

```
enum ShowFoto::ShowfotoItemModel::ShowfotoItemModelRoles
```

## Enumerator

ShowfotoItemModelPointerRole	An ShowfotoItemModel* pointer to this model
ThumbnailRole	Returns a thumbnail pixmap. May be implemented by subclasses. Returns either a valid pixmap or a null QVariant.
ExtraDataRole	Return (optional) extraData field
ExtraDataDuplicateCount	Returns the number of duplicate indexes for the same image id

## 6.1595.2 Member Function Documentation

### 6.1595.2.1 addShowfotoItemInfoSynchronously()

```
void ShowFoto::ShowfotoItemModel::addShowfotoItemInfoSynchronously (
    const ShowfotoItemInfo & info )
```

addShowfotoItemInfo() is asynchronous if a preprocessor is set. This method first adds the info, synchronously. Only afterwards, the preprocessor will have the opportunity to process it. This method also bypasses any incremental updates.

### 6.1595.2.2 allRefreshingFinished

```
void ShowFoto::ShowfotoItemModel::allRefreshingFinished ( ) [signal]
```

Signals that the model has finished currently with all scheduled refreshing, full or incremental, and all preprocessing. The model is in polished, clean situation right now.

### 6.1595.2.3 clearShowfotoItemInfos()

```
void ShowFoto::ShowfotoItemModel::clearShowfotoItemInfos ( )
```

Clears the ShowfotoItemInfos and resets the model.

### 6.1595.2.4 indexForShowfotoItemInfo()

```
QModelIndex ShowFoto::ShowfotoItemModel::indexForShowfotoItemInfo (
    const ShowfotoItemInfo & info ) const
```

Return the index of a given [ShowfotoItemInfo](#), if it exists in the model.

### 6.1595.2.5 indexForUrl()

```
QModelIndex ShowFoto::ShowfotoItemModel::indexForUrl (
    const QUrl & fileUrl ) const
```

Returns the index or [ShowfotoItemInfo](#) object from the underlying data for the given file url. In case of multiple occurrences of the same file, the simpler overrides returns any one found first, use the QList methods to retrieve all occurrences.

### 6.1595.2.6 itemInfosAboutToBeAdded

```
void ShowFoto::ShowfotoItemModel::itemInfosAboutToBeAdded (
    const QList< ShowfotoItemInfo > & infos ) [signal]
```

Informs that ItemInfos will be added to the model. This signal is sent before the model data is changed and views are informed.

### 6.1595.2.7 itemInfosAboutToBeRemoved

```
void ShowFoto::ShowfotoItemModel::itemInfosAboutToBeRemoved (
    const QList< ShowfotoItemInfo > & infos ) [signal]
```

Informs that ShowfotoItemInfos will be removed from the model. This signal is sent before the model data is changed and views are informed. Note: You need to explicitly enable sending of this signal. It is not sent in [clearShowfotoItemInfos\(\)](#).

### 6.1595.2.8 itemInfosAdded

```
void ShowFoto::ShowfotoItemModel::itemInfosAdded (
    const QList< ShowfotoItemInfo > & infos ) [signal]
```

Informs that ItemInfos have been added to the model. This signal is sent after the model data is changed and views are informed.

### 6.1595.2.9 itemInfosRemoved

```
void ShowFoto::ShowfotoItemModel::itemInfosRemoved (
    const QList< ShowfotoItemInfo > & infos ) [signal]
```

Informs that ShowfotoItemInfos have been removed from the model. This signal is sent after the model data is changed and views are informed. Note: You need to explicitly enable sending of this signal. It is not sent in [clearShowfotoItemInfos\(\)](#).

### 6.1595.2.10 preprocess

```
void ShowFoto::ShowfotoItemModel::preprocess (
    const QList< ShowfotoItemInfo > & infos ) [signal]
```

Connect to this signal only if you are the current preprocessor.

### 6.1595.2.11 readyForIncrementalRefresh

```
void ShowFoto::ShowfotoItemModel::readyForIncrementalRefresh ( ) [signal]
```

Signals that the model is right now ready to start an incremental refresh. This is guaranteed only for the scope of emitting this signal.

### 6.1595.2.12 removeIndex()

```
void ShowFoto::ShowfotoItemModel::removeIndex (
    const QModelIndex & index )
```

Remove the given infos or indexes directly from the model.

### 6.1595.2.13 requestIncrementalRefresh()

```
void ShowFoto::ShowfotoItemModel::requestIncrementalRefresh ( ) [protected]
```

As soon as the model is ready to start an incremental refresh, the signal [readyForIncrementalRefresh\(\)](#) will be emitted. The signal will be emitted inline if the model is ready right now.

### 6.1595.2.14 retrieveShowfotoItemInfo()

```
ShowfotoItemInfo ShowFoto::ShowfotoItemModel::retrieveShowfotoItemInfo (
    const QModelIndex & index ) [static]
```

Retrieve the [ShowfotoItemInfo](#) object from the data() function of the given index The index may be from a [QSortFilterProxyModel](#) as long as an [ShowfotoItemModel](#) is at the end.

### 6.1595.2.15 rowCount()

```
int ShowFoto::ShowfotoItemModel::rowCount (
    const QModelIndex & parent ) const [override]
```

QAbstractListModel implementations

### 6.1595.2.16 setKeepsFileUrlCache()

```
void ShowFoto::ShowfotoItemModel::setKeepsFileUrlCache (
    bool keepCache )
```

If a cache is kept, lookup by file path is fast, without a cache it is O(n). Default is false.

### 6.1595.2.17 setSendRemovalSignals()

```
void ShowFoto::ShowfotoItemModel::setSendRemovalSignals (
    bool send )
```

DragDrop methods Enable sending of [itemInfosAboutToBeRemoved](#) and [itemsInfosRemoved](#) signals. Default: false

### 6.1595.2.18 setShowfotoItemInfos()

```
void ShowFoto::ShowfotoItemModel::setShowfotoItemInfos (
    const QList< ShowfotoItemInfo > & infos )
```

Clears and adds infos.

**6.1595.2.19 showfotoItemInfo()** [1/2]

```
ShowfotoItemInfo ShowFoto::ShowfotoItemModel::showfotoItemInfo (
    const QModelIndex & index ) const
```

Returns the [ShowfotoItemInfo](#) object, reference from the underlying data pointed to by the index. For [ShowfotoItemInfo](#) and [ShowfotoItemInfoRef](#) If the index is not valid they will return a null [ShowfotoItemInfo](#), and 0 respectively, [ShowfotoItemInfoRef](#) must not be called with an invalid index as it will crash.

**6.1595.2.20 showfotoItemInfo()** [2/2]

```
ShowfotoItemInfo ShowFoto::ShowfotoItemModel::showfotoItemInfo (
    int row ) const
```

Returns the [ShowfotoItemInfo](#) object, reference from the underlying data of the given row (parent is the invalid QModelIndex, column is 0). Note that [ShowfotoItemInfoRef](#) must not be called with an invalid index as it will crash.

**6.1595.2.21 showfotoItemInfosAboutToBeRemoved()**

```
virtual void ShowFoto::ShowfotoItemModel::showfotoItemInfosAboutToBeRemoved (
    int ,
    int ) [inline], [protected], [virtual]
```

Called before [rowsAboutToBeRemoved](#)

**6.1595.2.22 showfotoItemInfosCleared()**

```
virtual void ShowFoto::ShowfotoItemModel::showfotoItemInfosCleared ( ) [inline], [protected],
[virtual]
```

Called when the internal storage is cleared.

Reimplemented in [ShowFoto::ShowfotoThumbnailModel](#).

**6.1595.2.23 startIncrementalRefresh()**

```
void ShowFoto::ShowfotoItemModel::startIncrementalRefresh ( ) [protected]
```

Starts an incremental refresh operation. You shall only call this method from a slot connected to [readyForIncrementalRefresh\(\)](#). To initiate an incremental refresh, call [requestIncrementalRefresh\(\)](#).

**6.1596 ShowFoto::ShowfotoItemSortSettings Class Reference****Public Types**

- enum **CategorizationMode** { **NoCategories** , **CategoryByFolder** , **CategoryByFormat** }
- enum **SortOrder** { **AscendingOrder** = Qt::AscendingOrder , **DescendingOrder** = Qt::DescendingOrder , **DefaultOrder** }
- enum **SortRole** { **SortByCreationDate** , **SortByFileName** , **SortByFileSize** }

## Public Member Functions

- int [compare](#) (const [ShowfotoItemInfo](#) &left, const [ShowfotoItemInfo](#) &right) const
- int **compare** (const [ShowfotoItemInfo](#) &left, const [ShowfotoItemInfo](#) &right, SortRole sortRole) const
- int [compareCategories](#) (const [ShowfotoItemInfo](#) &left, const [ShowfotoItemInfo](#) &right) const
- bool **isCategorized** () const
- bool [lessThan](#) (const QVariant &left, const QVariant &right) const
- bool [lessThan](#) (const [ShowfotoItemInfo](#) &left, const [ShowfotoItemInfo](#) &right) const
- bool **operator==** (const [ShowfotoItemSortSettings](#) &other) const
- void **setCategorizationMode** (CategorizationMode mode)
  - *Categories* -----
- void **setCategorizationSortOrder** ([SortOrder](#) order)
- void **setSortOrder** ([SortOrder](#) order)
- void **setSortRole** (SortRole role)
  - *Showfoto Items Sorting* -----

## Static Public Member Functions

- template<typename T >  
static int **compareByOrder** (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int [compareByOrder](#) (int compareResult, Qt::SortOrder sortOrder)
- template<typename T >  
static int [compareValue](#) (const T &a, const T &b)
- static Qt::SortOrder **defaultSortOrderForCategorizationMode** (CategorizationMode mode)
- static Qt::SortOrder **defaultSortOrderForSortRole** (SortRole role)
- template<typename T >  
static bool [lessThanByOrder](#) (const T &a, const T &b, Qt::SortOrder sortOrder)
- static int [naturalCompare](#) (const QString &a, const QString &b, Qt::SortOrder sortOrder, Qt::CaseSensitivity caseSensitive=Qt::CaseSensitive)

## Public Attributes

- Qt::CaseSensitivity **categorizationCaseSensitivity** = Qt::CaseSensitive
- CategorizationMode **categorizationMode** = NoCategories
- [SortOrder](#) **categorizationSortOrder** = [DefaultOrder](#)
- Qt::SortOrder **currentCategorizationSortOrder** = Qt::AscendingOrder
- Qt::SortOrder **currentSortOrder** = Qt::AscendingOrder
- Qt::CaseSensitivity **sortCaseSensitivity** = Qt::CaseSensitive
- [SortOrder](#) **sortOrder** = [DefaultOrder](#)
- SortRole **sortRole** = SortByFileName

## 6.1596.1 Member Enumeration Documentation

### 6.1596.1.1 SortOrder

enum [ShowFoto::ShowfotoItemSortSettings::SortOrder](#)

#### Enumerator

DefaultOrder	sort order depends on the chosen sort role
--------------	--

## 6.1596.2 Member Function Documentation

### 6.1596.2.1 compare()

```
int ShowFoto::ShowfotoItemSortSettings::compare (
    const ShowfotoItemInfo & left,
    const ShowfotoItemInfo & right ) const
```

Compares the showfotoItemInfos left and right. Return -1 if left is less than right, 1 if left is greater than right, and 0 if left equals right comparing the current sort role's value. Adheres to set sort role and sort order.

### 6.1596.2.2 compareByOrder()

```
static int ShowFoto::ShowfotoItemSortSettings::compareByOrder (
    int compareResult,
    Qt::SortOrder sortOrder ) [inline], [static]
```

Takes a typical result from a compare method (0 is equal, -1 is less than, 1 is greater than) and applies the given sort order to it.

### 6.1596.2.3 compareCategories()

```
int ShowFoto::ShowfotoItemSortSettings::compareCategories (
    const ShowfotoItemInfo & left,
    const ShowfotoItemInfo & right ) const
```

Compares the categories of left and right ShowfotoItemInfos. It returns -1 if the left [ShowfotoItemInfo](#) is less than right, and 0 if both fall in the same category, and 1 if the left [ShowfotoItemInfo](#) is greater than right. Adheres to set categorization mode and current category sort order.

### 6.1596.2.4 compareValue()

```
template<typename T >
static int ShowFoto::ShowfotoItemSortSettings::compareValue (
    const T & a,
    const T & b ) [inline], [static]
```

Returns the usual compare result of -1, 0, or 1 for lessThan, equals and greaterThan.

### 6.1596.2.5 lessThan() [1/2]

```
bool ShowFoto::ShowfotoItemSortSettings::lessThan (
    const QVariant & left,
    const QVariant & right ) const
```

Returns true if left QVariant is less than right. Adheres to current sort role and sort order.

### 6.1596.2.6 lessThan() [2/2]

```
bool ShowFoto::ShowfotoItemSortSettings::lessThan (
    const ShowfotoItemInfo & left,
    const ShowfotoItemInfo & right ) const
```

Returns true if left is less than right. Adheres to current sort role and sort order.

### 6.1596.2.7 lessThanByOrder()

```
template<typename T >
static bool ShowFoto::ShowfotoItemSortSettings::lessThanByOrder (
    const T & a,
    const T & b,
    Qt::SortOrder sortOrder ) [inline], [static]
```

Returns  $a < b$  if sortOrder is Ascending, or  $b < a$  if order is descending

### 6.1596.2.8 naturalCompare()

```
static int ShowFoto::ShowfotoItemSortSettings::naturalCompare (
    const QString & a,
    const QString & b,
    Qt::SortOrder sortOrder,
    Qt::CaseSensitivity caseSensitive = Qt::CaseSensitive ) [inline], [static]
```

Compares the two string by natural comparison and adheres to given sort order

## 6.1596.3 Member Data Documentation

### 6.1596.3.1 currentCategorizationSortOrder

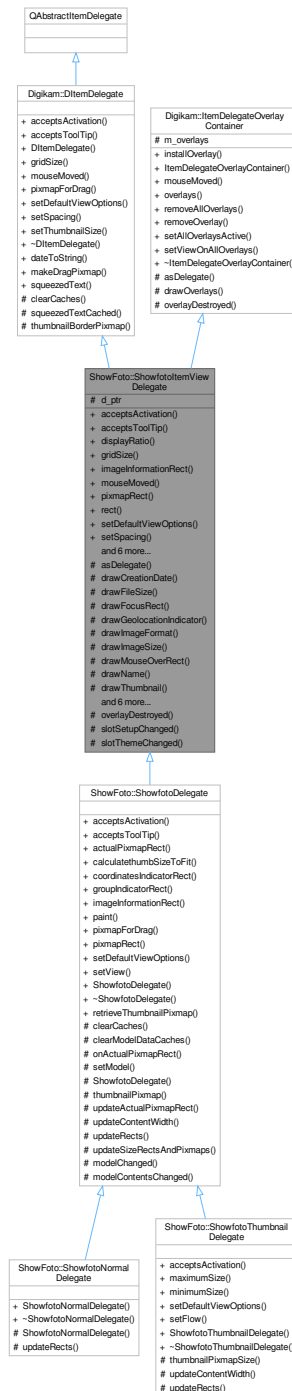
```
Qt::SortOrder ShowFoto::ShowfotoItemSortSettings::currentCategorizationSortOrder = Qt::←
AscendingOrder
```

Only Ascending or Descending, never be DefaultOrder



## 6.1597 ShowFoto::ShowfotoItemViewDelegate Class Reference

Inheritance diagram for ShowFoto::ShowfotoItemViewDelegate:



### Signals

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

## Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

## Public Member Functions

- bool **acceptsActivation** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool **acceptsToolTip** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double **displayRatio** () const
- QSize **gridSize** () const override
- virtual QRect **imageInformationRect** () const
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- virtual QRect **pixmapRect** () const
- QRect **rect** () const
- void **setDefaultViewOptions** (const QStyleOptionViewItem &option) override
- void **setSpacing** (int spacing) override
- void **setThumbnailSize** (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from DItemDelegate*
- **ShowfotoItemViewDelegate** (QWidget \*const parent)
- QSize **sizeHint** (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int **spacing** () const
- [ThumbnailSize](#) **thumbnailSize** () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- **DItemDelegate** (QObject \*const parent=nullptr)
- virtual QPixmap **pixmapForDrag** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const =0

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void **installOverlay** ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void **mouseMoved** (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > **overlays** () const
- void **removeAllOverlays** ()
- void **removeOverlay** ([ItemDelegateOverlay](#) \*overlay)
- void **setAllOverlaysActive** (bool active)
- void **setViewOnAllOverlays** (QAbstractItemView \*view)

## Protected Slots

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

## Protected Member Functions

- `QAbstractItemDelegate * asDelegate ()` override  
*Returns the delegate, typically, the derived class.*
- void **drawCreationDate** (`QPainter *p`, `const QRect &dateRect`, `const QDateTime &date`) const
- void **drawFileSize** (`QPainter *p`, `const QRect &r`, `qulonglong bytes`) const
- void **drawFocusRect** (`QPainter *p`, `const QStyleOptionViewItem &option`, `bool isSelected`) const
- void **drawGeolocationIndicator** (`QPainter *p`, `const QRect &r`) const
- void **drawImageFormat** (`QPainter *p`, `const QRect &dimsRect`, `const QString &mime`) const
- void **drawImageSize** (`QPainter *p`, `const QRect &dimsRect`, `const QSize &dims`) const
- void **drawMouseOverRect** (`QPainter *p`, `const QStyleOptionViewItem &option`) const
- void **drawName** (`QPainter *p`, `const QRect &nameRect`, `const QString &name`) const
- `QRect drawThumbnail` (`QPainter *p`, `const QRect &thumbRect`, `const QPixmap &background`, `const QPixmap &thumbnail`) const  
*Use the tool methods for painting in subclasses.*
- virtual void **invalidatePaintingCache** ()  
*reimplement these in subclasses*
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (`int maxWidth`)
- **ShowfotoItemViewDelegate** (`ShowfotoItemViewDelegatePrivate &dd`, `QWidget *const parent`)
- virtual void **updateSizeRectsAndPixmaps** ()=0

## Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- virtual void **clearCaches** ()
- `QString squeezedTextCached` (`QPainter *const p`, `int width`, `const QString &text`) const
- `QPixmap thumbnailBorderPixmap` (`const QSize &pixSize`, `bool isGrouped=false`) const

## Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (`QPainter *p`, `const QStyleOptionViewItem &option`, `const QModelIndex &index`) const
- virtual void **overlayDestroyed** (`QObject *o`)  
*Declare as slot in the derived class calling this method.*

## Protected Attributes

- `ShowfotoItemViewDelegatePrivate *const d_ptr` = nullptr

## Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- `QList< ItemDelegateOverlay * > m_overlays`

## Additional Inherited Members

## Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static `QString dateToString` (`const QDateTime &datetime`)
- static `QPixmap makeDragPixmap` (`const QStyleOptionViewItem &option`, `const QList< QModelIndex > &indexes`, `double displayRatio`, `const QPixmap &suggestedPixmap=QPixmap()`)
- static `QString squeezedText` (`const QFontMetrics &fm`, `int width`, `const QString &text`)

## 6.1597.1 Member Function Documentation

### 6.1597.1.1 acceptsActivation()

```
bool ShowFoto::ShowfotoItemViewDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect = nullptr ) const [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

### 6.1597.1.2 acceptsToolTip()

```
bool ShowFoto::ShowfotoItemViewDelegate::acceptsToolTip (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * tooltipRect = nullptr ) const [override], [virtual]
```

These methods take four parameters: The position on viewport, the rect on viewport, the index, and optionally a parameter into which, if the return value is true, a rectangle can be written for which the return value will be true as well.

Implements [Digikam::DItemDelegate](#).

### 6.1597.1.3 asDelegate()

```
QAbstractItemDelegate * ShowFoto::ShowfotoItemViewDelegate::asDelegate ( ) [override], [protected], [virtual]
```

Implements [Digikam::ItemDelegateOverlayContainer](#).

### 6.1597.1.4 gridSize()

```
QSize ShowFoto::ShowfotoItemViewDelegate::gridSize ( ) const [override], [virtual]
```

Returns the gridsize to be set by the view. It's sizeHint plus spacing.

Implements [Digikam::DItemDelegate](#).

### 6.1597.1.5 imageInformationRect()

```
QRect ShowFoto::ShowfotoItemViewDelegate::imageInformationRect ( ) const [virtual]
```

Returns the area where the image information is drawn, or null if empty / not supported. The image information is textual or graphical information, but not the pixmap. The `ratingRect()` will e.g. typically be contained in this area.

Reimplemented in [ShowFoto::ShowfotoDelegate](#).

### 6.1597.1.6 mouseMoved()

```
void ShowFoto::ShowfotoItemViewDelegate::mouseMoved (
    QMouseEvent * e,
    const QRect & visualRect,
    const QModelIndex & index ) [override], [virtual]
```

#### Note

to be called by ItemViewCategorized only

Implements [Digikam::DItemDelegate](#).

### 6.1597.1.7 pixmapRect()

```
QRect ShowFoto::ShowfotoItemViewDelegate::pixmapRect ( ) const [virtual]
```

Returns the area where the pixmap is drawn, or null if not supported

Reimplemented in [ShowFoto::ShowfotoDelegate](#).

### 6.1597.1.8 setDefaultViewOptions()

```
void ShowFoto::ShowfotoItemViewDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. option.rect shall be the viewport rectangle. Call on resize, font change.

Implements [Digikam::DItemDelegate](#).

Reimplemented in [ShowFoto::ShowfotoThumbnailDelegate](#).

### 6.1597.1.9 setSpacing()

```
void ShowFoto::ShowfotoItemViewDelegate::setSpacing (
    int spacing ) [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

### 6.1597.1.10 setThumbnailSize()

```
void ShowFoto::ShowfotoItemViewDelegate::setThumbnailSize (
    const ThumbnailSize & thumbSize ) [override], [virtual]
```

Implements [Digikam::DItemDelegate](#).

## 6.1598 ShowFoto::ShowfotoItemViewDelegatePrivate Class Reference

Inheritance diagram for ShowFoto::ShowfotoItemViewDelegatePrivate:



### Public Member Functions

- virtual void `clearRects()`
  - Resets cached rects. Remember to reimplement in subclass for added rects.*
- void `init>ShowfotoItemViewDelegatePrivate *const _q, QWidget *const _widget)`

## Public Attributes

- QWidget \* **displayWidget** = nullptr
- QFont **font**
- QFont **fontCom**
- QFont **fontReg**
- QFont **fontXtra**
- QSize **gridSize**
- int **margin** = 5
- QRect **oneRowComRect**
- QRect **oneRowRegRect**
- QRect **oneRowXtraRect**
- [ShowfotoItemViewDelegate](#) \* **q** = nullptr
- int **radius** = 3
  - constant values for drawing*
- QVector< QPixmap > **ratingPixmap**
- QRect **rect**
- QPixmap **regPixmap**
- QPixmap **selPixmap**
- int **spacing** = 0
- [ThumbnailSize](#) **thumbSize** = [ThumbnailSize](#)(0)

## 6.1598.1 Member Function Documentation

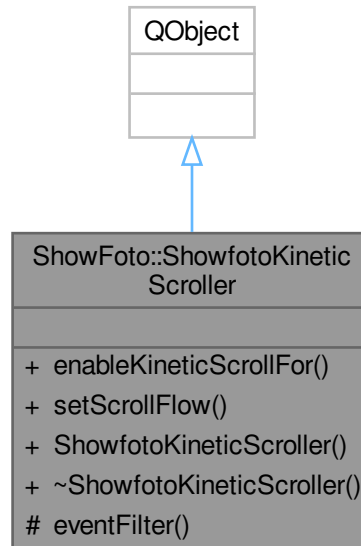
### 6.1598.1.1 clearRects()

```
void ShowFoto::ShowfotoItemViewDelegatePrivate::clearRects ( ) [virtual]
```

Reimplemented in [ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate](#).

## 6.1599 ShowFoto::ShowfotoKineticScroller Class Reference

Inheritance diagram for ShowFoto::ShowfotoKineticScroller:



### Public Member Functions

- void [enableKineticScrollFor](#) (QAbstractScrollArea \*const scrollArea)
- void **setScrollFlow** (QListView::Flow flow)
- **ShowfotoKineticScroller** (QObject \*const parent=nullptr)

### Protected Member Functions

- bool **eventFilter** (QObject \*object, QEvent \*event) override  
*intercepts mouse events to make the scrolling work*

### 6.1599.1 Detailed Description

Vertical kinetic scroller implementation without overshoot and bouncing. A temporary solution to get kinetic-like scrolling on Symbian.

### 6.1599.2 Member Function Documentation

#### 6.1599.2.1 enableKineticScrollFor()

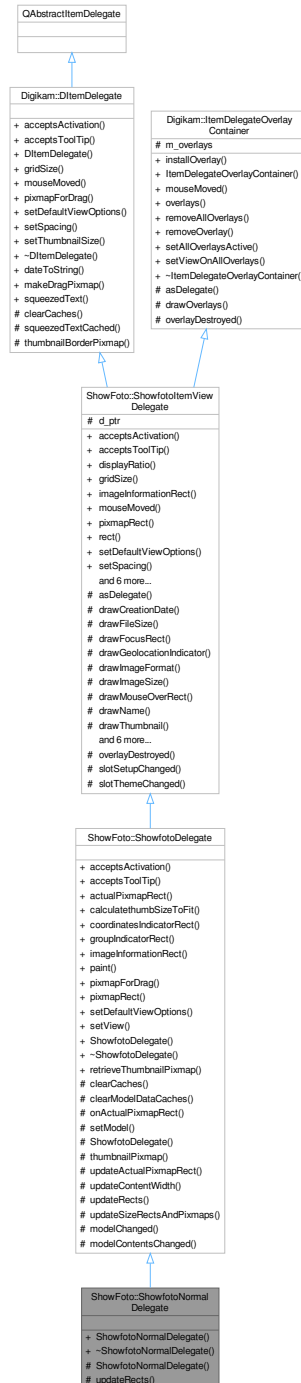
```
void ShowFoto::ShowfotoKineticScroller::enableKineticScrollFor (
    QAbstractScrollArea *const scrollArea )
```

NOTE: enabled for one widget only, new calls remove previous association



## 6.1600 ShowFoto::ShowfotoNormalDelegate Class Reference

Inheritance diagram for ShowFoto::ShowfotoNormalDelegate:



### Public Member Functions

- [ShowfotoNormalDelegate](#) ([ShowfotoThumbnailBar](#) \*const bar, QObject \*const parent=nullptr)

## Public Member Functions inherited from [ShowFoto::ShowfotoDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect [actualPixmapRect](#) (const QModelIndex &index) const
- int [calculatethumbSizeToFit](#) (int ws)
- QRect [coordinatesIndicatorRect](#) () const
- QRect [groupIndicatorRect](#) () const
- QRect [imageInformationRect](#) () const override
- void [paint](#) (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setView](#) ([ShowfotoThumbnailBar](#) \*view)
- [ShowfotoDelegate](#) (QWidget \*const parent)

## Public Member Functions inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- bool [acceptsActivation](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect=nullptr) const override
- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double [displayRatio](#) () const
- QSize [gridSize](#) () const override
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- QRect [rect](#) () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from [DItemDelegate](#)*
- [ShowfotoItemViewDelegate](#) (QWidget \*const parent)
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int [spacing](#) () const
- [ThumbnailSize](#) [thumbnailSize](#) () const

## Public Member Functions inherited from [Digikam::DItemDelegate](#)

- [DItemDelegate](#) (QObject \*const parent=nullptr)

## Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void [installOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > [overlays](#) () const
- void [removeAllOverlays](#) ()
- void [removeOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- void [setAllOverlaysActive](#) (bool active)
- void [setViewOnAllOverlays](#) (QAbstractItemView \*view)

**Protected Member Functions**

- **ShowfotoNormalDelegate** ([ShowfotoNormalDelegatePrivate](#) &dd, [ShowfotoThumbnailBar](#) \*const bar, QObject \*const parent=nullptr)
- void [updateRects](#) () override

**Protected Member Functions inherited from [ShowFoto::ShowfotoDelegate](#)**

- void [clearCaches](#) () override
- virtual void [clearModelDataCaches](#) ()
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- **ShowfotoDelegate** ([ShowfotoDelegate::ShowfotoDelegatePrivate](#) &dd, QWidget \*const parent)
- virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- virtual void [updateContentWidth](#) ()
- void [updateSizeRectsAndPxmmaps](#) () override

**Protected Member Functions inherited from [ShowFoto::ShowfotoItemViewDelegate](#)**

- QAbstractItemDelegate \* [asDelegate](#) () override  
*Returns the delegate, typically, the derived class.*
- void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
- void **drawImageFormat** (QPainter \*p, const QRect &dimsRect, const QString &mime) const
- void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail) const  
*Use the tool methods for painting in subclasses.*
- virtual void **invalidatePaintingCache** ()  
*reimplement these in subclasses*
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- **ShowfotoItemViewDelegate** ([ShowfotoItemViewDelegatePrivate](#) &dd, QWidget \*const parent)

**Protected Member Functions inherited from [Digikam::DItemDelegate](#)**

- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

**Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)**

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)  
*Declare as slot in the derived class calling this method.*

## Additional Inherited Members

### Signals inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

### Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

### Static Public Member Functions inherited from [ShowFoto::ShowfotoDelegate](#)

- static QPixmap **retrieveThumbnailPixmap** (const QModelIndex &index, int thumbnailSize)

### Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

### Protected Slots inherited from [ShowFoto::ShowfotoDelegate](#)

- void **modelChanged** ()
- void **modelContentsChanged** ()

### Protected Slots inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

### Protected Attributes inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- [ShowfotoItemViewDelegatePrivate](#) \*const **d\_ptr** = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > **m\_overlays**

## 6.1600.1 Member Function Documentation

### 6.1600.1.1 updateRects()

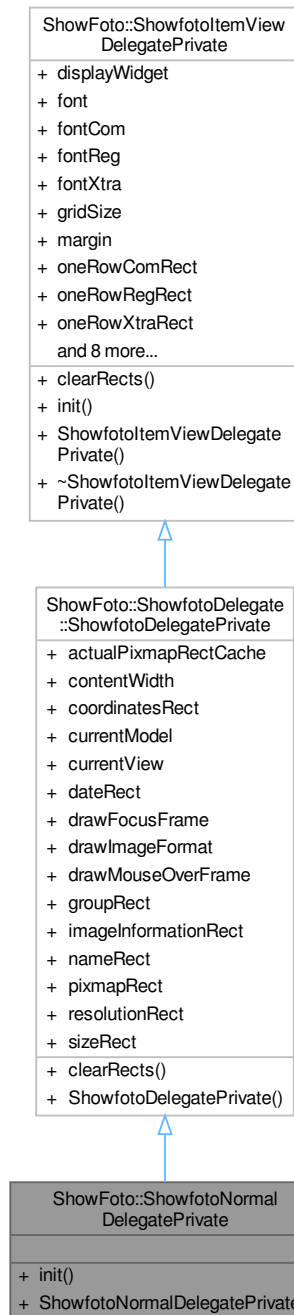
```
void ShowFoto::ShowfotoNormalDelegate::updateRects ( ) [override], [protected], [virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The paint() method operates depending on these rects.

Implements [ShowFoto::ShowfotoDelegate](#).

## 6.1601 ShowFoto::ShowfotoNormalDelegatePrivate Class Reference

Inheritance diagram for ShowFoto::ShowfotoNormalDelegatePrivate:



### Public Member Functions

- void **init** ([ShowfotoNormalDelegate](#) \*const q, [ShowfotoThumbnailBar](#) \*const parent)

## Public Member Functions inherited from ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate

- void `clearRects` () override  
*Resets cached rects. Remember to reimplement in subclass for added rects.*

## Public Member Functions inherited from ShowFoto::ShowfotoItemViewDelegatePrivate

- void `init` (ShowfotoItemViewDelegate \*const \_q, QWidget \*const \_widget)

## Additional Inherited Members

## Public Attributes inherited from ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate

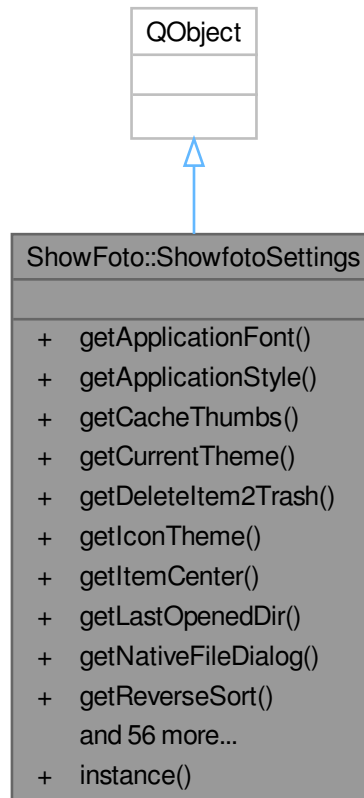
- QCache< int, QRect > `actualPixmapRectCache`
- int `contentWidth` = 0
- QRect `coordinatesRect`
- QAbstractItemModel \* `currentModel` = nullptr
- ShowfotoThumbnailBar \* `currentView` = nullptr
- QRect `dateRect`
- bool `drawFocusFrame` = true
- bool `drawImageFormat` = true
- bool `drawMouseOverFrame` = true
- QRect `groupRect`
- QRect `imageInformationRect`
- QRect `nameRect`
- QRect `pixmapRect`
- QRect `resolutionRect`
- QRect `sizeRect`

## Public Attributes inherited from ShowFoto::ShowfotoItemViewDelegatePrivate

- QWidget \* `displayWidget` = nullptr
- QFont `font`
- QFont `fontCom`
- QFont `fontReg`
- QFont `fontXtra`
- QSize `gridSize`
- int `margin` = 5
- QRect `oneRowComRect`
- QRect `oneRowRegRect`
- QRect `oneRowXtraRect`
- ShowfotoItemViewDelegate \* `q` = nullptr
- int `radius` = 3  
*constant values for drawing*
- QVector< QPixmap > `ratingPixmaps`
- QRect `rect`
- QPixmap `regPixmap`
- QPixmap `selPixmap`
- int `spacing` = 0
- ThumbnailSize `thumbSize` = ThumbnailSize(0)

## 6.1602 ShowFoto::ShowfotoSettings Class Reference

Inheritance diagram for ShowFoto::ShowfotoSettings:



### Public Member Functions

- QFont **getApplicationFont** () const
- QString **getApplicationStyle** () const
- bool **getCacheThumbs** () const
- QString **getCurrentTheme** () const
- bool **getDeleteltem2Trash** () const
- QString **getIconTheme** () const
- bool **getItemCenter** () const
- QString **getLastOpenedDir** () const
- bool **getNativeFileDialog** () const
- bool **getReverseSort** () const
- int **getRightSideBarStyle** () const
- bool **getShowCoordinates** () const
- bool **getShowFileDate** () const
- bool **getShowFileDim** () const
- bool **getShowFileName** () const
- bool **getShowFileSize** () const



- bool **getShowFileType** () const
- bool **getShowFormatOverThumbnail** () const
- bool **getShowPhotoDate** () const
- bool **getShowPhotoExpo** () const
- bool **getShowPhotoFlash** () const
- bool **getShowPhotoFocal** () const
- bool **getShowPhotoLens** () const
- bool **getShowPhotoMake** () const
- bool **getShowPhotoMode** () const
- bool **getShowPhotoWB** () const
- bool **getShowSplash** () const
- bool **getShowToolTip** () const
- int **getSortRole** () const
- QFont **getToolTipFont** () const
- int **getUpdateType** () const
- bool **getUpdateWithDebug** () const
- void **readSettings** ()
- void **setApplicationFont** (const QFont &fnt)
- void **setApplicationStyle** (const QString &style)
- void **setCacheThumbs** (bool item)
- void **setCurrentTheme** (const QString &theme)
- void **setDeleteItem2Trash** (bool D2t)
- void **setIconTheme** (const QString &theme)
- void **setItemCenter** (bool item)
- void **setLastOpenedDir** (const QString &dir)
- void **setNativeFileDialog** (bool item)
- void **setReverseSort** (bool reverse)
- void **setRightSideBarStyle** (int style)
- void **setShowCoordinates** (bool show)
- void **setShowFileDate** (bool show)
- void **setShowFileDim** (bool show)
- void **setShowFileName** (bool show)
- void **setShowFileSize** (bool show)
- void **setShowFileType** (bool show)
- void **setShowFormatOverThumbnail** (bool show)
- void **setShowPhotoDate** (bool show)
- void **setShowPhotoExpo** (bool show)
- void **setShowPhotoFlash** (bool show)
- void **setShowPhotoFocal** (bool show)
- void **setShowPhotoLens** (bool show)
- void **setShowPhotoMake** (bool show)
- void **setShowPhotoMode** (bool show)
- void **setShowPhotoWB** (bool show)
- void **setShowSplash** (bool show)
- void **setShowToolTip** (bool show)
- void **setSortRole** (int order)
- void **setToolTipFont** (const QFont &font)
- void **setUpdateType** (int type)
- void **setUpdateWithDebug** (bool dbg)
- void **syncConfig** ()

### Static Public Member Functions

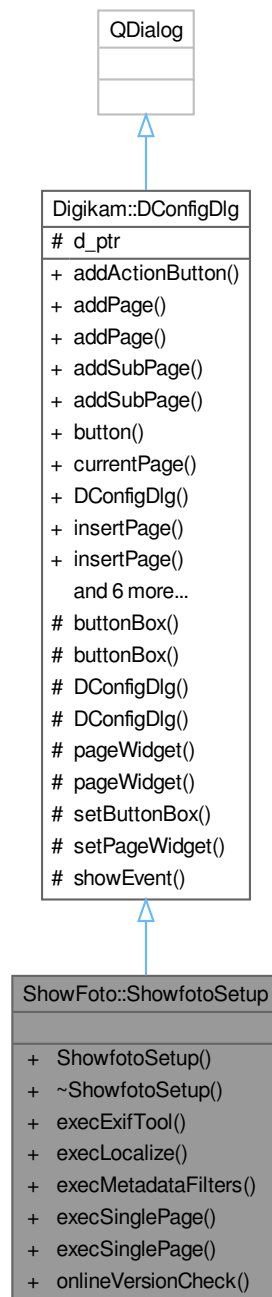
- static [ShowfotoSettings](#) \* **instance** ()

## Friends

- class **ShowfotoSettingsCreator**

## 6.1603 ShowFoto::ShowfotoSetup Class Reference

Inheritance diagram for ShowFoto::ShowfotoSetup:



## Public Types

- enum **Page** {  
**LastPageUsed** = -1 , **EditorPage** = 0 , **MetadataPage** , **ToolTipPage** ,  
**RawPage** , **IOFilesPage** , **ICCPAGE** , **PluginsPage** ,  
**MiscellaneousPage** , **SetupPageEnumLast** }

## Public Types inherited from [Digikam::DConfigDlg](#)

- enum [FaceType](#) {  
**Auto** = DConfigDlgView::Auto , **Plain** = DConfigDlgView::Plain , **List** = DConfigDlgView::List , **Tree** =  
DConfigDlgView::Tree ,  
**Tabbed** = DConfigDlgView::Tabbed }

## Public Member Functions

- **ShowfotoSetup** (QWidget \*const parent=nullptr, Page page=LastPageUsed)

## Public Member Functions inherited from [Digikam::DConfigDlg](#)

- void [addActionButton](#) (QAbstractButton \*const button)
- void [addPage](#) (DConfigDlgWdgItem \*const item)
- DConfigDlgWdgItem \* [addPage](#) (QWidget \*const widget, const QString &name)
- void [addSubPage](#) (DConfigDlgWdgItem \*const parent, DConfigDlgWdgItem \*const item)
- DConfigDlgWdgItem \* [addSubPage](#) (DConfigDlgWdgItem \*const parent, QWidget \*const widget, const QString &name)
- QPushButton \* [button](#) (QDialogButtonBox::StandardButton which) const
- DConfigDlgWdgItem \* [currentPage](#) () const
- DConfigDlg (QWidget \*const parent=nullptr, Qt::WindowFlags flags=Qt::WindowFlags())
- void [insertPage](#) (DConfigDlgWdgItem \*const before, DConfigDlgWdgItem \*const item)
- DConfigDlgWdgItem \* [insertPage](#) (DConfigDlgWdgItem \*const before, QWidget \*const widget, const QString &name)
- void [removePage](#) (DConfigDlgWdgItem \*const item)
- void [setConfigGroup](#) (const QString &group)
- void [setCurrentPage](#) (DConfigDlgWdgItem \*const item)
- void [setFaceType](#) (FaceType faceType)
- void [setStandardButtons](#) (QDialogButtonBox::StandardButtons buttons)
- [~DConfigDlg](#) () override

## Static Public Member Functions

- static bool **execExifTool** (QWidget \*const parent)
- static bool **execLocalize** (QWidget \*const parent)
- static bool **execMetadataFilters** (QWidget \*const parent, int tab)
- static bool [execSinglePage](#) (Page page)
- static bool **execSinglePage** (QWidget \*const parent, Page page)
- static void **onlineVersionCheck** ()

## Additional Inherited Members

### Signals inherited from [Digikam::DConfigDlg](#)

- void [currentPageChanged](#) ([DConfigDlgWdgItem](#) \*current, [DConfigDlgWdgItem](#) \*before)
- void [pageRemoved](#) ([DConfigDlgWdgItem](#) \*page)

### Protected Member Functions inherited from [Digikam::DConfigDlg](#)

- QDialogButtonBox \* [buttonBox](#) ()
- const QDialogButtonBox \* [buttonBox](#) () const
- **DConfigDlg** ([DConfigDlgPrivate](#) &dd, [DConfigDlgWdg](#) \*const widget, QWidget \*const parent, Qt::Window↔Flags flags=Qt::WindowFlags())
- [DConfigDlg](#) ([DConfigDlgWdg](#) \*const widget, QWidget \*const parent, Qt::WindowFlags flags=Qt::Window↔Flags())
- [DConfigDlgWdg](#) \* [pageWidget](#) ()
- const [DConfigDlgWdg](#) \* [pageWidget](#) () const
- void [setButtonBox](#) (QDialogButtonBox \*const box)
- void [setPageWidget](#) ([DConfigDlgWdg](#) \*const widget)
- void **showEvent** (QShowEvent \*) override

### Protected Attributes inherited from [Digikam::DConfigDlg](#)

- [DConfigDlgPrivate](#) \*const **d\_ptr** = nullptr

## 6.1603.1 Member Function Documentation

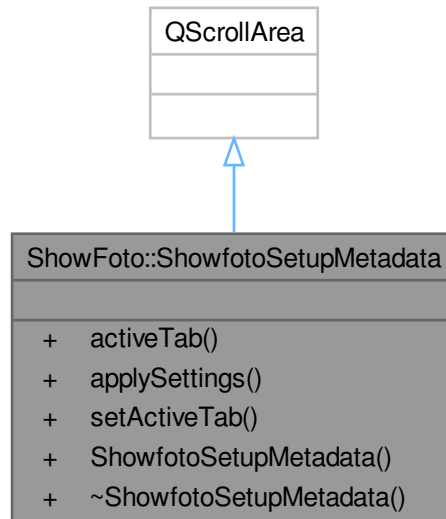
### 6.1603.1.1 [execSinglePage\(\)](#)

```
bool ShowFoto::ShowfotoSetup::execSinglePage (
    Page page ) [static]
```

Show a setup dialog. Only the specified page will be available.

## 6.1604 ShowFoto::ShowfotoSetupMetadata Class Reference

Inheritance diagram for ShowFoto::ShowfotoSetupMetadata:



### Public Types

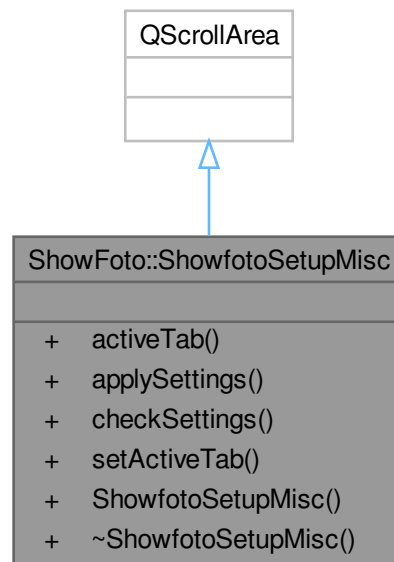
- enum **MetadataTab** {  
    **Behavior** = 0 , **ExifViewer** , **MakernotesViewer** , **IptcViewer** ,  
    **XmpViewer** , **ExifTool** }

### Public Member Functions

- MetadataTab **activeTab** () const
- void **applySettings** ()
- void **setActiveTab** (MetadataTab tab)
- **ShowfotoSetupMetadata** (QWidget \*const parent=nullptr)

## 6.1605 ShowFoto::ShowfotoSetupMisc Class Reference

Inheritance diagram for ShowFoto::ShowfotoSetupMisc:



### Public Types

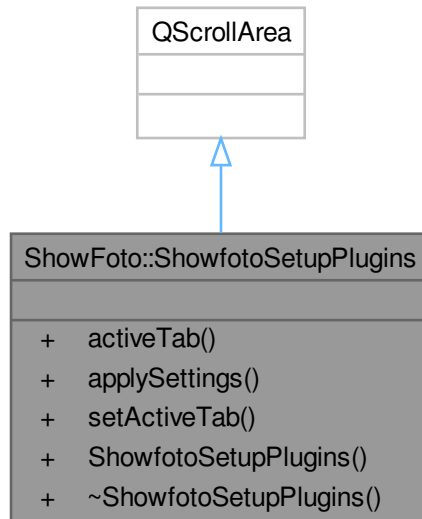
- enum `MiscTab` { `Behaviour = 0` , `Appearance` , `SpellCheck` , `Localize` , `System` }
- enum `SortOrder` { `SortByDate = 0` , `SortByName` , `SortByFileSize` }

### Public Member Functions

- `MiscTab activeTab ()` const
- void `applySettings ()`
- bool `checkSettings ()`
- void `setActiveTab (MiscTab tab)`
- `ShowfotoSetupMisc (QWidget *const parent=nullptr)`

## 6.1606 ShowFoto::ShowfotoSetupPlugins Class Reference

Inheritance diagram for ShowFoto::ShowfotoSetupPlugins:



### Public Types

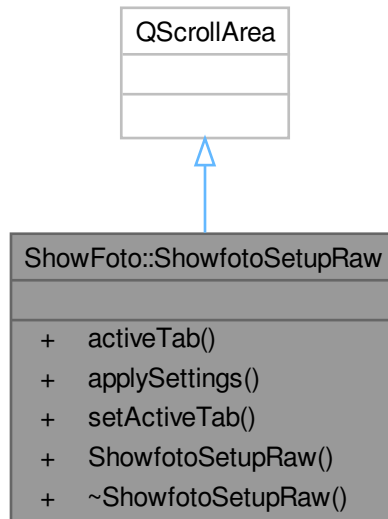
- enum `PluginTab` { `Generic = 0` , `Editor` , `Loaders` }

### Public Member Functions

- `PluginTab` **activeTab** () const
- void **applySettings** ()
- void **setActiveTab** (PluginTab tab)
- **ShowfotoSetupPlugins** (QWidget \*const parent=nullptr)

## 6.1607 ShowFoto::ShowfotoSetupRaw Class Reference

Inheritance diagram for ShowFoto::ShowfotoSetupRaw:



### Public Types

- enum `RAWTab` { `RAWBehavior = 0` , `RAWDefaultSettings` }

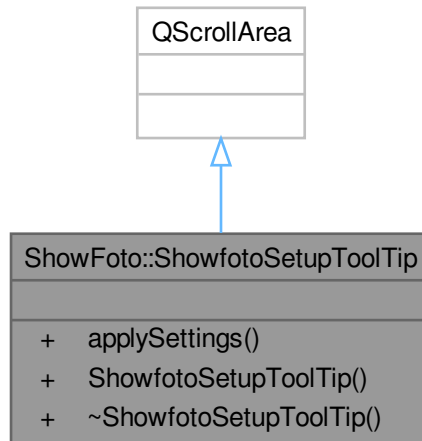
### Public Member Functions

- `RAWTab` **activeTab** () const
- void **applySettings** ()
- void **setActiveTab** (RAWTab tab)
- **ShowfotoSetupRaw** (QWidget \*const parent=nullptr)



## 6.1608 ShowFoto::ShowfotoSetupToolTip Class Reference

Inheritance diagram for ShowFoto::ShowfotoSetupToolTip:

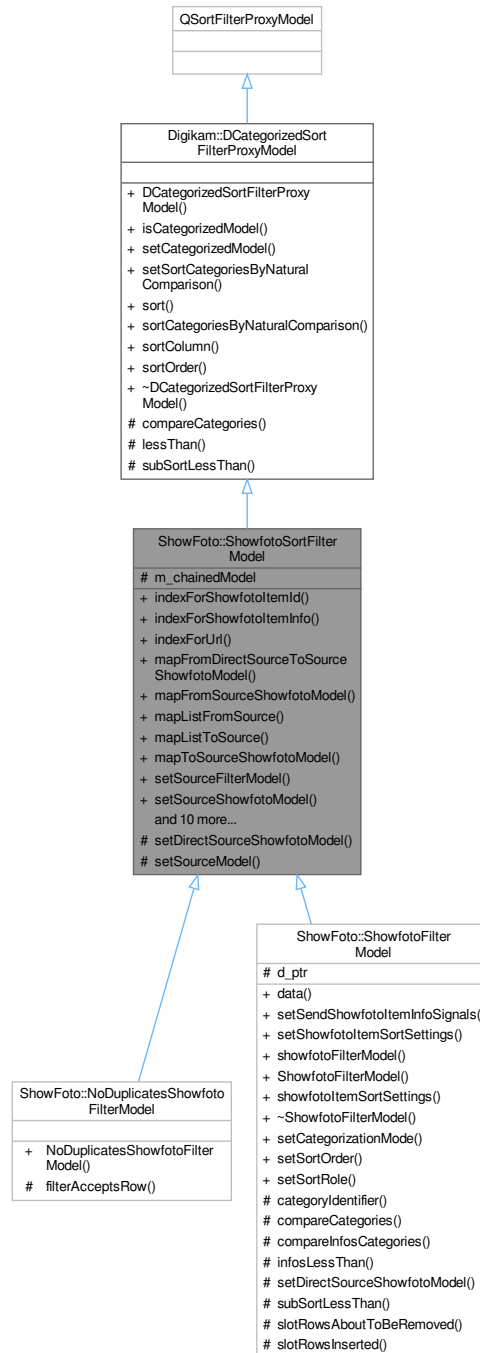


### Public Member Functions

- void **applySettings** ()
- **ShowfotoSetupToolTip** (QWidget \*const parent=nullptr)

## 6.1609 ShowFoto::ShowfotoSortFilterModel Class Reference

Inheritance diagram for ShowFoto::ShowfotoSortFilterModel:



### Public Member Functions

- QModelIndex **indexForShowfotoItemId** (qulonglong id) const
- QModelIndex **indexForShowfotoItemInfo** (const [ShowfotoItemInfo](#) &info) const

- QModelIndex **indexForUrl** (const QUrl &fileUrl) const
- QModelIndex **mapFromDirectSourceToSourceShowfotoModel** (const QModelIndex &sourceModelIndex) const
- QModelIndex **mapFromSourceShowfotoModel** (const QModelIndex &showfotoModelIndex) const
- QList< QModelIndex > **mapListFromSource** (const QList< QModelIndex > &sourceIndexes) const
- QList< QModelIndex > **mapListToSource** (const QList< QModelIndex > &indexes) const
- QModelIndex **mapToSourceShowfotoModel** (const QModelIndex &proxyIndex) const
- void **setSourceFilterModel** (ShowfotoSortFilterModel \*const sourceModel)
- void **setSourceShowfotoModel** (ShowfotoItemModel \*const sourceModel)
- virtual ShowfotoFilterModel \* **showfotoFilterModel** () const  
*Returns this, any chained ShowfotoFilterModel, or 0.*
- qlonglong **showfotoItemId** (const QModelIndex &index) const
- QList< qlonglong > **showfotoItemIds** (const QList< QModelIndex > &indexes) const
- ShowfotoItemInfo **showfotoItemInfo** (const QModelIndex &index) const
- QList< ShowfotoItemInfo > **showfotoItemInfos** (const QList< QModelIndex > &indexes) const
- QList< ShowfotoItemInfo > **showfotoItemInfosSorted** () const
- ShowfotoSortFilterModel (QObject \*const parent=nullptr)
- ShowfotoSortFilterModel \* **sourceFilterModel** () const
- ShowfotoItemModel \* **sourceShowfotoModel** () const

### Public Member Functions inherited from Digikam::DCategorizedSortFilterProxyModel

- DCategorizedSortFilterProxyModel (QObject \*const parent=nullptr)
- bool **isCategorizedModel** () const
- void **setCategorizedModel** (bool categorizedModel)
- void **setSortCategoriesByNaturalComparison** (bool sortCategoriesByNaturalComparison)
- void **sort** (int column, Qt::SortOrder order=Qt::AscendingOrder) override
- bool **sortCategoriesByNaturalComparison** () const
- int **sortColumn** () const
- Qt::SortOrder **sortOrder** () const

### Protected Member Functions

- virtual void **setDirectSourceShowfotoModel** (ShowfotoItemModel \*const sourceModel)  
*Reimplement if needed. Called only when model shall be set as (direct) sourceModel.*
- void **setSourceModel** (QAbstractItemModel \*sourceModel) override

### Protected Member Functions inherited from Digikam::DCategorizedSortFilterProxyModel

- virtual int **compareCategories** (const QModelIndex &left, const QModelIndex &right) const
- bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const override
- virtual bool **subSortLessThan** (const QModelIndex &left, const QModelIndex &right) const

### Protected Attributes

- ShowfotoSortFilterModel \* **m\_chainedModel** = nullptr

## Additional Inherited Members

### Public Types inherited from [Digikam::DCategorizedSortFilterProxyModel](#)

- enum [AdditionalRoles](#) { [CategoryDisplayRole](#) = 0x17CE990A , [CategorySortRole](#) = 0x27857E60 }

## 6.1609.1 Member Function Documentation

### 6.1609.1.1 [mapToSourceShowfotoModel\(\)](#)

```
QModelIndex ShowFoto::ShowfotoSortFilterModel::mapToSourceShowfotoModel (
    const QModelIndex & proxyIndex ) const
```

Convenience methods mapped to [ShowfotoItemModel](#). Mentioned indexes returned come from the source [Showfoto](#) image model.

### 6.1609.1.2 [setDirectSourceShowfotoModel\(\)](#)

```
void ShowFoto::ShowfotoSortFilterModel::setDirectSourceShowfotoModel (
    ShowfotoItemModel *const sourceModel ) [protected], [virtual]
```

Reimplemented in [ShowFoto::ShowfotoFilterModel](#).

### 6.1609.1.3 [showfotoFilterModel\(\)](#)

```
ShowfotoFilterModel * ShowFoto::ShowfotoSortFilterModel::showfotoFilterModel ( ) const [virtual]
```

Reimplemented in [ShowFoto::ShowfotoFilterModel](#).

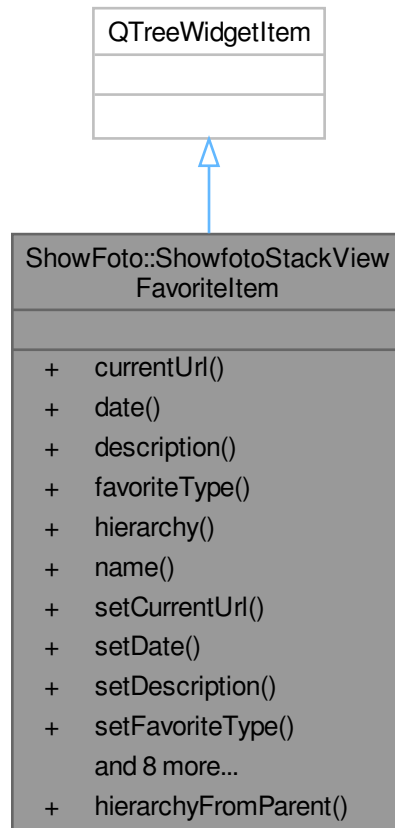
### 6.1609.1.4 [showfotoItemInfosSorted\(\)](#)

```
QList< ShowfotoItemInfo > ShowFoto::ShowfotoSortFilterModel::showfotoItemInfosSorted ( ) const
```

Returns a list of all showfoto infos, sorted according to this model. If you do not need a sorted list, use [ShowfotoItemModel](#)'s [showfotoItemInfo\(\)](#) method.

## 6.1610 ShowFoto::ShowfotoStackViewFavoriteItem Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewFavoriteItem:



### Public Types

- enum `FavoriteType` { `FavoriteRoot` = -1 , `FavoriteFolder` , `FavoriteItem` }

### Public Member Functions

- `QUrl` **currentUrl** () const
- `QDate` **date** () const
- `QString` **description** () const
- `int` **favoriteType** () const
- `QString` **hierarchy** () const
- `QString` **name** () const
- `void` **setCurrentUrl** (const `QUrl` &url)
- `void` **setDate** (const `QDate` &date)
- `void` **setDescription** (const `QString` &desc)
- `void` **setFavoriteType** (int favoriteType)

- void **setHierarchy** (const QString &desc)
- void **setName** (const QString &name)
- void **setUrls** (const QList< QUrl > &)
- **ShowfotoStackViewFavoriteItem** (QTreeWidgetItem \*const parent)
- **ShowfotoStackViewFavoriteItem** (QTreeWidgetItem \*const parent, int favType)
- QList< QUrl > **urls** () const
- QStringList **urlsToPaths** () const

### Static Public Member Functions

- static QString **hierarchyFromParent** (const QString &name, [ShowfotoStackViewFavoriteItem](#) \*const pitem)

## 6.1610.1 Member Enumeration Documentation

### 6.1610.1.1 FavoriteType

```
enum ShowFoto::ShowfotoStackViewFavoriteItem::FavoriteType
```

#### Enumerator

FavoriteRoot	Favorite is root item from hierarchy.
FavoriteFolder	Favorite is a simple folder in hierarchy.
FavoriteItem	Favorite is a hierarchy item including all properties.

## 6.1610.2 Member Function Documentation

### 6.1610.2.1 hierarchyFromParent()

```
QString ShowFoto::ShowfotoStackViewFavoriteItem::hierarchyFromParent (
    const QString & name,
    ShowfotoStackViewFavoriteItem *const pitem ) [static]
```

Helper static method to get hierarchy path from item. 'name' is the title and 'pitem" the parent instance.

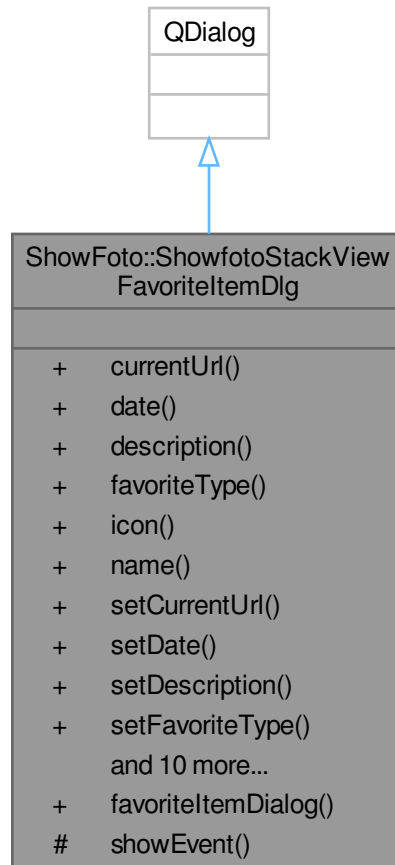
### 6.1610.2.2 urlsToPaths()

```
QStringList ShowFoto::ShowfotoStackViewFavoriteItem::urlsToPaths ( ) const
```

Helper method to get a list local paths from image urls included in favorite item.

## 6.1611 ShowFoto::ShowfotoStackViewFavoriteItemDlg Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewFavoriteItemDlg:



### Public Member Functions

- `QUrl` **currentUrl** () const
- `QDate` **date** () const
- `QString` **description** () const
- `int` **favoriteType** () const
- `QString` **icon** () const
- `QString` **name** () const
- `void` **setCurrentUrl** (const `QUrl` &url)
- `void` **setDate** (const `QDate` &name)
- `void` **setDescription** (const `QString` &desc)
- `void` **setFavoriteType** (int favoriteType)
- `void` **setIcon** (const `QString` &icon)
- `void` **setIconSize** (int size)
- `void` **setName** (const `QString` &name)
- `void` **setParentItem** ([ShowfotoStackViewFavoriteItem](#) \*const pItem)

- void **setSortOrder** (int order)
- void **setSortRole** (int role)
- void **setUrls** (const QList< QUrl > &urls)
- **ShowfotoStackViewFavoriteItemDlg** ([ShowfotoStackViewFavoriteList](#) \*const list, bool create=false)
- QList< QUrl > **urls** () const

### Static Public Member Functions

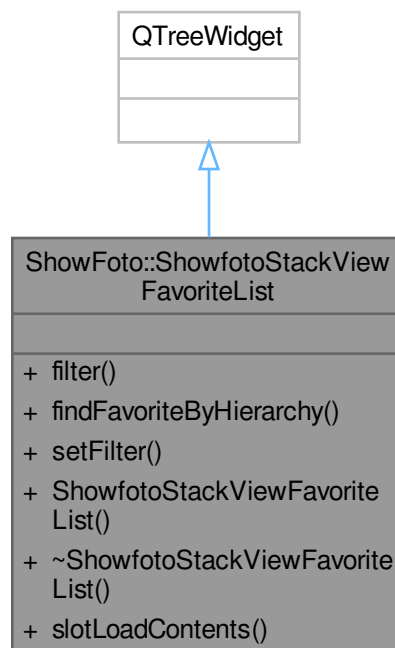
- static bool **favoriteItemDialog** ([ShowfotoStackViewFavoriteList](#) \*const list, QString &name, int &favoriteType, QString &desc, QDate &date, QString &icon, QList< QUrl > &urls, QUrl &current, int iconSize, int sortOrder, int sortRole, [ShowfotoStackViewFavoriteItem](#) \*const pItem, bool create=false)

### Protected Member Functions

- void **showEvent** (QShowEvent \*) override

## 6.1612 ShowFoto::ShowfotoStackViewFavoriteList Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewFavoriteList:



### Public Slots

- void **slotLoadContents** ()



## Signals

- void **signalAddFavorite** ()
- void **signalAddFavorite** (const QList< QUrl > &, const QUrl &current)
- void **signalLoadContentsFromFiles** (const QStringList &files, const QString &current)
- void **signalSearchResult** (int)

## Public Member Functions

- QString **filter** () const
- **ShowfotoStackViewFavoriteItem** \* **findFavoriteByHierarchy** (const QString &hierarchy)
- void **setFilter** (const QString &filter, Qt::CaseSensitivity cs)
- **ShowfotoStackViewFavoriteList** (**ShowfotoStackViewFavorites** \*const parent)

## 6.1612.1 Member Function Documentation

### 6.1612.1.1 filter()

```
QString ShowFoto::ShowfotoStackViewFavoriteList::filter ( ) const
```

Return the current string used to filter the favorites list.

### 6.1612.1.2 setFilter()

```
void ShowFoto::ShowfotoStackViewFavoriteList::setFilter (
    const QString & filter,
    Qt::CaseSensitivity cs )
```

Set the string used to filter the favorites list. [signalSearchResult\(\)](#) is emitted when all is done.

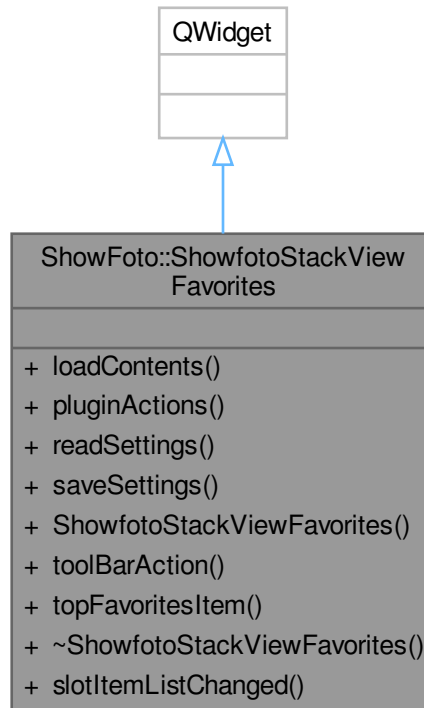
### 6.1612.1.3 signalSearchResult

```
void ShowFoto::ShowfotoStackViewFavoriteList::signalSearchResult (
    int ) [signal]
```

Signal emitted when filtering is done through `slotSetFilter()`. Number of favorites found is sent when item relevant of filtering match the query.

## 6.1613 ShowFoto::ShowfotoStackViewFavorites Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewFavorites:



### Public Slots

- void **slotItemListChanged** (int nbitems)

### Signals

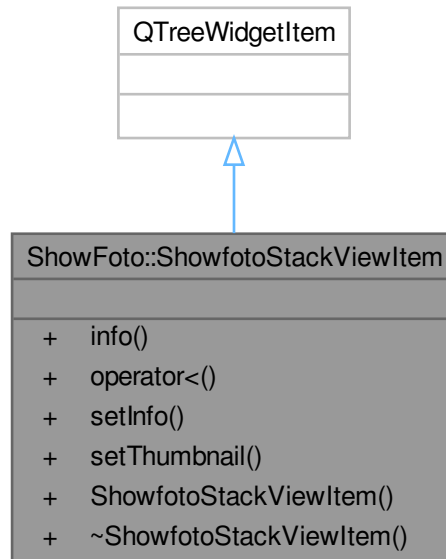
- void **signalLoadContents** ()
- void **signalLoadContentsFromFiles** (const QStringList &files, const QString &current)

### Public Member Functions

- void **loadContents** ()
- QList< QAction \* > **pluginActions** () const
- bool **readSettings** ()
- bool **saveSettings** ()
- **ShowfotoStackViewFavorites** ([ShowfotoStackViewSideBar](#) \*const sidebar)
- QAction \* **toolBarAction** (const QString &name) const
- QTreeWidgetItem \* **topFavoritesItem** () const

## 6.1614 ShowFoto::ShowfotoStackViewItem Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewItem:

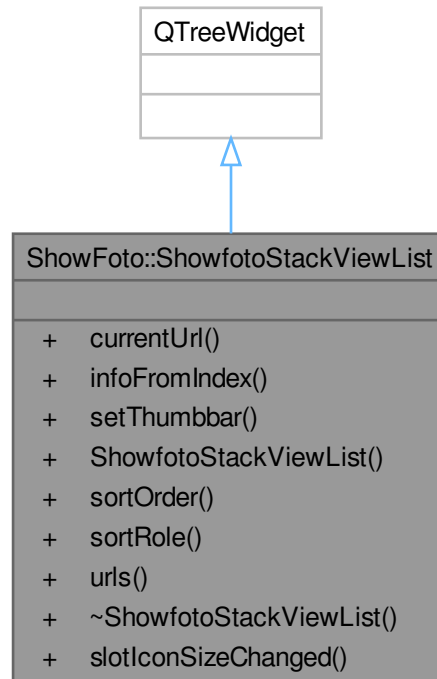


### Public Member Functions

- [ShowfotoItemInfo](#) **info** () const
- bool **operator**< (const [QTreeWidgetItem](#) &other) const override
- void **setInfo** (const [ShowfotoItemInfo](#) &)
- void **setThumbnail** (const [QPixmap](#) &)
- **ShowfotoStackViewItem** ([ShowfotoStackViewList](#) \*const parent)

## 6.1615 ShowFoto::ShowfotoStackViewList Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewList:



### Public Types

- enum `StackViewRole` { `FileName` = 0 , `FileSize` , `FileType` , `FileDate` }
- enum `ThumbnailSize` { `SizeSmall` = 32 , `SizeMedium` = 48 , `SizeLarge` = 64 , `SizeHuge` = 96 }

### Public Slots

- void `slotIconSizeChanged` (int)

### Signals

- void `signalAddFavorite` ()
- void `signalClearItemsList` ()
- void `signalItemListItemChanged` (int nbitems)
- void `signalRemoveItemInfos` (const QList< `ShowfotoItemInfo` > &infos)
- void `signalShowfotoItemInfoActivated` (const `ShowfotoItemInfo` &info)

## Public Member Functions

- `QUrl` **currentUrl** () const
- `ShowfotoItemInfo` **infoFromIndex** (const `QModelIndex` &index) const
- void **setThumbbar** (`ShowfotoThumbnailBar` \*const thumbbar)
- `ShowfotoStackViewList` (`ShowfotoStackViewSideBar` \*const view)
- int **sortOrder** () const
- int **sortRole** () const
- `QList< QUrl >` **urls** ()

## 6.1615.1 Member Enumeration Documentation

### 6.1615.1.1 StackViewRole

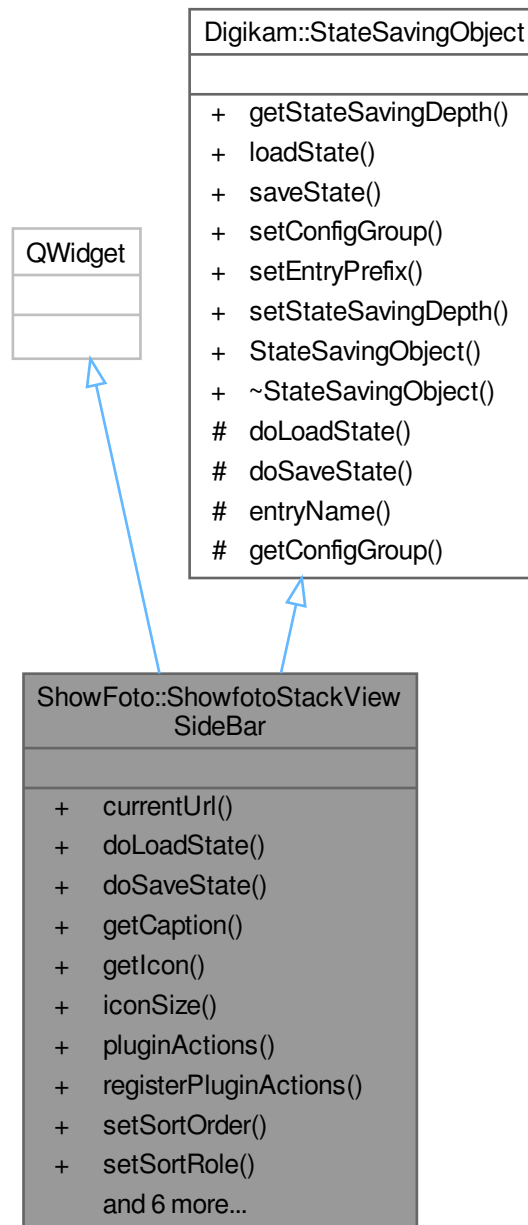
enum `ShowFoto::ShowfotoStackViewList::StackViewRole`

#### Enumerator

<code>FileDate</code>	Metadata date if exists, else Modifier date.
-----------------------	--

## 6.1616 ShowFoto::ShowfotoStackViewSideBar Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewSideBar:



### Signals

- void **signalAddFavorite** ()
- void **signalClearItemsList** ()
- void **signalLoadContentsFromFiles** (const QStringList &files, const QString &current)
- void **signalRemoveItemInfos** (const QList< [ShowfotoItemInfo](#) > &infos)
- void **signalShowfotoItemInfoActivated** (const [ShowfotoItemInfo](#) &info)

**Public Member Functions**

- `QUrl` **currentUrl** () const
- void **doLoadState** () override
- void **doSaveState** () override
- const `QString` **getCaption** ()
- const `QIcon` **getIcon** ()
- int **iconSize** () const
- `QList< QAction * >` **pluginActions** () const
- void **registerPluginActions** (const `QList< DPluginAction * >` &actions)
- void **setSortOrder** (int order)
- void **setSortRole** (int role)
- void **setThumbbar** (`ShowfotoThumbbar *const` thumbbar)
- `ShowfotoStackViewSideBar` (`Showfoto *const` parent)
- int **sortOrder** () const
- int **sortRole** () const
- `QList< QUrl >` **urls** () const

**Public Member Functions inherited from [Digikam::StateSavingObject](#)**

- `StateSavingDepth` **getStateSavingDepth** () const
- void **loadState** ()
- void **saveState** ()
- virtual void **setConfigGroup** (const `KConfigGroup` &group)
- virtual void **setEntryPrefix** (const `QString` &prefix)
- void **setStateSavingDepth** (const `StateSavingDepth` depth)
- `StateSavingObject` (`QObject *const` host)
- virtual `~StateSavingObject` ()

**Additional Inherited Members****Public Types inherited from [Digikam::StateSavingObject](#)**

- enum `StateSavingDepth` { `INSTANCE` , `DIRECT_CHILDREN` , `RECURSIVE` }

**Protected Member Functions inherited from [Digikam::StateSavingObject](#)**

- `QString` **entryName** (const `QString` &base) const
- `KConfigGroup` **getConfigGroup** () const

**6.1616.1 Member Function Documentation****6.1616.1.1 doLoadState()**

```
void ShowFoto::ShowfotoStackViewSideBar::doLoadState ( ) [override], [virtual]
```

Implement this hook method for state loading. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

### 6.1616.1.2 doSaveState()

```
void ShowFoto::ShowfotoStackViewSideBar::doSaveState ( ) [override], [virtual]
```

Implement this hook method for state saving. Use [getConfigGroup\(\)](#) and [entryName\(\)](#) for the implementation.

Implements [Digikam::StateSavingObject](#).

## 6.1617 ShowFoto::ShowfotoStackViewToolTip Class Reference

Inheritance diagram for ShowFoto::ShowfotoStackViewToolTip:





### Public Member Functions

- void **setIndex** (const QModelIndex &index)
- **ShowfotoStackViewToolTip** ([ShowfotoStackViewList](#) \*const view)

### Public Member Functions inherited from [Digikam::DItemToolTip](#)

- **DItemToolTip** (QWidget \*const parent=nullptr)

### Additional Inherited Members

### Protected Member Functions inherited from [Digikam::DItemToolTip](#)

- bool **event** (QEvent \*) override
- void **paintEvent** (QPaintEvent \*) override
- void **renderArrows** ()
- void **reposition** ()
- void **resizeEvent** (QResizeEvent \*) override
- bool **toolTipsEmpty** () const
- void **updateToolTip** ()



## Public Slots inherited from [ShowFoto::ShowfotoCategorizedView](#)

- void [hintAt](#) (const [ShowfotoItemInfo](#) &info)
- void [setCurrentInfo](#) (const [ShowfotoItemInfo](#) &info)
- void [setCurrentUrl](#) (const [QUrl](#) &url)
- void [setCurrentWhenAvailable](#) (qulonglong [ShowfotoItemId](#))
- void [setSelectedShowfotoItemInfos](#) (const [QList](#)< [ShowfotoItemInfo](#) > &infos)
- void [setSelectedUrls](#) (const [QList](#)< [QUrl](#) > &urlList)
- void [setThumbnailSize](#) (int size)

## Public Slots inherited from [Digikam::ItemViewCategorized](#)

- void [copy](#) () override
- void [cut](#) () override
- void [hideIndexNotification](#) ()
- void [paste](#) () override
- void [showIndexNotification](#) (const [QModelIndex](#) &index, const [QString](#) &message)

## Public Slots inherited from [Digikam::DCategorizedView](#)

- void [reset](#) () override

## Public Member Functions

- [ShowfotoItemInfo](#) [findItemByUrl](#) (const [QUrl](#) &url)
- [QModelIndex](#) [firstIndex](#) () const
- void [installOverlays](#) ()
- [QModelIndex](#) [lastIndex](#) () const
- [QModelIndex](#) [nextIndex](#) (const [QModelIndex](#) &index) const
- [QModelIndex](#) [previousIndex](#) (const [QModelIndex](#) &index) const
- void [setFlow](#) ([QListView::Flow](#) newFlow)
- void [setModelsFiltered](#) ([ShowfotoItemModel](#) \*model, [ShowfotoSortFilterModel](#) \*filterModel)
- void [setScrollBarPolicy](#) ([Qt::ScrollBarPolicy](#) policy)
- [ShowfotoThumbnailBar](#) ([QWidget](#) \*const parent=nullptr)
- int [thumbnailIndexForUrl](#) (const [QUrl](#) &url) const

## Public Member Functions inherited from [ShowFoto::ShowfotoCategorizedView](#)

- void [addOverlay](#) ([ItemDelegateOverlay](#) \*overlay, [ShowfotoDelegate](#) \*delegate=nullptr)
- [ShowfotoItemInfo](#) [currentInfo](#) () const
- [QUrl](#) [currentUrl](#) () const
- [ShowfotoDelegate](#) \* [delegate](#) () const
- [QItemSelectionModel](#) \* [getSelectionModel](#) () const
- [ShowfotoItemInfo](#) [nextInfo](#) (const [ShowfotoItemInfo](#) &info)
- [ShowfotoItemInfo](#) [nextInOrder](#) (const [ShowfotoItemInfo](#) &startingPoint, int nth)
- [ShowfotoItemInfo](#) [previousInfo](#) (const [ShowfotoItemInfo](#) &info)
- void [removeOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- [QList](#)< [ShowfotoItemInfo](#) > [selectedShowfotoItemInfos](#) () const
- [QList](#)< [ShowfotoItemInfo](#) > [selectedShowfotoItemInfosCurrentFirst](#) () const
- [QList](#)< [QUrl](#) > [selectedUrls](#) () const
- void [setModels](#) ([ShowfotoItemModel](#) \*model, [ShowfotoSortFilterModel](#) \*filterModel)

- virtual void **setThumbnailSize** (const [ThumbnailSize](#) &size)
- **ShowfotoCategorizedView** (QWidget \*const parent=nullptr)
- [ShowfotoFilterModel](#) \* **showfotoFilterModel** () const
- QList< [ShowfotoItemInfo](#) > **showfotoItemInfos** () const
- [ShowfotoItemModel](#) \* **showfotoItemModel** () const
- [ShowfotoSortFilterModel](#) \* **showfotoSortFilterModel** () const
- [ShowfotoThumbnailModel](#) \* **showfotoThumbnailModel** () const
- [ThumbnailSize](#) **thumbnailSize** () const
- void **toIndex** (const QUrl &url)
- QList< QUrl > **urls** () const

## Public Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **awayFromSelection** ()
- [DItemDelegate](#) \* **delegate** () const
- void **invertSelection** ()
- bool **isToolTipEnabled** () const
- **ItemViewCategorized** (QWidget \*const parent=nullptr)
- int **numberOfSelectedIndexes** () const
- void **scrollTo** (const QModelIndex &index, ScrollHint hint=EnsureVisible) override
- void **scrollToRelaxed** (const QModelIndex &index, ScrollHint hint=EnsureVisible)
- void **setInitialSelectedItem** (bool enabled)
- void **setScrollCurrentToCenter** (bool enabled)
- void **setScrollStepGranularity** (int factor)
- void **setSelectedIndexes** (const QList< QModelIndex > &indexes)
- void **setSpacing** (int spacing)
- void **setToolTipEnabled** (bool enabled)
- void **setUsePointingHandCursor** (bool useCursor)
- void **toFirstIndex** ()
- void **toIndex** (const QModelIndex &index)
- void **toLastIndex** ()
- void **toNextIndex** ()
- void **toPreviousIndex** ()

## Public Member Functions inherited from [Digikam::DCategorizedView](#)

- virtual QModelIndexList **categorizedIndexesIn** (const QRect &rect) const
- virtual QModelIndex **categoryAt** (const QPoint &point) const
- [DCategoryDrawer](#) \* **categoryDrawer** () const
- virtual QItemSelectionRange **categoryRange** (const QModelIndex &index) const
- virtual QRect **categoryVisualRect** (const QModelIndex &index) const
- **DCategorizedView** (QWidget \*const parent=nullptr)
- QModelIndex **indexAt** (const QPoint &point) const override
- void **setCategoryDrawer** ([DCategoryDrawer](#) \*categoryDrawer)
- void **setDrawDraggedItems** (bool drawDraggedItems)
- void **setGridSize** (const QSize &size)
- void **setModel** (QAbstractItemModel \*model) override
- QRect **visualRect** (const QModelIndex &index) const override

## Public Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual void **copy** ()
- virtual void **cut** ()
- virtual void **paste** ()

## Protected Member Functions

- bool **event** (QEvent \*) override

## Protected Member Functions inherited from [ShowFoto::ShowfotoCategorizedView](#)

- virtual void **activated** (const [ShowfotoItemInfo](#) &info, Qt::KeyboardModifiers modifiers)  
*Reimplement these in a subclass.*
- void **currentChanged** (const QModelIndex &index, const QModelIndex &previous) override
- [AbstractItemDragDropHandler](#) \* **dragDropHandler** () const override
- QSortFilterProxyModel \* **filterModel** () const override  
*reimplemented from parent class*
- void **indexActivated** (const QModelIndex &index, Qt::KeyboardModifiers modifiers) override
- QModelIndex **nextIndexHint** (const QModelIndex &indexToAnchor, const QItemSelectionRange &removed) const override
- void **paintEvent** (QPaintEvent \*e) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([ShowfotoDelegate](#) \*delegate)
- void **showContextMenuOnIndex** (QContextMenuEvent \*event, const QModelIndex &index) override  
*Reimplement these in a subclass.*
- virtual void **showContextMenuOnInfo** (QContextMenuEvent \*event, const [ShowfotoItemInfo](#) &info)
- void **updateGeometries** () override

## Protected Member Functions inherited from [Digikam::ItemViewCategorized](#)

- void **contextMenuEvent** (QContextMenuEvent \*event) override  
*reimplemented from parent class*
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- QModelIndex **indexForCategoryAt** (const QPoint &pos) const
- void **keyPressEvent** (QKeyEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- QModelIndex **mapIndexForDragDrop** (const QModelIndex &index) const override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- QPixmap **pixmapForDrag** (const QList< QModelIndex > &indexes) const override
- void **reset** () override
- void **resizeEvent** (QResizeEvent \*e) override
- void **rowsAboutToBeRemoved** (const QModelIndex &parent, int start, int end) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- void **rowsRemoved** (const QModelIndex &parent, int start, int end) override
- void **selectionChanged** (const QItemSelection &, const QItemSelection &) override
- void **setItemDelegate** ([DItemDelegate](#) \*delegate)

- void **setToolTip** (ItemViewToolTip \*tip)
- virtual void **showContextMenu** (QContextMenuEvent \*event)
- virtual bool **showToolTip** (const QModelIndex &index, QStyleOptionViewItem &option, QHelpEvent \*e=nullptr)
- void **updateDelegateSizes** ()
- void **userInteraction** ()
- bool **viewportEvent** (QEvent \*event) override
- void **wheelEvent** (QWheelEvent \*event) override

### Protected Member Functions inherited from [Digikam::DCategorizedView](#)

- void **dragLeaveEvent** (QDragLeaveEvent \*event) override
- void **dragMoveEvent** (QDragMoveEvent \*event) override
- void **dropEvent** (QDropEvent \*event) override
- void **leaveEvent** (QEvent \*event) override
- void **mouseMoveEvent** (QMouseEvent \*event) override
- void **mousePressEvent** (QMouseEvent \*event) override
- void **mouseReleaseEvent** (QMouseEvent \*event) override
- QModelIndex **moveCursor** (CursorAction cursorAction, Qt::KeyboardModifiers modifiers) override
- void **paintEvent** (QPaintEvent \*event) override
- void **resizeEvent** (QResizeEvent \*event) override
- void **setSelection** (const QRect &rect, QItemSelectionModel::SelectionFlags flags) override
- void **startDrag** (Qt::DropActions supportedActions) override

### Protected Member Functions inherited from [Digikam::DragDropViewImplementation](#)

- virtual QAbstractItemView \* **asView** ()=0
- bool **decodelsCutSelection** (const QMimeData \*mimeData)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*e)
- void **dropEvent** (QDropEvent \*e)
- void **encodelsCutSelection** (QMimeData \*mime, bool isCutSelection)
- void **startDrag** (Qt::DropActions supportedActions)

### Additional Inherited Members

### Signals inherited from [ShowFoto::ShowfotoCategorizedView](#)

- void **currentChanged** (const ShowfotoItemInfo &info)
- void **deselected** (const QList< ShowfotoItemInfo > &nowDeselectedInfos)
- void **modelChanged** ()
- void **selected** (const QList< ShowfotoItemInfo > &newSelectedInfos)
- void **showfotoItemInfoActivated** (const ShowfotoItemInfo &info)

### Signals inherited from [Digikam::ItemViewCategorized](#)

- void **clicked** (const QMouseEvent \*e, const QModelIndex &index)
- void **entered** (const QMouseEvent \*e, const QModelIndex &index)
- void **keyPressed** (QKeyEvent \*e)
- void **selectionChanged** ()
- void **selectionCleared** ()
- void **viewportClicked** (const QMouseEvent \*e)
- void **zoomInStep** ()
- void **zoomOutStep** ()

## Protected Slots inherited from [Digikam::ItemViewCategorized](#)

- void **layoutAboutToBeChanged** ()
- void **layoutWasChanged** ()
- void **slotActivated** (const QModelIndex &index)
- void **slotClicked** (const QModelIndex &index)
- void **slotEntered** (const QModelIndex &index)
- virtual void **slotSetupChanged** ()
- virtual void **slotThemeChanged** ()

## Protected Slots inherited from [Digikam::DCategorizedView](#)

- void **currentChanged** (const QModelIndex &current, const QModelIndex &previous) override
- void **rowsInserted** (const QModelIndex &parent, int start, int end) override
- virtual void **rowsInsertedArtificial** (const QModelIndex &parent, int start, int end)
- virtual void **slotLayoutChanged** ()
- void **updateGeometries** () override

## 6.1618.1 Member Function Documentation

### 6.1618.1.1 setModelsFiltered()

```
void ShowFoto::ShowfotoThumbnailBar::setModelsFiltered (
    ShowfotoItemModel * model,
    ShowfotoSortFilterModel * filterModel )
```

This installs a duplicate filter model, if the ShwofotoItemModel may contain duplicates. Otherwise, just use `setModels()`.

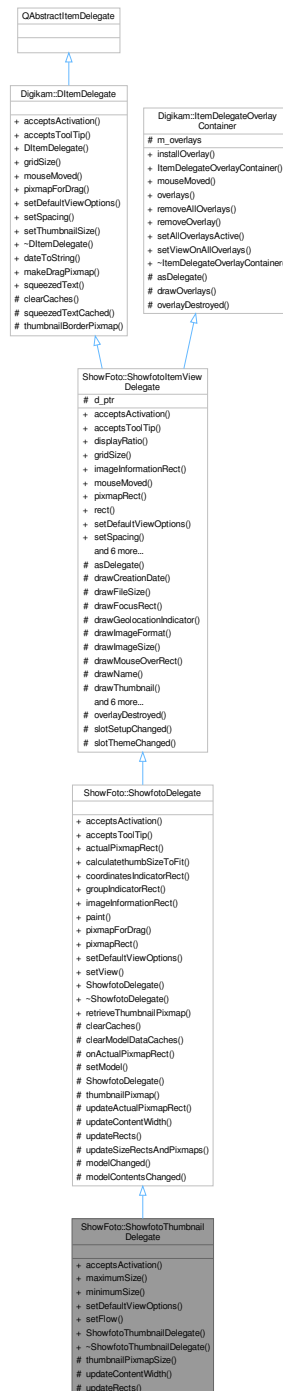
### 6.1618.1.2 setScrollBarPolicy()

```
void ShowFoto::ShowfotoThumbnailBar::setScrollBarPolicy (
    Qt::ScrollBarPolicy policy )
```

Sets the policy always for the one scroll bar which is relevant, depending on orientation.

## 6.1619 ShowFoto::ShowfotoThumbnailDelegate Class Reference

Inheritance diagram for ShowFoto::ShowfotoThumbnailDelegate:



### Public Member Functions

- bool `acceptsActivation` (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*activationRect) const override



- int [maximumSize](#) () const
- int [minimumSize](#) () const
- void [setDefaultViewOptions](#) (const QStyleOptionViewItem &option) override
- void [setFlow](#) (QListView::Flow flow)
- **ShowfotoThumbnailDelegate** ([ShowfotoThumbnailBar](#) \*const parent)

### Public Member Functions inherited from [ShowFoto::ShowfotoDelegate](#)

- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- QRect [actualPixmapRect](#) (const QModelIndex &index) const
- int [calculatethumbSizeToFit](#) (int ws)
- QRect [coordinatesIndicatorRect](#) () const
- QRect [groupIndicatorRect](#) () const
- QRect [imageInformationRect](#) () const override
- void [paint](#) (QPainter \*painter, const QStyleOptionViewItem &option, const QModelIndex &index) const override
- QPixmap [pixmapForDrag](#) (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes) const override
- QRect [pixmapRect](#) () const override
- void [setView](#) ([ShowfotoThumbnailBar](#) \*view)
- **ShowfotoDelegate** (QWidget \*const parent)

### Public Member Functions inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- bool [acceptsToolTip](#) (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*tooltipRect=nullptr) const override
- double [displayRatio](#) () const
- QSize [gridSize](#) () const override
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index) override
- QRect [rect](#) () const
- void [setSpacing](#) (int spacing) override
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize) override  
*reimplemented from DItemDelegate*
- **ShowfotoItemViewDelegate** (QWidget \*const parent)
- QSize [sizeHint](#) (const QStyleOptionViewItem &option, const QModelIndex &index) const override
- int [spacing](#) () const
- [ThumbnailSize](#) [thumbnailSize](#) () const

### Public Member Functions inherited from [Digikam::DItemDelegate](#)

- **DItemDelegate** (QObject \*const parent=nullptr)

### Public Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- void [installOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- [ItemDelegateOverlayContainer](#) ()=default
- void [mouseMoved](#) (QMouseEvent \*e, const QRect &visualRect, const QModelIndex &index)
- QList< [ItemDelegateOverlay](#) \* > [overlays](#) () const
- void [removeAllOverlays](#) ()
- void [removeOverlay](#) ([ItemDelegateOverlay](#) \*overlay)
- void [setAllOverlaysActive](#) (bool active)
- void [setViewOnAllOverlays](#) (QAbstractItemView \*view)

### Protected Member Functions

- int **thumbnailPixmapSize** (bool withHighlight, int size)
- void **updateContentWidth** () override
- void **updateRects** () override

### Protected Member Functions inherited from [ShowFoto::ShowfotoDelegate](#)

- void **clearCaches** () override
- virtual void **clearModelDataCaches** ()
- bool **onActualPixmapRect** (const QPoint &pos, const QRect &visualRect, const QModelIndex &index, QRect \*actualRect) const
- void **setModel** (QAbstractItemModel \*model)
- **ShowfotoDelegate** ([ShowfotoDelegate::ShowfotoDelegatePrivate](#) &dd, QWidget \*const parent)
- virtual QPixmap **thumbnailPixmap** (const QModelIndex &index) const
- void **updateActualPixmapRect** (const QModelIndex &index, const QRect &rect)
- void **updateSizeRectsAndPixmaps** () override

### Protected Member Functions inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- QAbstractItemDelegate \* **asDelegate** () override  
*Returns the delegate, typically, the derived class.*
- void **drawCreationDate** (QPainter \*p, const QRect &dateRect, const QDateTime &date) const
- void **drawFileSize** (QPainter \*p, const QRect &r, qlonglong bytes) const
- void **drawFocusRect** (QPainter \*p, const QStyleOptionViewItem &option, bool isSelected) const
- void **drawGeolocationIndicator** (QPainter \*p, const QRect &r) const
- void **drawImageFormat** (QPainter \*p, const QRect &dimsRect, const QString &mime) const
- void **drawImageSize** (QPainter \*p, const QRect &dimsRect, const QSize &dims) const
- void **drawMouseOverRect** (QPainter \*p, const QStyleOptionViewItem &option) const
- void **drawName** (QPainter \*p, const QRect &nameRect, const QString &name) const
- QRect **drawThumbnail** (QPainter \*p, const QRect &thumbRect, const QPixmap &background, const QPixmap &thumbnail) const  
*Use the tool methods for painting in subclasses.*
- virtual void **invalidatePaintingCache** ()  
*reimplement these in subclasses*
- void **prepareBackground** ()
- void **prepareFonts** ()
- void **prepareMetrics** (int maxWidth)
- **ShowfotoItemViewDelegate** ([ShowfotoItemViewDelegatePrivate](#) &dd, QWidget \*const parent)

### Protected Member Functions inherited from [Digikam::DItemDelegate](#)

- QString **squeezedTextCached** (QPainter \*const p, int width, const QString &text) const
- QPixmap **thumbnailBorderPixmap** (const QSize &pixSize, bool isGrouped=false) const

### Protected Member Functions inherited from [Digikam::ItemDelegateOverlayContainer](#)

- virtual void **drawOverlays** (QPainter \*p, const QStyleOptionViewItem &option, const QModelIndex &index) const
- virtual void **overlayDestroyed** (QObject \*o)  
*Declare as slot in the derived class calling this method.*

## Additional Inherited Members

### Signals inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- void **hideNotification** ()
- void **requestNotification** (const QModelIndex &index, const QString &message)

### Signals inherited from [Digikam::DItemDelegate](#)

- void **gridSizeChanged** (const QSize &newSize)
- void **visualChange** ()

### Static Public Member Functions inherited from [ShowFoto::ShowfotoDelegate](#)

- static QPixmap **retrieveThumbnailPixmap** (const QModelIndex &index, int thumbnailSize)

### Static Public Member Functions inherited from [Digikam::DItemDelegate](#)

- static QString **dateToString** (const QDateTime &datetime)
- static QPixmap **makeDragPixmap** (const QStyleOptionViewItem &option, const QList< QModelIndex > &indexes, double displayRatio, const QPixmap &suggestedPixmap=QPixmap())
- static QString **squeezedText** (const QFontMetrics &fm, int width, const QString &text)

### Protected Slots inherited from [ShowFoto::ShowfotoDelegate](#)

- void **modelChanged** ()
- void **modelContentsChanged** ()

### Protected Slots inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- void **overlayDestroyed** (QObject \*o) override
- void **slotSetupChanged** ()
- void **slotThemeChanged** ()

### Protected Attributes inherited from [ShowFoto::ShowfotoItemViewDelegate](#)

- [ShowfotoItemViewDelegatePrivate](#) \*const **d\_ptr** = nullptr

### Protected Attributes inherited from [Digikam::ItemDelegateOverlayContainer](#)

- QList< [ItemDelegateOverlay](#) \* > **m\_overlays**

## 6.1619.1 Member Function Documentation

### 6.1619.1.1 acceptsActivation()

```
bool ShowFoto::ShowfotoThumbnailDelegate::acceptsActivation (
    const QPoint & pos,
    const QRect & visualRect,
    const QModelIndex & index,
    QRect * activationRect ) const [override], [virtual]
```

Reimplemented from [ShowFoto::ShowfotoDelegate](#).

### 6.1619.1.2 maximumSize()

```
int ShowFoto::ShowfotoThumbnailDelegate::maximumSize ( ) const
```

Returns the minimum or maximum viewport size in the limiting dimension, width or height, depending on current flow.

### 6.1619.1.3 setDefaultViewOptions()

```
void ShowFoto::ShowfotoThumbnailDelegate::setDefaultViewOptions (
    const QStyleOptionViewItem & option ) [override], [virtual]
```

Style option with standard values to use for cached rendering. option.rect shall be the viewport rectangle. Call on resize, font change.

Reimplemented from [ShowFoto::ShowfotoDelegate](#).

### 6.1619.1.4 updateContentWidth()

```
void ShowFoto::ShowfotoThumbnailDelegate::updateContentWidth ( ) [override], [protected],
[virtual]
```

Reimplement this to set contentWidth. This is the maximum width of all content rectangles, typically excluding margins on both sides.

Reimplemented from [ShowFoto::ShowfotoDelegate](#).

### 6.1619.1.5 updateRects()

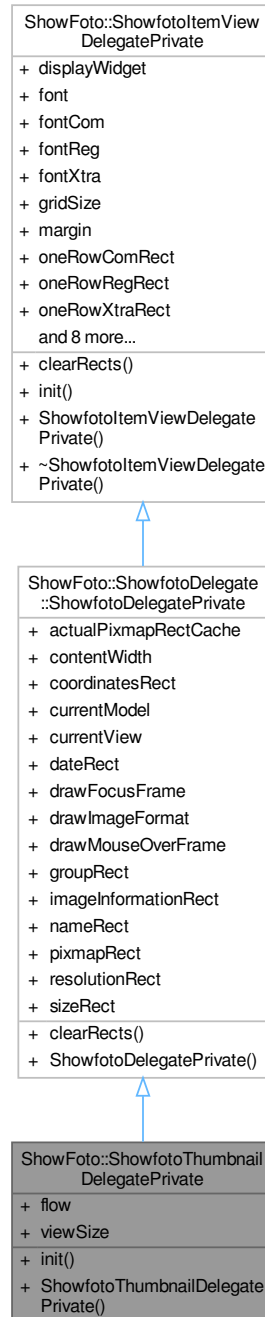
```
void ShowFoto::ShowfotoThumbnailDelegate::updateRects ( ) [override], [protected], [virtual]
```

In a subclass, you need to implement this method to set up the rects for drawing. The paint() method operates depending on these rects.

Implements [ShowFoto::ShowfotoDelegate](#).

## 6.1620 ShowFoto::ShowfotoThumbnailDelegatePrivate Class Reference

Inheritance diagram for ShowFoto::ShowfotoThumbnailDelegatePrivate:



### Public Member Functions

- void `init` ([ShowfotoThumbnailDelegate](#) \*const q)

## Public Member Functions inherited from [ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate](#)

- void [clearRects](#) () override  
*Resets cached rects. Remember to reimplement in subclass for added rects.*

## Public Member Functions inherited from [ShowFoto::ShowfotoItemViewDelegatePrivate](#)

- void [init](#) ([ShowfotoItemViewDelegate](#) \*const \_q, QWidget \*const \_widget)

## Public Attributes

- QListView::Flow [flow](#) = QListView::LeftToRight
- QRect [viewSize](#)

## Public Attributes inherited from [ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate](#)

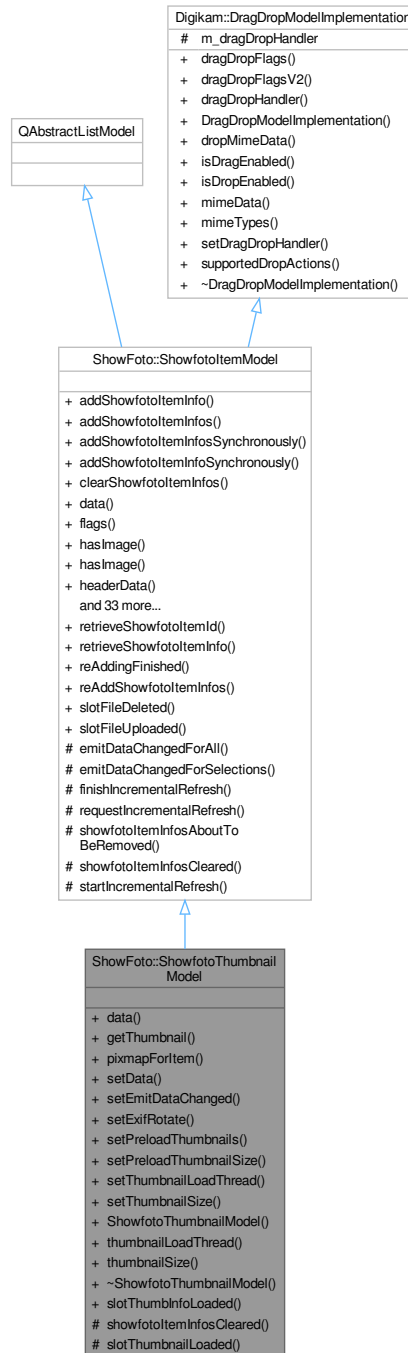
- QCache< int, QRect > [actualPixmapRectCache](#)
- int [contentWidth](#) = 0
- QRect [coordinatesRect](#)
- QAbstractItemModel \* [currentModel](#) = nullptr
- [ShowfotoThumbnailBar](#) \* [currentView](#) = nullptr
- QRect [dateRect](#)
- bool [drawFocusFrame](#) = true
- bool [drawImageFormat](#) = true
- bool [drawMouseOverFrame](#) = true
- QRect [groupRect](#)
- QRect [imageInformationRect](#)
- QRect [nameRect](#)
- QRect [pixmapRect](#)
- QRect [resolutionRect](#)
- QRect [sizeRect](#)

## Public Attributes inherited from [ShowFoto::ShowfotoItemViewDelegatePrivate](#)

- QWidget \* [displayWidget](#) = nullptr
- QFont [font](#)
- QFont [fontCom](#)
- QFont [fontReg](#)
- QFont [fontXtra](#)
- QSize [gridSize](#)
- int [margin](#) = 5
- QRect [oneRowComRect](#)
- QRect [oneRowRegRect](#)
- QRect [oneRowXtraRect](#)
- [ShowfotoItemViewDelegate](#) \* [q](#) = nullptr
- int [radius](#) = 3  
*constant values for drawing*
- QVector< QPixmap > [ratingPixmaps](#)
- QRect [rect](#)
- QPixmap [regPixmap](#)
- QPixmap [selPixmap](#)
- int [spacing](#) = 0
- [ThumbnailSize](#) [thumbSize](#) = [ThumbnailSize](#)(0)

## 6.1621 ShowFoto::ShowfotoThumbnailModel Class Reference

Inheritance diagram for ShowFoto::ShowfotoThumbnailModel:



### Public Slots

- void **slotThumbInfoLoaded** (const [ShowfotoItemInfo](#) &info, const QImage &thumbnailImage)

## Public Slots inherited from [ShowFoto::ShowfotoItemModel](#)

- void **reAddingFinished** ()
- void **reAddShowfotoItemInfos** (const ShowfotoItemInfoList &infos)
- void **slotFileDeleted** (const QString &folder, const QString &file, bool status)
- void **slotFileUploaded** (const [ShowfotoItemInfo](#) &info)

## Signals

- void **signalItemThumbnail** (const [ShowfotoItemInfo](#) &info, const QPixmap &pix)
- void **signalThumbInfo** (const [ShowfotoItemInfo](#) &info, const QImage &thumbnailImage) const
- void **thumbnailAvailable** (const QModelIndex &index, int requestedSize)
- void **thumbnailFailed** (const QModelIndex &index, int requestedSize)

## Signals inherited from [ShowFoto::ShowfotoItemModel](#)

- void [allRefreshingFinished](#) ()
- void [itemInfosAboutToBeAdded](#) (const QList< [ShowfotoItemInfo](#) > &infos)
- void [itemInfosAboutToBeRemoved](#) (const QList< [ShowfotoItemInfo](#) > &infos)
- void [itemInfosAdded](#) (const QList< [ShowfotoItemInfo](#) > &infos)
- void [itemInfosRemoved](#) (const QList< [ShowfotoItemInfo](#) > &infos)
- void [preprocess](#) (const QList< [ShowfotoItemInfo](#) > &infos)
- void **processAdded** (const QList< [ShowfotoItemInfo](#) > &infos)
- void [readyForIncrementalRefresh](#) ()

## Public Member Functions

- QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const override
- bool **getThumbnail** (const [ShowfotoItemInfo](#) &itemInfo, QImage &thumbnail) const
- bool **pixmapForItem** (const QString &url, QPixmap &pix) const
- bool [setData](#) (const QModelIndex &index, const QVariant &value, int role=Qt::DisplayRole) override
- void [setEmitDataChanged](#) (bool emitSignal)
- void **setExifRotate** (bool rotate)
- void [setPreloadThumbnails](#) (bool preload)
- void [setPreloadThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize)
- void [setThumbnailLoadThread](#) ([ThumbnailLoadThread](#) \*thread)
- void [setThumbnailSize](#) (const [ThumbnailSize](#) &thumbSize)
- [ShowfotoThumbnailModel](#) (QWidget \*const parent)
- [ThumbnailLoadThread](#) \* **thumbnailLoadThread** () const
- [ThumbnailSize](#) **thumbnailSize** () const



## Public Member Functions inherited from ShowFoto::ShowfotoItemModel

- void **addShowfotoItemInfo** (const ShowfotoItemInfo &info)
- void **addShowfotoItemInfos** (const QList< ShowfotoItemInfo > &infos)
- void **addShowfotoItemInfosSynchronously** (const QList< ShowfotoItemInfo > &infos)
- void **addShowfotoItemInfoSynchronously** (const ShowfotoItemInfo &info)
- void **clearShowfotoItemInfos** ()
- QVariant **data** (const QModelIndex &index, int role) const override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **hasImage** (const ShowfotoItemInfo &info) const
- bool **hasImage** (qulonglong id) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role) const override
- QModelIndex **index** (int row, int column, const QModelIndex &parent) const override
- QList< QModelIndex > **indexesForShowfotoItemId** (qulonglong id) const
- QList< QModelIndex > **indexesForShowfotoItemInfo** (const ShowfotoItemInfo &info) const
- QList< QModelIndex > **indexesForUrl** (const QUrl &fileUrl) const
- QModelIndex **indexForShowfotoItemId** (qulonglong id) const
- QModelIndex **indexForShowfotoItemInfo** (const ShowfotoItemInfo &info) const
- QModelIndex **indexForUrl** (const QUrl &fileUrl) const
- bool **isEmpty** () const
- int **numberOfIndexesForShowfotoItemId** (qulonglong id) const
- int **numberOfIndexesForShowfotoItemInfo** (const ShowfotoItemInfo &info) const
- void **removeIndex** (const QModelIndex &index)
- void **removeIndexes** (const QList< QModelIndex > &indexes)
- void **removeShowfotoItemInfo** (const ShowfotoItemInfo &info)
- void **removeShowfotoItemInfos** (const QList< ShowfotoItemInfo > &infos)
- int **rowCount** (const QModelIndex &parent) const override
- void **setKeepsFileUrlCache** (bool keepCache)
- DECLARE\_MODEL\_DRAG\_DROP\_METHODS void **setSendRemovalSignals** (bool send)
- void **setShowfotoItemInfos** (const QList< ShowfotoItemInfo > &infos)
- qulonglong **showfotoItemId** (const QModelIndex &index) const
- qulonglong **showfotoItemId** (int row) const
- QList< qulonglong > **showfotoItemIds** () const
- QList< qulonglong > **showfotoItemIds** (const QList< QModelIndex > &indexes) const
- ShowfotoItemInfo **showfotoItemInfo** (const QModelIndex &index) const
- ShowfotoItemInfo **showfotoItemInfo** (const QUrl &fileUrl) const
- ShowfotoItemInfo **showfotoItemInfo** (int row) const
- ShowfotoItemInfo & **showfotoItemInfoRef** (const QModelIndex &index) const
- ShowfotoItemInfo & **showfotoItemInfoRef** (int row) const
- QList< ShowfotoItemInfo > **showfotoItemInfos** () const
- ShowfotoItemInfoList **showfotoItemInfos** (const QList< QModelIndex > &indexes) const
- QList< ShowfotoItemInfo > **showfotoItemInfos** (const QUrl &fileUrl) const
- ShowfotoItemModel (QObject \*const parent)
- QList< ShowfotoItemInfo > **uniqueShowfotoItemInfos** () const

## Public Member Functions inherited from Digikam::DragDropModelImplementation

- virtual Qt::ItemFlags **dragDropFlags** (const QModelIndex &index) const
- Qt::ItemFlags **dragDropFlagsV2** (const QModelIndex &index) const
- AbstractItemDragDropHandler \* **dragDropHandler** () const
- DragDropModelImplementation ()=default
- bool **dropMimeData** (const QMimeData \*, Qt::DropAction, int, int, const QModelIndex &)
- virtual bool **isDragEnabled** (const QModelIndex &index) const
- virtual bool **isDropEnabled** (const QModelIndex &index) const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const
- QStringList **mimeTypes** () const
- void **setDragDropHandler** (AbstractItemDragDropHandler \*handler)
- Qt::DropActions **supportedDropActions** () const

### Protected Slots

- void **slotThumbnailLoaded** (const [LoadingDescription](#) &loadingDescription, const QPixmap &thumb)

### Protected Member Functions

- void [showfotoItemInfosCleared](#) () override

### Protected Member Functions inherited from [ShowFoto::ShowfotoItemModel](#)

- void **emitDataChangedForAll** ()
- void **emitDataChangedForSelections** (const QItemSelection &selection)
- void **finishIncrementalRefresh** ()
- void [requestIncrementalRefresh](#) ()
- virtual void [showfotoItemInfosAboutToBeRemoved](#) (int, int)
- void [startIncrementalRefresh](#) ()

### Additional Inherited Members

### Public Types inherited from [ShowFoto::ShowfotoItemModel](#)

- enum [ShowfotoItemModelRoles](#) {  
[ShowfotoItemModelPointerRole](#) = Qt::UserRole , **ShowfotoItemModelInternalId** = Qt::UserRole + 1 ,  
[ThumbnailRole](#) = Qt::UserRole + 2 , [ExtraDataRole](#) = Qt::UserRole + 3 ,  
[ExtraDataDuplicateCount](#) = Qt::UserRole + 6 , **FilterModelRoles** = Qt::UserRole + 100 }

### Static Public Member Functions inherited from [ShowFoto::ShowfotoItemModel](#)

- static qlonglong **retrieveShowfotoItemId** (const QModelIndex &index)
- static [ShowfotoItemInfo](#) **retrieveShowfotoItemInfo** (const QModelIndex &index)

### Protected Attributes inherited from [Digikam::DragDropModelImplementation](#)

- [AbstractItemDragDropHandler](#) \* **m\_dragDropHandler** = nullptr

## 6.1621.1 Constructor & Destructor Documentation

### 6.1621.1.1 ShowfotoThumbnailModel()

```
ShowFoto::ShowfotoThumbnailModel::ShowfotoThumbnailModel (
    QWidget *const parent ) [explicit]
```

An ItemModel that supports thumbnail loading. You need to set a ThumbnailLoadThread to enable thumbnail loading. Adjust the thumbnail size to your needs. Note that setKeepsFilePatindexesForPathhCache is enabled per default.

## 6.1621.2 Member Function Documentation

### 6.1621.2.1 data()

```
QVariant ShowFoto::ShowfotoThumbnailModel::data (
    const QModelIndex & index,
    int role = Qt::DisplayRole ) const [override]
```

Handles the ThumbnailRole. If the pixmap is available, returns it in the QVariant. If it still needs to be loaded, returns a null QVariant and emits thumbnailAvailable() as soon as it is available.

### 6.1621.2.2 setData()

```
bool ShowFoto::ShowfotoThumbnailModel::setData (
    const QModelIndex & index,
    const QVariant & value,
    int role = Qt::DisplayRole ) [override]
```

You can override the current thumbnail size by giving an integer value for ThumbnailRole. Set a null QVariant to use the thumbnail size set by [setThumbnailSize\(\)](#) again. The index given here is ignored for this purpose.

### 6.1621.2.3 setEmitDataChanged()

```
void ShowFoto::ShowfotoThumbnailModel::setEmitDataChanged (
    bool emitSignal )
```

Enable emitting dataChanged() when a thumbnail becomes available. The thumbnailAvailable() signal will be emitted in any case. Default is true.

### 6.1621.2.4 setPreloadThumbnails()

```
void ShowFoto::ShowfotoThumbnailModel::setPreloadThumbnails (
    bool preload )
```

Enable preloading of thumbnails: If preloading is enabled, for every entry in the model a thumbnail generation is started. Default: false.

### 6.1621.2.5 setPreloadThumbnailSize()

```
void ShowFoto::ShowfotoThumbnailModel::setPreloadThumbnailSize (
    const ThumbnailSize & thumbSize )
```

If you want to fix a size for preloading, do it here.

### 6.1621.2.6 setThumbnailLoadThread()

```
void ShowFoto::ShowfotoThumbnailModel::setThumbnailLoadThread (
    ThumbnailLoadThread * thread )
```

Enable thumbnail loading and set the thread that shall be used. The thumbnail size of this thread will be adjusted.

### 6.1621.2.7 `setThumbnailSize()`

```
void ShowFoto::ShowfotoThumbnailModel::setThumbnailSize (
    const ThumbnailSize & thumbSize )
```

Set the thumbnail size to use.

### 6.1621.2.8 `showfotoItemInfosCleared()`

```
void ShowFoto::ShowfotoThumbnailModel::showfotoItemInfosCleared ( ) [override], [protected],
[virtual]
```

Called when the internal storage is cleared.

Reimplemented from [ShowFoto::ShowfotoItemModel](#).

# Index

- [\\_k\\_dataChanged](#)
  - [Digikam::DConfigDlgViewPrivate, 943](#)
- [\\_k\\_modelChanged](#)
  - [Digikam::DConfigDlgViewPrivate, 943](#)
- [~ActionJob](#)
  - [Digikam::ActionJob, 211](#)
- [~Album](#)
  - [Digikam::Album, 255](#)
- [~AlbumModificationHelper](#)
  - [Digikam::AlbumModificationHelper, 316](#)
- [~AlbumSelectTreeView](#)
  - [Digikam::AlbumSelectTreeView, 349](#)
- [~BackendGeonamesRG](#)
  - [Digikam::BackendGeonamesRG, 435](#)
- [~BackendGeonamesUSRG](#)
  - [Digikam::BackendGeonamesUSRG, 438](#)
- [~BackendGoogleMaps](#)
  - [Digikam::BackendGoogleMaps, 442](#)
- [~BackendMarble](#)
  - [Digikam::BackendMarble, 450](#)
- [~BackendOsmRG](#)
  - [Digikam::BackendOsmRG, 457](#)
- [~CoreDB](#)
  - [Digikam::CoreDB, 678](#)
- [~DConfigDlg](#)
  - [Digikam::DConfigDlg, 911](#)
- [~DConfigDlgMngr](#)
  - [Digikam::DConfigDlgMngr, 921](#)
- [~DConfigDlgModel](#)
  - [Digikam::DConfigDlgModel, 926](#)
- [~DConfigDlgView](#)
  - [Digikam::DConfigDlgView, 939](#)
- [~DConfigDlgWdg](#)
  - [Digikam::DConfigDlgWdg, 946](#)
- [~DConfigDlgWdgItem](#)
  - [Digikam::DConfigDlgWdgItem, 953](#)
- [~DConfigDlgWdgModel](#)
  - [Digikam::DConfigDlgWdgModel, 958](#)
- [~DDatePicker](#)
  - [Digikam::DDatePicker, 972](#)
- [~DDateTimeEdit](#)
  - [Digikam::DDateTimeEdit, 988](#)
- [~DFontProperties](#)
  - [Digikam::DFontProperties, 1029](#)
- [~DHueSaturationSelector](#)
  - [Digikam::DHueSaturationSelector, 1040](#)
- [~DNotificationPopup](#)
  - [Digikam::DNotificationPopup, 1249](#)
- [~DNotificationWidget](#)
  - [Digikam::DNotificationWidget, 1260](#)
- [~DPlainTextEdit](#)
  - [Digikam::DPlainTextEdit, 1295](#)
- [~DPlugin](#)
  - [Digikam::DPlugin, 1299](#)
- [~DPluginBqm](#)
  - [Digikam::DPluginBqm, 1308](#)
- [~DPluginDImg](#)
  - [Digikam::DPluginDImg, 1324](#)
- [~DPluginEditor](#)
  - [Digikam::DPluginEditor, 1329](#)
- [~DPluginGeneric](#)
  - [Digikam::DPluginGeneric, 1333](#)
- [~DPluginRawImport](#)
  - [Digikam::DPluginRawImport, 1342](#)
- [~DPopupFrame](#)
  - [Digikam::DPopupFrame, 1350](#)
- [~DRawDecoder](#)
  - [Digikam::DRawDecoder, 1368](#)
- [~DRawDecoderSettings](#)
  - [Digikam::DRawDecoderSettings, 1377](#)
- [~DRawDecoding](#)
  - [Digikam::DRawDecoding, 1385](#)
- [~DRawInfo](#)
  - [Digikam::DRawInfo, 1389](#)
- [~DTextEdit](#)
  - [Digikam::DTextEdit, 1413](#)
- [~DynamicThread](#)
  - [Digikam::DynamicThread, 1453](#)
- [~EditableSearchTreeView](#)
  - [Digikam::EditableSearchTreeView, 1463](#)
- [~ExifToolProcess](#)
  - [Digikam::ExifToolProcess, 1528](#)
- [~FileSaveOptionsBox](#)
  - [Digikam::FileSaveOptionsBox, 1687](#)
- [~FilterSideBarWidget](#)
  - [Digikam::FilterSideBarWidget, 1716](#)
- [~HistogramPainter](#)
  - [Digikam::HistogramPainter, 1887](#)
- [~ItemInfo](#)
  - [Digikam::ItemInfo, 2295](#)
- [~JpegRotator](#)
  - [Digikam::JPEGUtils::JpegRotator, 2478](#)
- [~LoadSaveThread](#)
  - [Digikam::LoadSaveThread, 2570](#)
- [~MapWidget](#)
  - [Digikam::MapWidget, 2622](#)
- [~MetaEngine](#)
  - [Digikam::MetaEngine, 2674](#)

- ~NormalSearchTreeView
  - Digikam::NormalSearchTreeView, 2780
- ~RGInfo
  - Digikam::RGInfo, 2978
- ~RGWidget
  - Digikam::RGWidget, 2989
- ~SchemeManager
  - Digikam::SchemeManager, 3024
- ~SearchModificationHelper
  - Digikam::SearchModificationHelper, 3115
- ~SidebarWidget
  - Digikam::SidebarWidget, 3225
- ~SqueezedComboBox
  - Digikam::SqueezedComboBox, 3253
- ~StateSavingObject
  - Digikam::StateSavingObject, 3261
- ~TagFilterView
  - Digikam::TagFilterView, 3350
- ~TagFolderView
  - Digikam::TagFolderView, 3355
- ~TagModificationHelper
  - Digikam::TagModificationHelper, 3381
- a01
  - Digikam::GeodeticCalculator, 1767
- abortInitialization
  - Digikam::ScanController, 3011
- aboutToDeactivate
  - Digikam::RecognitionWorker, 2949
  - Digikam::TrainerWorker, 3533
  - Digikam::WorkerObject, 3618
- AboutToEditMetadata
  - Digikam::ItemMetadataAdjustmentHint, 2344
- aboutToQuitLoop
  - Digikam::WorkerObject, 3618
- aboutToSetInfo
  - Digikam::FaceGroup, 1557
- aboutToShowContextMenu
  - Digikam::DDateTable, 981
- absoluteToRelative
  - Digikam::TagRegion, 3394
- AbstractAlbumModel
  - Digikam::AbstractAlbumModel, 143
- AbstractAlbumTreeView
  - Digikam::AbstractAlbumTreeView, 150
- AbstractAlbumTreeViewSelectComboBox
  - Digikam::AbstractAlbumTreeViewSelectComboBox, 161
- AbstractCheckableAlbumModel
  - Digikam::AbstractCheckableAlbumModel, 168
- AbstractCheckableAlbumTreeView
  - Digikam::AbstractCheckableAlbumTreeView, 174
- AbstractSearchGroupContainer
  - Digikam::AbstractSearchGroupContainer, 196
- AbstractWidgetDelegateOverlay
  - Digikam::AbstractWidgetDelegateOverlay, 202
- acceptedCharacters
  - Digikam::DPlainTextEdit, 1295
  - Digikam::DTextEdit, 1413
- accepts
  - Digikam::AbstractItemDragDropHandler, 188
  - Digikam::AlbumDragDropHandler, 265
  - Digikam::AlbumModelDragDropHandler, 314
  - Digikam::ImportDragDropHandler, 2047
  - Digikam::ItemDragDropHandler, 2227
  - Digikam::MapDragDropHandler, 2614
  - Digikam::TagDragDropHandler, 3341
  - ShowFoto::ShowfotoDragDropHandler, 3671
- acceptsActivation
  - Digikam::ImportDelegate, 2036
  - Digikam::ImportThumbnailDelegate, 2124
  - Digikam::ItemDelegate, 2212
  - Digikam::ItemThumbnailDelegate, 2430
  - Digikam::ItemViewDelegate, 2452
  - Digikam::ItemViewImportDelegate, 2462
  - ShowFoto::ShowfotoDelegate, 3665
  - ShowFoto::ShowfotoItemViewDelegate, 3708
  - ShowFoto::ShowfotoThumbnailDelegate, 3756
- acceptsMimeType
  - Digikam::AbstractItemDragDropHandler, 188
  - Digikam::AlbumModelDragDropHandler, 314
- acceptsMouseClicked
  - Digikam::ImportPreviewView, 2093
  - Digikam::ItemPreviewView, 2363
- acceptsToolTip
  - Digikam::DItemDelegate, 1152
  - Digikam::ImportDelegate, 2036
  - Digikam::ItemDelegate, 2212
  - Digikam::ItemViewDelegate, 2452
  - Digikam::ItemViewImportDelegate, 2462
  - ShowFoto::ShowfotoDelegate, 3665
  - ShowFoto::ShowfotoItemViewDelegate, 3708
- accessCol
  - Digikam, 135
- AccessMode
  - Digikam::LoadSaveThread, 2569
- accessMode
  - Digikam::SharedLoadingTask, 3194
- AccessModeRead
  - Digikam::LoadSaveThread, 2569
- AccessModeReadWrite
  - Digikam::LoadSaveThread, 2569
- accessRow
  - Digikam, 135
- Action
  - Digikam::ExifToolProcess, 1526
- action
  - Digikam::DImageHistory::Entry, 1070
  - Digikam::Token, 3511
- ActionCategory
  - Digikam::DPluginAction, 1304
- actionCollections
  - Digikam::TagsActionMngr, 3397
- actionForIndex
  - Digikam::ActionItemModel, 210
- ActionItemModel
  - Digikam::ActionItemModel, 210

- ActionJob
  - Digikam::ActionJob, [211](#)
- ActionJobCollection
  - Digikam, [123](#)
- actionRequested
  - Digikam::DCategoryDrawer, [892](#)
- actions
  - Digikam::DPluginEditor, [1329](#)
  - Digikam::DPluginGeneric, [1333](#)
- actionString
  - Digikam::ExifToolParser::Private, [1525](#)
- ActionType
  - Digikam::DPluginAction, [1304](#)
- actionType
  - Digikam::DPluginAction, [1304](#)
- activated
  - Digikam::DigikamItemView, [1063](#)
  - Digikam::ImportCategorizedView, [2016](#)
  - Digikam::ImportIconView, [2063](#)
  - Digikam::ItemCategorizedView, [2187](#)
- ActiveBackground
  - Digikam::SchemeManager, [3022](#)
- activated
  - Digikam::DPluginConfView, [1311](#)
- ActiveIconText
  - Digikam::DMultiTabBar, [1199](#)
- activeNextTab
  - Digikam::Sidebar, [3217](#)
- activePreviousTab
  - Digikam::Sidebar, [3217](#)
- ActiveText
  - Digikam::SchemeManager, [3024](#)
- adaptColumnsToContent
  - Digikam::AbstractAlbumTreeView, [150](#)
- add
  - Digikam::FaceTagsEditor, [1639](#)
  - Digikam::KDTreeBase, [2488](#)
- addAction
  - Digikam::ContextMenuHelper, [661](#), [662](#)
  - Digikam::DNotificationWidget, [1260](#)
  - Digikam::ImportContextMenuHelper, [2023](#), [2024](#)
- addActionButton
  - Digikam::DConfigDlg, [912](#)
- addActionDeleteFaceTag
  - Digikam::ContextMenuHelper, [662](#)
- addActionNewAlbum
  - Digikam::ContextMenuHelper, [662](#)
- addActionNewTag
  - Digikam::ContextMenuHelper, [662](#)
- addActions
  - Digikam::AbstractAlbumTreeView::ContextMenuElement, [157](#)
- addActionsToConfigurationMenu
  - Digikam::BackendGoogleMaps, [442](#)
  - Digikam::BackendMarble, [450](#)
- addActionTagToFaceTag
  - Digikam::ContextMenuHelper, [663](#)
- addAlbum
  - Digikam::CoreDB, [678](#)
- addAlbumCheckUncheckActions
  - Digikam::ContextMenuHelper, [663](#)
- addAlbumRoot
  - Digikam::CoreDB, [679](#)
- addAlbums
  - Digikam::AlbumHistory, [279](#)
- addAsReferredImage
  - Digikam::DImg, [1078](#)
- addAssignTagsMenu
  - Digikam::ContextMenuHelper, [663](#)
  - Digikam::ImportContextMenuHelper, [2024](#)
- addCamItemInfoSynchronously
  - Digikam::ImportItemModel, [2068](#)
- addCheckUncheckContextMenuActions
  - Digikam::AbstractAlbumTreeViewSelectComboBox, [162](#)
- addColumnAt
  - Digikam::TableViewModel, [3325](#)
- addComment
  - Digikam::ItemComments, [2195](#)
- addCreateTagFromAddressbookMenu
  - Digikam::ContextMenuHelper, [663](#)
- addCurrent
  - Digikam::AltLangStrEdit, [371](#)
- addCurrentUniqueImageId
  - Digikam::DImg, [1078](#)
- addCustomContextMenuActions
  - Digikam::AbstractAlbumTreeView, [150](#)
  - Digikam::AlbumSelectTreeView, [349](#)
  - Digikam::EditableSearchTreeView, [1463](#)
  - Digikam::NormalSearchTreeView, [2780](#)
  - Digikam::TagCheckView, [3336](#)
  - Digikam::TagFilterView, [3350](#)
  - Digikam::TagFolderView, [3356](#)
- addDataInTree
  - Digikam::RGTagModel, [2983](#)
- Added
  - Digikam::CollectionImageChangeset, [605](#)
- AddEntryToExisting
  - Digikam::ItemCopyright, [2204](#)
- addExportMenu
  - Digikam::ContextMenuHelper, [664](#)
- addExternalTags
  - Digikam::RGTagModel, [2983](#)
- addFace
  - Digikam::FaceGroup, [1557](#)
- addGenerator
  - Digikam::DImgFilterManager, [1100](#)
- addGotoMenu
  - Digikam::ContextMenuHelper, [664](#)
- addGroupMenu
  - Digikam::ContextMenuHelper, [664](#)
  - Digikam::ImportContextMenuHelper, [2025](#)
- addGroupToLayout
  - Digikam::AbstractSearchGroupContainer, [196](#)
  - Digikam::SearchGroup, [3104](#)
  - Digikam::SearchView, [3139](#)

- addHeadline
  - Digikam::ItemComments, [2195](#)
- addHistory
  - Digikam::ItemHistoryGraph, [2275](#)
- addHook
  - Digikam::ItemQueryPostHooks, [2384](#)
- addId
  - Digikam::DbKeysCollection, [865](#)
- addIdentity
  - Digikam::FacialRecognitionWrapper, [1652](#)
  - Digikam::IdentityProvider, [1951](#)
- addIdentityAttributes
  - Digikam::FacialRecognitionWrapper, [1652](#)
- addIdentityDebug
  - Digikam::FacialRecognitionWrapper, [1653](#)
  - Digikam::IdentityProvider, [1951](#)
- addImageMetadata
  - Digikam::CoreDB, [679](#)
- addImageRelation
  - Digikam::CoreDB, [679](#)
- addImageRelations
  - Digikam::CoreDB, [679](#)
- addImageTagProperty
  - Digikam::CoreDB, [680](#)
- addImportMenu
  - Digikam::ContextMenuHelper, [664](#)
- addIQSAction
  - Digikam::ContextMenuHelper, [665](#)
- addItem
  - Digikam::CoreDB, [680](#)
  - Digikam::DExpanderBox, [1015](#)
  - Digikam::ItemVisibilityController, [2473](#)
  - Digikam::SinglePhotoPreviewLayout, [3243](#)
  - Digikam::TagMngrListModel, [3364](#)
- addItemInfo
  - Digikam::ItemModel, [2349](#)
- addItemInformation
  - Digikam::CoreDB, [680](#)
- addItemInfoSynchronously
  - Digikam::ItemModel, [2349](#)
- addItemPosition
  - Digikam::CoreDB, [680](#)
- addItemTag
  - Digikam::CoreDB, [681](#)
- AdditionalRoles
  - Digikam::DCategorizedSortFilterProxyModel, [878](#)
- addLabelsAction
  - Digikam::ContextMenuHelper, [665](#)
  - Digikam::ImportContextMenuHelper, [2025](#)
- addListener
  - Digikam::SharedLoadingTask, [3194](#)
- addLoadingProcess
  - Digikam::LoadingCache, [2545](#)
- addLocation
  - Digikam::CollectionManager, [613](#)
- addManually
  - Digikam::FacePipeline, [1567](#)
- addMoreWorkers
  - Digikam::FacePipelineDetect, [1579](#)
  - Digikam::FacePipelineDetectRecognize, [1583](#)
  - Digikam::FacePipelineEdit, [1587](#)
  - Digikam::FacePipelineRecognize, [1601](#)
  - Digikam::FacePipelineReset, [1605](#)
  - Digikam::FacePipelineRetrain, [1609](#)
- addNewData
  - Digikam::RGTagModel, [2983](#)
- addNewTag
  - Digikam::RGTagModel, [2983](#)
- addNormalTag
  - Digikam::FaceTagsEditor, [1639](#)
  - Digikam::FaceUtils, [1650](#)
- addObject
  - Digikam::VisibilityController, [3603](#)
- addOpenAndNavigateActions
  - Digikam::ContextMenuHelper, [665](#)
- addOverlay
  - Digikam::ImportCategorizedView, [2016](#)
  - ShowFoto::ShowfotoCategorizedView, [3654](#)
- addPage
  - Digikam::DConfigDlg, [912](#)
  - Digikam::DConfigDlgWdg, [946](#), [947](#)
  - Digikam::DConfigDlgWdgModel, [958](#)
- addPrepareHook
  - Digikam::ItemFilterModel, [2243](#)
- addProfile
  - Digikam::lccProfilesMenuAction, [1930](#)
- addProfiles
  - Digikam::lccProfilesMenuAction, [1930](#)
- addProfileSqueezed
  - Digikam::lccProfilesComboBox, [1927](#)
- addProfilesSqueezed
  - Digikam::lccProfilesComboBox, [1927](#)
- addProgressItem
  - Digikam::PrivateProgressItemCreator, [2865](#)
  - Digikam::ProgressManager, [2877](#)
- addProperty
  - Digikam::ItemTagPair, [2418](#)
  - Digikam::TagProperties, [3386](#)
- addQueueManagerMenu
  - Digikam::ContextMenuHelper, [666](#)
- addRelations
  - Digikam::ItemHistoryGraph, [2276](#)
- addRemoveAllTags
  - Digikam::ContextMenuHelper, [666](#)
- addRemoveTagsMenu
  - Digikam::ContextMenuHelper, [666](#)
  - Digikam::ImportContextMenuHelper, [2025](#)
- addRotateMenu
  - Digikam::ImportContextMenuHelper, [2026](#)
- addScannedHistory
  - Digikam::ItemHistoryGraph, [2276](#)
- addSearch
  - Digikam::CoreDB, [681](#)
- addSeparator
  - Digikam::ContextMenuHelper, [666](#)
  - Digikam::ImportContextMenuHelper, [2026](#)



- addServicesMenu
  - Digikam::ContextMenuHelper, [667](#)
  - Digikam::ImportContextMenuHelper, [2026](#)
- addShowfotoItemInfoSynchronously
  - ShowFoto::ShowfotoItemModel, [3698](#)
- addSpacerTag
  - Digikam::RGTagModel, [2984](#)
- addSqueezedItem
  - Digikam::SqueezedComboBox, [3253](#)
- addStandardActionCopy
  - Digikam::ContextMenuHelper, [667](#)
- addStandardActionCut
  - Digikam::ContextMenuHelper, [667](#)
- addStandardActionItemDelete
  - Digikam::ContextMenuHelper, [667](#)
- addStandardActionLightTable
  - Digikam::ContextMenuHelper, [668](#)
- addStandardActionPaste
  - Digikam::ContextMenuHelper, [668](#)
- addStandardActionThumbnail
  - Digikam::ContextMenuHelper, [668](#)
- addSubMenu
  - Digikam::ContextMenuHelper, [669](#)
  - Digikam::ImportContextMenuHelper, [2026](#)
- addSubPage
  - Digikam::DConfigDlg, [913](#)
  - Digikam::DConfigDlgWdg, [947](#), [948](#)
  - Digikam::DConfigDlgWdgModel, [959](#)
- addTag
  - Digikam::CoreDB, [682](#)
- addTagPaths
  - Digikam::ItemInfo, [2296](#)
- addTagProperty
  - Digikam::CoreDB, [682](#)
- addTagsToItems
  - Digikam::CoreDB, [682](#)
- addTitle
  - Digikam::ItemComments, [2195](#)
- addToDownloadHistory
  - Digikam::CoreDB, [683](#)
- addToGroup
  - Digikam::ItemInfo, [2296](#)
- addToken
  - Digikam::Rule, [2996](#)
- addToXmpTagStringBag
  - Digikam::DMetadata, [1184](#)
  - Digikam::MetaEngine, [2674](#)
- addTraining
  - Digikam::IdentityProvider, [1951](#)
- addUngroupedModel
  - Digikam::MapWidget, [2622](#)
- addVideoMetadata
  - Digikam::CoreDB, [683](#)
- addWidget
  - Digikam::DConfigDlgMgr, [921](#)
  - Digikam::VisibilityController, [3603](#)
- AdjacencyFlags
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1841](#)
- adjustBackground
  - Digikam::SchemeManager, [3025](#)
- adjustBoundariesToGroupedMarkers
  - Digikam::MapWidget, [2622](#)
- adjustedEnvironmentForApplImage
  - Digikam, [127](#)
- adjustForeground
  - Digikam::SchemeManager, [3025](#)
- AdjustmentStatus
  - Digikam::ItemMetadataAdjustmentHint, [2343](#)
- adjustReferredImages
  - Digikam::DImageHistory, [1068](#)
- adjustToOrientation
  - Digikam::TagRegion, [3394](#)
- advance
  - Digikam::ProgressItem, [2870](#)
- AdvancedMetadataTab
  - Digikam::AdvancedMetadataTab, [236](#)
- affectsMultiple
  - Digikam::ItemDelegateOverlay, [2218](#)
- Album
  - Digikam::Album, [255](#)
- album
  - Digikam::CollectionManager, [614](#)
  - Digikam::CoreDbUrl, [739](#)
- albumAt
  - Digikam::ItemCategorizedView, [2187](#)
- albumChooser
  - Digikam::DBInfolface, [854](#)
  - Digikam::DInfoInterface, [1128](#)
- albumChooserItems
  - Digikam::DBInfolface, [854](#)
- albumCleared
  - Digikam::AbstractAlbumModel, [143](#)
  - Digikam::AbstractCheckableAlbumModel, [168](#)
  - Digikam::AbstractCountingAlbumModel, [180](#)
- AlbumCopyMoveHint
  - Digikam::AlbumCopyMoveHint, [263](#)
- albumCount
  - Digikam::AbstractCountingAlbumModel, [180](#)
- albumData
  - Digikam::AbstractAlbumModel, [143](#)
  - Digikam::AbstractCheckableAlbumModel, [168](#)
  - Digikam::AbstractCountingAlbumModel, [180](#)
  - Digikam::AlbumModel, [312](#)
  - Digikam::SearchModel, [3112](#)
  - Digikam::TagModel, [3378](#)
- AlbumDataRole
  - Digikam::AbstractAlbumModel, [142](#)
- albumForId
  - Digikam::AbstractCountingAlbumModel, [181](#)
  - Digikam::AlbumModel, [312](#)
  - Digikam::DateAlbumModel, [807](#)
  - Digikam::SearchModel, [3112](#)
  - Digikam::TagModel, [3378](#)
- albumForIndex

- Digikam::AbstractAlbumModel, 143
- albumForSelectedItems
  - Digikam::AlbumLabelsSearchHandler, 282
- AlbumGlobalIdRole
  - Digikam::AbstractAlbumModel, 142
- albumId
  - Digikam::ItemInfo, 2296
- AlbumIdRole
  - Digikam::AbstractAlbumModel, 142
- albumInfo
  - Digikam::DBInfolface, 854
- albumItems
  - Digikam::DBInfolface, 854
- AlbumModel
  - Digikam::AlbumModel, 312
- albumModel
  - Digikam::AbstractCheckableAlbumTreeView, 175
- AlbumModificationHelper
  - Digikam::AlbumModificationHelper, 316
- albumName
  - Digikam::AbstractCountingAlbumModel, 181
  - Digikam::DateAlbumModel, 807
- AlbumParser
  - Digikam::AlbumParser, 321
- AlbumPointerRole
  - Digikam::AbstractAlbumModel, 142
- albumRelativePath
  - Digikam::ItemInfoCache, 2309
- albumRoot
  - Digikam::CollectionManager, 614
  - Digikam::CoreDbUrl, 739
- albumRootId
  - Digikam::CoreDbUrl, 739
  - Digikam::ItemInfo, 2296
- albumRootLabel
  - Digikam::CollectionManager, 614
- albumRootPath
  - Digikam::CollectionLocation, 609
  - Digikam::CollectionManager, 614
  - Digikam::CoreDbUrl, 739
- AlbumSelectors
  - Digikam::AlbumSelectors, 341
- AlbumSelectTreeView
  - Digikam::AlbumSelectTreeView, 349
- albumsItems
  - Digikam::DBInfolface, 854
- albumsListing
  - Digikam::AlbumsDBJobsThread, 329
- AlbumSortRole
  - Digikam::AbstractAlbumModel, 142
- AlbumTitleRole
  - Digikam::AbstractAlbumModel, 142
- albumTitles
  - Digikam::AlbumManager, 286
- albumType
  - Digikam::AbstractAlbumModel, 143
- AlbumTypeRole
  - Digikam::AbstractAlbumModel, 142
- albumUrl
  - Digikam::CoreDbUrl, 739
- alignFace
  - Digikam::DNNOpenFaceExtractor, 1240
  - Digikam::DNNSFaceExtractor, 1244
- All
  - Digikam::QueueListView, 2893
- allActions
  - Digikam::DXmlGuiWindow, 1446
- allAlbumItems
  - Digikam::DBInfolface, 855
  - Digikam::DMetaInfolface, 1194
- allAlbumsCleared
  - Digikam::AbstractAlbumModel, 143
  - Digikam::AbstractCheckableAlbumModel, 168
  - Digikam::AbstractCountingAlbumModel, 181
- allAvailableAlbumRootPaths
  - Digikam::CollectionManager, 614
- allAvailableLocations
  - Digikam::CollectionManager, 614
- allDAlbums
  - Digikam::AlbumManager, 286
- allFilePaths
  - Digikam::VersionFileOperation, 3574
- AllIconsText
  - Digikam::DMultiTabBar, 1199
- allIdentities
  - Digikam::FacialRecognitionWrapper, 1653
  - Digikam::IdentityProvider, 1951
- allImages
  - Digikam::ItemHistoryGraph, 2276
- allItemInfoListFromCurrentQueue
  - Digikam::BqmlInfolface, 529
- AllItems
  - Digikam::AutotagsAssignment, 431
  - Digikam::ImageQualitySorter, 1986
- allLanguagesRFC3066
  - Digikam::AltLangStrEdit, 371
- allLocations
  - Digikam::CollectionManager, 615
- allNeedGroupResolving
  - Digikam::ItemIconView, 2290
- allowLift
  - Digikam::CoreDbOperationGroup, 734
  - Digikam::FaceDbOperationGroup, 1552
- allPAlbums
  - Digikam::AlbumManager, 287
- allPersonNames
  - Digikam::FaceTags, 1635
- allPersonPaths
  - Digikam::FaceTags, 1635
- allPersonTags
  - Digikam::FaceTags, 1635
- allRefreshingFinished
  - Digikam::ImportItemModel, 2068
  - Digikam::ItemModel, 2349
  - ShowFoto::ShowfotoItemModel, 3698
- allSAlbums

- Digikam::AlbumManager, [287](#)
- allAlbums
  - Digikam::AlbumManager, [287](#)
- allUrls
  - Digikam::ItemIconView, [2290](#)
- AlreadyScannedHandling
  - Digikam::FaceScanSettings, [1623](#)
- AlternateBackground
  - Digikam::SchemeManager, [3022](#)
- altitude
  - Digikam::ItemPosition, [2355](#)
- altitudeFormatted
  - Digikam::ItemPosition, [2355](#)
- AltLangMap
  - Digikam::MetaEngine, [2672](#)
- AltLangStrEdit
  - Digikam::AltLangStrEdit, [371](#)
- AlwaysShowInclusiveCounts
  - Digikam::AbstractAlbumTreeView, [150](#)
- ambientAcceleration
  - Digikam::DRawInfo, [1389](#)
- ambientElevationAngle
  - Digikam::DRawInfo, [1389](#)
- ambientHumidity
  - Digikam::DRawInfo, [1389](#)
- ambientPressure
  - Digikam::DRawInfo, [1389](#)
- ambientTemperature
  - Digikam::DRawInfo, [1389](#)
- ambientWaterDepth
  - Digikam::DRawInfo, [1390](#)
- analyser
  - Digikam::EditorToolThreaded, [1481](#)
- anchor
  - Digikam::DNotificationPopup, [1250](#)
- animatedHide
  - Digikam::DNotificationWidget, [1261](#)
- animatedShow
  - Digikam::DNotificationWidget, [1261](#)
- animatedShowTemporized
  - Digikam::DNotificationWidget, [1261](#)
- AnimatedVisibility
  - Digikam::AnimatedVisibility, [377](#)
- aperture
  - Digikam::DRawInfo, [1390](#)
- append
  - Digikam::DTrashItemModel, [1423](#)
- appendButton
  - Digikam::DMultiTabBar, [1200](#)
- appendControlButtonsWidget
  - Digikam::DItemsList, [1159](#)
- AppendDecorationRole
  - Digikam::SetupCollectionModel, [3171](#)
- appendJobs
  - Digikam::ActionThreadBase, [220](#)
- appendPlugin
  - Digikam::DPluginLoader::Private, [1339](#)
- appendPluginToBlackList
  - Digikam::DPluginLoader, [1336](#)
- appendPluginToWhiteList
  - Digikam::DPluginLoader, [1336](#)
- appendTab
  - Digikam::DMultiTabBar, [1200](#)
  - Digikam::Sidebar, [3217](#)
- appliedFilterActions
  - Digikam::FilterActionFilter, [1711](#)
- apply
  - Digikam::BatchTool, [466](#)
  - Digikam::DImgBuiltinFilter, [1091](#)
  - Digikam::DPluginConfView, [1311](#)
  - Digikam::IccTransform, [1942](#)
  - Digikam::ItemComments, [2195](#)
  - Digikam::ItemPosition, [2355](#)
- APPLY\_CHANGES
  - Digikam::ExifToolProcess, [1527](#)
- APPLY\_CHANGES\_EXV
  - Digikam::ExifToolProcess, [1527](#)
- APPLY\_METADATA\_FILE
  - Digikam::ExifToolProcess, [1527](#)
- applyCacheToBackend
  - Digikam::MapWidget, [2623](#)
- applyCacheToWidget
  - Digikam::BackendMarble, [450](#)
- applyChanges
  - Digikam::ExifToolParser, [1521](#)
  - Digikam::MetaEngine, [2674](#)
- applyFilter
  - Digikam::BatchTool, [466](#)
- applyMetadata
  - Digikam::FileActionMngrDatabaseWorker, [1672](#)
- applyMetadataFile
  - Digikam::ExifToolParser, [1522](#)
- applySettings
  - Digikam::AlbumFolderViewSideBarWidget, [276](#)
  - Digikam::DateFolderViewSideBarWidget, [813](#)
  - Digikam::FuzzySearchSideBarWidget, [1755](#)
  - Digikam::GPSSearchSideBarWidget, [1830](#)
  - Digikam::ImportItemPropertiesSideBarImport, [2075](#)
  - Digikam::LabelsSideBarWidget, [2496](#)
  - Digikam::PeopleSideBarWidget, [2832](#)
  - Digikam::QueuePool, [2900](#)
  - Digikam::SearchSideBarWidget, [3120](#)
  - Digikam::SidebarWidget, [3225](#)
  - Digikam::TagViewSideBarWidget, [3434](#)
  - Digikam::TimelineSideBarWidget, [3505](#)
- applySettingsToWidget
  - Digikam::GeolocationSettings, [1778](#)
- applyTagIdentityMapping
  - Digikam::FaceTags, [1636](#)
- ApplyTransform
  - Digikam::LoadingDescription, [2552](#)
- applyTransform
  - Digikam::Canvas, [565](#)
- areaCoordinates
  - Digikam::CoreDbUrl, [740](#)

- arrowDirection
  - Digikam::DSelector, [1400](#)
- asDateTimeLocal
  - Digikam, [127](#)
- asDateTimeUTC
  - Digikam, [127](#)
- asDBDateTime
  - Digikam::BdEngineBackend, [482](#)
- asDelegate
  - Digikam::ItemDelegateOverlayContainer, [2221](#)
  - Digikam::ItemViewDelegate, [2452](#)
  - Digikam::ItemViewImportDelegate, [2462](#)
  - Digikam::VersionsDelegate, [3584](#)
  - ShowFoto::ShowfotoItemViewDelegate, [3708](#)
- askGroupingOperateOnAll
  - Digikam::ApplicationSettings, [388](#)
- aspectRatio
  - Digikam::ItemInfo, [2296](#)
- aspectRatioToString
  - Digikam::ItemPropertiesTab, [2380](#)
- asQObject
  - Digikam::ParallelAdapter< A >, [2821](#)
  - Digikam::ParallelWorkers, [2825](#)
- asQtCaseSensitivity
  - Digikam::CollectionLocation, [609](#)
- assignColorLabel
  - Digikam::FileActionMngrDatabaseWorker, [1672](#)
- assignDate
  - Digikam::DDateEdit, [968](#)
- assigned
  - Digikam::AssignNameWidget, [407](#)
- AssignNameWidget
  - Digikam::AssignNameWidget, [407](#)
- assignPickLabel
  - Digikam::FileActionMngrDatabaseWorker, [1672](#)
- assignRating
  - Digikam::FileActionMngrDatabaseWorker, [1672](#)
- assignTag
  - Digikam::ItemTagPair, [2418](#)
- assignTags
  - Digikam::FileActionMngrDatabaseWorker, [1672](#)
- asView
  - Digikam::DragDropViewImplementation, [1365](#)
- atEnd
  - Digikam::EmptyImageListProvider, [1505](#)
  - Digikam::QListImageListProvider, [2890](#)
- atLeastOneUpdateToProcess
  - Digikam::TimeAdjustContainer, [3502](#)
- attribute
  - Digikam::Identity, [1950](#)
- attributeForType
  - Digikam::FaceTagsIface, [1646](#)
- attributesForFlags
  - Digikam::FaceTagsIface, [1646](#)
- attributesMap
  - Digikam::Identity, [1950](#)
- autoBrightness
  - Digikam::DRawDecoderSettings, [1377](#)
- AutoCrop
  - Digikam::AutoCrop, [418](#)
- autoDelete
  - Digikam::DNotificationPopup, [1250](#)
- autoExifTransform
  - Digikam::JPEGUtils::JpegRotator, [2478](#)
- autoHideTimeout
  - Digikam::DConfigDlgTitle, [931](#)
- autoInnerCrop
  - Digikam::AutoCrop, [418](#)
- AutotagsAssignment
  - Digikam::AutotagsAssignment, [431](#)
- AutotagsAssignmentScanMode
  - Digikam::AutotagsAssignment, [430](#)
- autoWBAdjustmentFromColor
  - Digikam::WBFilter, [3609](#)
- availablePairs
  - Digikam::ItemTagPair, [2418](#)
- avg
  - Digikam::Haar::SignatureData, [1873](#)
- AVI
  - Digikam::VidSlideSettings, [3595](#)
- azimuth
  - Digikam::GeodeticCalculator, [1763](#)
- Backend
  - Digikam::MetaEngine, [2673](#)
- backend
  - Digikam::CoreDbAccess, [722](#)
- BackendGeonamesRG
  - Digikam::BackendGeonamesRG, [435](#)
- BackendGeonamesUSRG
  - Digikam::BackendGeonamesUSRG, [438](#)
- backendHumanName
  - Digikam::BackendGoogleMaps, [442](#)
  - Digikam::BackendMarble, [450](#)
  - Digikam::LookupAltitudeGeonames, [2587](#)
- backendName
  - Digikam::BackendGeonamesRG, [435](#)
  - Digikam::BackendGeonamesUSRG, [438](#)
  - Digikam::BackendGoogleMaps, [443](#)
  - Digikam::BackendMarble, [450](#)
  - Digikam::BackendOsmRG, [457](#)
  - Digikam::LookupAltitudeGeonames, [2587](#)
  - Digikam::MetaEngine, [2675](#)
  - Digikam::RGBBackend, [2978](#)
- BackendOsmRG
  - Digikam::BackendOsmRG, [457](#)
- background
  - Digikam::SchemeManager, [3025](#)
- backgroundColor
  - Digikam::DFontProperties, [1029](#)
- BackgroundRole
  - Digikam::SchemeManager, [3022](#)
- backup
  - Digikam::Sidebar, [3217](#)
- Balloon
  - Digikam::DNotificationPopup, [1249](#)
- BarMode

- Digikam::DZoomBar, [1457](#)
- baselineExposure
  - Digikam::DRawInfo, [1390](#)
- baseName
  - Digikam::DefaultVersionNamingScheme, [1000](#)
  - Digikam::VersionNamingScheme, [3580](#)
- BaseTool
  - Digikam::BatchTool, [465](#)
- BasicDImgFilterGenerator
  - Digikam::BasicDImgFilterGenerator< T >, [462](#)
- BatchSetList
  - Digikam, [123](#)
- BatchTool
  - Digikam::BatchTool, [466](#)
- BatchToolGroup
  - Digikam::BatchTool, [465](#)
- BatchToolSettings
  - Digikam, [124](#)
- BatchToolsList
  - Digikam, [124](#)
- bcg
  - Digikam::DRawDecoding, [1386](#)
- BdEngineBackend
  - Digikam::BdEngineBackend, [482](#)
- beginFileMetadataWrite
  - Digikam::ScanController, [3011](#)
- beginTransaction
  - Digikam::BdEngineBackend, [482](#)
- BehaviorEnum
  - Digikam::ICCSettingsContainer, [1941](#)
- bestMatchesForImageWithThreshold
  - Digikam::HaarIface, [1877](#)
- bestRepresentativeIndexFromList
  - Digikam::AbstractMarkerTiler, [191](#)
  - Digikam::GeoModelHelper, [1781](#)
  - Digikam::GPSGeofaceModelHelper, [1802](#)
  - Digikam::GPSMarkerTiler, [1821](#)
  - Digikam::ItemGPSModelHelper, [2273](#)
  - Digikam::ItemMarkerTiler, [2340](#)
  - Digikam::MapViewModelHelper, [2616](#)
- bindAlbum
  - Digikam::AlbumModificationHelper, [316](#)
- bindMultipleTags
  - Digikam::TagModificationHelper, [3381](#)
- bindTag
  - Digikam::TagModificationHelper, [3381](#)
- bitBlendImage
  - Digikam::DImg, [1078](#)
- bitBlendImageOnColor
  - Digikam::DImg, [1078](#)
- bitBltImage
  - Digikam::DImg, [1078](#)
- bitsDepth
  - Digikam::DImg, [1079](#)
- black
  - Digikam::WBContainer, [3605](#)
- blackPoint
  - Digikam::DRawDecoderSettings, [1377](#)
- Digikam::DRawInfo, [1390](#)
- blackPointCh
  - Digikam::DRawInfo, [1390](#)
- BlackWhiteConversionType
  - Digikam::BWSepiaContainer, [532](#)
- blendZero
  - Digikam::DColor, [898](#)
- blockedEventTypes
  - Digikam::DWItemDelegate, [1435](#)
- BLUERAY
  - Digikam::VidSlideSettings, [3596](#)
- BlurFilter
  - Digikam::BlurFilter, [508](#)
- BorderFilter
  - Digikam::BorderFilter, [525](#)
- bottomBarPixmap
  - Digikam::SearchView, [3139](#)
- boundAlbum
  - Digikam::AlbumModificationHelper, [316](#)
- boundingRect
  - Digikam::DImgChildItem, [1095](#)
- boundMultipleTags
  - Digikam::TagModificationHelper, [3381](#)
- boundTag
  - Digikam::TagModificationHelper, [3381](#)
- Boxed
  - Digikam::DNotificationPopup, [1249](#)
- branchFromIndex
  - Digikam::RGTagModel, [2984](#)
- brightness
  - Digikam::DRawDecoderSettings, [1378](#)
- buildCollectionTrashCounters
  - Digikam::IOJobsManager, [2157](#)
- bundleProperties
  - Digikam::OnlineVersionChecker, [2795](#)
- Button
  - Digikam::SchemeManager, [3022](#)
- button
  - Digikam::DConfigDlg, [914](#)
  - Digikam::DMultiTabBar, [1200](#)
- buttonBox
  - Digikam::DConfigDlg, [914](#)
- BWGeneric
  - Digikam::BWSepiaContainer, [532](#)
- BWIlfordSFX200
  - Digikam::BWSepiaContainer, [532](#)
- BWKodakHIE
  - Digikam::BWSepiaContainer, [532](#)
- BWNoFilter
  - Digikam::BWSepiaContainer, [532](#)
- BWNoTone
  - Digikam::BWSepiaContainer, [532](#)
- bytesDepth
  - Digikam::DImg, [1079](#)
- cacheByName
  - Digikam::ItemInfoCache, [2309](#)
- cacheCategory
  - Digikam::DCategorizedView::Private, [887](#)

- cachedRectCategory
  - Digikam::DCategorizedView::Private, [887](#)
- cachedRectIndex
  - Digikam::DCategorizedView::Private, [887](#)
- cacheIndex
  - Digikam::DCategorizedView::Private, [887](#)
- cacheKey
  - Digikam::LoadingDescription, [2554](#)
  - Digikam::SharedLoadingTask, [3194](#)
- calcHaar
  - Digikam::Haar::Calculator, [1872](#)
- calculate
  - Digikam::ImageHistogram, [1967](#)
- calculateInfos
  - Digikam::FindDuplicatesAlbumItem, [1721](#)
- calculationAboutToStart
  - Digikam::ImageHistogram, [1967](#)
- calculationStarted
  - Digikam::ImageHistogram, [1967](#)
- callRGBBackend
  - Digikam::BackendGeonamesRG, [435](#)
  - Digikam::BackendGeonamesUSRG, [438](#)
  - Digikam::BackendOsmRG, [457](#)
  - Digikam::RGBBackend, [2978](#)
- cameraAbout
  - Digikam::GPCamera, [1786](#)
  - Digikam::UMSCamera, [3557](#)
- cameraColorMatrix1
  - Digikam::DRawInfo, [1390](#)
- cameraController
  - Digikam::CameraThumbsCtrl, [554](#)
- cameraDriverType
  - Digikam::GPCamera, [1786](#)
  - Digikam::UMSCamera, [3557](#)
- cameraManual
  - Digikam::GPCamera, [1787](#)
  - Digikam::UMSCamera, [3557](#)
- cameraMD5ID
  - Digikam::GPCamera, [1787](#)
  - Digikam::UMSCamera, [3557](#)
- cameraMult
  - Digikam::DRawInfo, [1390](#)
- cameraSummary
  - Digikam::GPCamera, [1787](#)
  - Digikam::UMSCamera, [3558](#)
- camItemInfo
  - Digikam::ImportItemModel, [2068](#)
- camItemInfoActivated
  - Digikam::ImportCategorizedView, [2016](#)
- camItemInfosAdded
  - Digikam::ImportFilterModel, [2054](#)
- camItemInfosCleared
  - Digikam::ImportItemModel, [2068](#)
- camItemInfosSorted
  - Digikam::ImportSortFilterModel, [2111](#)
- canBeCanceled
  - Digikam::ProgressItem, [2870](#)
- canBeWrittenToMetadata
  - Digikam::TagsCache, [3400](#)
- cancel
  - Digikam::ActionJob, [212](#)
  - Digikam::ActionThreadBase, [220](#)
  - Digikam::BatchTool, [466](#)
  - Digikam::DRawDecoder, [1369](#)
  - Digikam::FacePipeline, [1567](#)
  - Digikam::GPCamera, [1787](#)
  - Digikam::LookupAltitudeGeonames, [2587](#)
  - Digikam::MLPipelineFoundation, [2729](#)
  - Digikam::ThumbnailImageCatcher, [3468](#)
  - Digikam::UMSCamera, [3558](#)
- cancelAllAndSuspendCollectionScan
  - Digikam::ScanController, [3011](#)
- cancelCompleteScan
  - Digikam::ScanController, [3011](#)
- cancelFilter
  - Digikam::DImgThreadedFilter, [1117](#)
  - Digikam::GreycstorationFilter, [1862](#)
- cancelled
  - Digikam::CollectionScanner, [625](#)
- cancelRequests
  - Digikam::BackendGeonamesRG, [436](#)
  - Digikam::BackendGeonamesUSRG, [439](#)
  - Digikam::BackendOsmRG, [458](#)
- canExport
  - Digikam::DPluginLoader, [1337](#)
- canImport
  - Digikam::DPluginLoader, [1337](#)
- canRead
  - Digikam::DPluginDImg, [1324](#)
- canWrite
  - Digikam::DPluginDImg, [1324](#)
- canWriteComment
  - Digikam::MetaEngine, [2675](#)
- canWriteExif
  - Digikam::MetaEngine, [2675](#)
- canWriteIptc
  - Digikam::MetaEngine, [2675](#)
- canWriteXmp
  - Digikam::MetaEngine, [2675](#)
- capture
  - Digikam::DKCamera, [1166](#)
  - Digikam::GPCamera, [1787](#)
  - Digikam::UMSCamera, [3558](#)
- CaseInsensitive
  - Digikam::CollectionLocation, [608](#)
- CaseSensitive
  - Digikam::CollectionLocation, [608](#)
- CaseSensitivity
  - Digikam::CollectionLocation, [608](#)
- caseSensitivity
  - Digikam::CollectionLocation, [609](#)
- categories
  - Digikam::DPlugin, [1299](#)
  - Digikam::DPluginBqm, [1308](#)
  - Digikam::DPluginDImg, [1324](#)
  - Digikam::DPluginEditor, [1329](#)

- Digikam::DPluginGeneric, [1333](#)
- Digikam::DPluginRawImport, [1342](#)
- CategorizationMode
  - Digikam::ItemSortSettings, [2414](#)
- CategorizationModeRole
  - Digikam::ImportFilterModel, [2054](#)
  - Digikam::ItemFilterModel, [2243](#)
  - ShowFoto::ShowfotoFilterModel, [3676](#)
- categorize
  - Digikam::ItemHistoryGraph, [2276](#)
- categorizedIndexesIn
  - Digikam::DCategorizedView, [884](#)
- Category
  - Digikam::FilterAction, [1705](#)
- category
  - Digikam::ItemInfo, [2296](#)
- CategoryAlbumIdRole
  - Digikam::ItemFilterModel, [2243](#)
- categoryAt
  - Digikam::DCategorizedView, [885](#)
- CategoryButtonDisplayRole
  - Digikam::SetupCollectionModel, [3171](#)
- CategoryDateRole
  - Digikam::ImportFilterModel, [2054](#)
  - Digikam::ItemFilterModel, [2243](#)
- CategoryDisplayRole
  - Digikam::DCategorizedSortFilterProxyModel, [878](#)
- CategoryFaceRole
  - Digikam::ItemFilterModel, [2243](#)
- CategoryFormatRole
  - Digikam::ImportFilterModel, [2054](#)
  - Digikam::ItemFilterModel, [2243](#)
  - ShowFoto::ShowfotoFilterModel, [3676](#)
- categoryHeight
  - Digikam::DCategoryDrawer, [892](#)
  - Digikam::ImportCategoryDrawer, [2021](#)
  - Digikam::ItemCategoryDrawer, [2191](#)
- categoryIdentifier
  - Digikam::ImportFilterModel, [2054](#)
  - Digikam::ItemFilterModel, [2243](#)
  - ShowFoto::ShowfotoFilterModel, [3676](#)
- categoryRange
  - Digikam::DCategorizedView, [885](#)
- CategorySortRole
  - Digikam::DCategorizedSortFilterProxyModel, [878](#)
- categoryUpperBound
  - Digikam::DCategorizedView::Private, [887](#)
- categoryVisualRect
  - Digikam::DCategorizedView, [885](#)
  - Digikam::DCategorizedView::Private, [888](#)
- center
  - Digikam::CMat, [604](#)
- centerOn
  - Digikam::BackendMarble, [450](#)
- CHANGE\_TIMESTAMPS
  - Digikam::ExifToolProcess, [1527](#)
- changeAlbumFromHistory
  - Digikam::AlbumFolderViewSideBarWidget, [276](#)
- Digikam::DateFolderViewSideBarWidget, [813](#)
- Digikam::FuzzySearchSideBarWidget, [1755](#)
- Digikam::GPSSearchSideBarWidget, [1830](#)
- Digikam::LabelsSideBarWidget, [2496](#)
- Digikam::PeopleSideBarWidget, [2832](#)
- Digikam::SearchSideBarWidget, [3120](#)
- Digikam::SidebarWidget, [3225](#)
- Digikam::TagViewSideBarWidget, [3434](#)
- Digikam::TimelineSideBarWidget, [3505](#)
- changeComment
  - Digikam::ItemComments, [2196](#)
- changed
  - Digikam::DConfigDlgWdgltem, [953](#)
- changeDatabase
  - Digikam::AlbumManager, [287](#)
- changedFlags
  - Digikam::DisjointMetadata, [1139](#)
- changedMap
  - Digikam::DConfigDlgMngr, [921](#)
- changeImageComment
  - Digikam::CoreDB, [683](#)
- changeImageMetadata
  - Digikam::CoreDB, [683](#)
- changeItemInformation
  - Digikam::CoreDB, [684](#)
- changeItemPosition
  - Digikam::CoreDB, [684](#)
- changeRegion
  - Digikam::FaceTagsEditor, [1640](#)
- changeSuggestedName
  - Digikam::FaceTagsEditor, [1640](#)
- changeTag
  - Digikam::FaceTagsEditor, [1640](#)
- changeThumbSize
  - Digikam::DTrashItemModel, [1423](#)
- changeTimestamps
  - Digikam::ExifToolParser, [1522](#)
- ChangeType
  - Digikam::ItemChangeHint, [2192](#)
- changeType
  - Digikam::CollectionManager, [615](#)
- changeVideoMetadata
  - Digikam::CoreDB, [684](#)
- channelToBinary
  - Digikam::ImageCurves, [1955](#)
- checkAllocation
  - Digikam::DImgLoader, [1104](#)
- checkAzimuth
  - Digikam::GeodeticCalculator, [1763](#)
- checkDatabaseDirsAfterFirstRun
  - Digikam::AlbumManager, [287](#)
- checkDatabaseSettings
  - Digikam::DatabaseSettingsWidget, [791](#)
- checkedDisplayTexts
  - Digikam::ChoiceSearchModel, [597](#)
- checkedKeys
  - Digikam::ChoiceSearchModel, [597](#)
- checkedTagsChanged

- Digikam::TagCheckView, [3336](#)
- checkForMigration
  - Digikam::MigrateFromDigikam4Page, [2717](#)
- checkHardWiredLocations
  - Digikam::CollectionManager, [615](#)
- checkIndex
  - Digikam::AbstractWidgetDelegateOverlay, [202](#)
  - Digikam::ActionVersionsOverlay, [225](#)
  - Digikam::AssignNameOverlay, [403](#)
  - Digikam::FaceRejectionOverlay, [1619](#)
  - Digikam::GroupIndicatorOverlay, [1867](#)
  - Digikam::ImportCoordinatesOverlay, [2030](#)
  - Digikam::ImportDownloadOverlay, [2044](#)
  - Digikam::ImportLockOverlay, [2081](#)
  - Digikam::ImportRotateOverlay, [2102](#)
  - Digikam::ItemCoordinatesOverlay, [2201](#)
  - Digikam::ItemFullScreenOverlay, [2265](#)
  - Digikam::ItemRotateOverlay, [2392](#)
  - Digikam::ShowHideVersionsOverlay, [3213](#)
  - ShowFoto::ShowfotoCoordinatesOverlay, [3659](#)
- checkIndexOnEnter
  - Digikam::AbstractWidgetDelegateOverlay, [202](#)
- checkLatitude
  - Digikam::GeodeticCalculator, [1763](#)
- checkLocation
  - Digikam::CollectionManager, [615](#)
- checkLongitude
  - Digikam::GeodeticCalculator, [1764](#)
- checkOrSetWALMode
  - Digikam::BdEngineBackend, [483](#)
- checkOrthodromicDistance
  - Digikam::GeodeticCalculator, [1764](#)
- checkPosition
  - Digikam::ItemQueryPostHooks, [2384](#)
- checkReadyForUse
  - Digikam::CoreDbAccess, [722](#)
  - Digikam::DbEngineAccess, [834](#)
  - Digikam::SimilarityDbAccess, [3234](#)
- checkSelection
  - Digikam::DItemsList, [1159](#)
- checkStateChanged
  - Digikam::AbstractCheckableAlbumModel, [168](#)
- checkToCancelWaitingData
  - Digikam::DRawDecoder, [1369](#)
- childAlbumIds
  - Digikam::Album, [255](#)
- childAlbums
  - Digikam::Album, [255](#)
- childAtRow
  - Digikam::Album, [256](#)
- childCount
  - Digikam::Album, [256](#)
- ChildMatch
  - Digikam::AlbumFilterModel, [269](#)
- ChildToParent
  - Digikam, [126](#)
- ChoiceSearchComboBox
  - Digikam::ChoiceSearchComboBox, [595](#)
- chooserMode
  - Digikam::DColorValueSelector, [905](#)
  - Digikam::DHueSaturationSelector, [1040](#)
- CLARKE\_1866
  - Digikam::Ellipsoid, [1493](#)
- Classifier
  - Digikam::MLPipelineFoundation, [2729](#)
  - Digikam::OpenCVDNNFaceRecognizer, [2799](#)
- classifier
  - Digikam::FacePipelineDetect, [1579](#)
  - Digikam::FacePipelineDetectRecognize, [1583](#)
  - Digikam::FacePipelineEdit, [1587](#)
  - Digikam::FacePipelineRecognize, [1601](#)
  - Digikam::FacePipelineReset, [1605](#)
  - Digikam::FacePipelineRetrain, [1609](#)
- cleanCache
  - Digikam::LoadingCacheInterface, [2550](#)
- CleanScan
  - Digikam::CollectionScanner, [624](#)
- cleanScan
  - Digikam::ItemScanner, [2398](#)
- cleanThumbnailCache
  - Digikam::LoadingCacheInterface, [2550](#)
- cleanUp
  - Digikam::AlbumManager, [287](#)
  - Digikam::DPlugin, [1299](#)
  - Digikam::DPluginLoader, [1337](#)
  - Digikam::LoadingCacheInterface, [2550](#)
- cleanupActions
  - Digikam::DXmlGuiWindow, [1446](#)
- cleanupDatabase
  - Digikam::CoreDbAccess, [722](#)
  - Digikam::SimilarityDbAccess, [3234](#)
- cleanupFilter
  - Digikam::DImgThreadedFilter, [1117](#)
- cleanupTags
  - Digikam::MetadataHub, [2635](#)
- clear
  - Digikam::AdvancedRenameWidget, [246](#)
  - Digikam::Album, [256](#)
  - Digikam::CollectionScannerHintContainerImplementation, [633](#)
  - Digikam::ItemHistoryGraph, [2276](#)
  - Digikam::ItemVisibilityController, [2473](#)
  - Digikam::TrackManager, [3528](#)
  - Digikam::UndoCache, [3564](#)
- ClearAll
  - Digikam::FaceScanSettings, [1624](#)
- clearAll
  - Digikam::DPluginConfView, [1311](#)
- clearAllActions
  - Digikam::DNotificationWidget, [1261](#)
- clearAllTraining
  - Digikam::FacialRecognitionWrapper, [1653](#)
  - Digikam::IdentityProvider, [1951](#)
- clearCaches
  - Digikam::ImportDelegate, [2036](#)
  - Digikam::ItemDelegate, [2212](#)



- ShowFoto::ShowfotoDelegate, 3666
- clearCamlItemInfos
  - Digikam::ImportItemModel, 2069
- clearComments
  - Digikam::MetaEngine, 2675
- clearCurrentAlbums
  - Digikam::AlbumManager, 288
- clearDNNTraining
  - Digikam::FaceDb, 1544
- clearExif
  - Digikam::MetaEngine, 2675
- clearFrom
  - Digikam::UndoCache, 3564
- clearGroup
  - Digikam::ItemInfo, 2296
- clearImageSimilarity
  - Digikam::SimilarityDb, 3228
- clearIptc
  - Digikam::MetaEngine, 2676
- clearItemInfos
  - Digikam::ItemModel, 2349
- clearModelDataCaches
  - Digikam::ImportDelegate, 2036
  - Digikam::ItemDelegate, 2212
  - ShowFoto::ShowfotoDelegate, 3666
- clearParseString
  - Digikam::AdvancedRenameWidget, 246
- clearRects
  - Digikam::ImportDelegate::ImportDelegatePrivate, 2041
  - Digikam::ItemDelegate::ItemDelegatePrivate, 2217
  - Digikam::ItemViewDelegatePrivate, 2456
  - Digikam::ItemViewImportDelegatePrivate, 2466
  - ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate, 3669
  - ShowFoto::ShowfotoItemViewDelegatePrivate, 3711
- clearReferredImages
  - Digikam::DImageHistory, 1068
- clearShowfotoItemInfos
  - ShowFoto::ShowfotoItemModel, 3698
- clearTraining
  - Digikam::FacialRecognitionWrapper, 1653
  - Digikam::IdentityProvider, 1952
  - Digikam::OpenCVDNNFaceRecognizer, 2800
- clearXmp
  - Digikam::MetaEngine, 2676
- clicked
  - Digikam::DNotificationPopup, 1250
  - Digikam::ItemViewCategorized, 2445
- climbTreeAndGetSpacers
  - Digikam::RGTagModel, 2984
- clone
  - Digikam::BatchTool, 466
- close
  - Digikam::BdEngineBackend, 483
  - Digikam::DPopupFrame, 1350
  - Digikam::IccProfile, 1923
  - Digikam::IccTransform, 1943
  - Digikam::PanIconFrame, 2815
- closestItem
  - Digikam::FaceGroup, 1557
- collapseOrExpandClicked
  - Digikam::DCategoryDrawer, 893
- CollectionImageChangeset
  - Digikam::CollectionImageChangeset, 606
- collectionName
  - Digikam::DbKeysCollection, 865
- color
  - Digikam::DFontProperties, 1029
- colorInfoFromOriginal
  - Digikam::Imageface, 1969
- colorKeys
  - Digikam::DRawInfo, 1390
- colorLabel
  - Digikam::DisjointMetadata, 1139
  - Digikam::ItemInfo, 2297
- colorLabelForTag
  - Digikam::TagsCache, 3400
- colorLabelFromTags
  - Digikam::TagsCache, 3400
- colorLabelInterval
  - Digikam::DisjointMetadata, 1139
- colorLabels
  - Digikam::ColorLabelWidget, 647
- colorLabelTags
  - Digikam::TagsCache, 3400
- ColorManagementSettings
  - Digikam::LoadingDescription, 2552
- colorModelToString
  - Digikam::DImg, 1079
- ColorRectPixmap
  - Digikam::LabelsTreeView, 2499
- ColorSet
  - Digikam::SchemeManager, 3022
- ColorTool
  - Digikam::BatchTool, 465
- colorValue
  - Digikam::DColorValueSelector, 905
  - Digikam::DHueSaturationSelector, 1040
- cols
  - Digikam::Mat, 2630
- columnAffectedByChangeset
  - Digikam::TableViewColumn, 3286
  - Digikam::TableViewColumns::ColumnDigikamProperties, 3297
- columnCount
  - Digikam::DConfigDlgWdgModel, 960
- columnHeader
  - Digikam::AbstractAlbumModel, 144
  - Digikam::AbstractSpecificAlbumModel, 200
- command
  - Digikam::ExifToolProcess, 1528
- comment
  - Digikam::DConfigDlgTitle, 931
  - Digikam::ItemInfo, 2297

- commentForLanguage
  - Digikam::ItemComments, [2196](#)
- comments
  - Digikam::DisjointMetadata, [1139](#)
- commit
  - Digikam::ItemScanner, [2398](#)
- commitTransaction
  - Digikam::BdEngineBackend, [483](#)
- compare
  - Digikam::CamItemSortSettings, [559](#)
  - Digikam::ItemSortSettings, [2415](#)
  - Digikam::TableViewColumn, [3286](#)
  - Digikam::TableViewColumns::ColumnAudioVideoProperties, [3293](#)
  - Digikam::TableViewColumns::ColumnDigikamProperties, [3297](#)
  - Digikam::TableViewColumns::ColumnFileProperties, [3302](#)
  - Digikam::TableViewColumns::ColumnGeoProperties, [3307](#)
  - Digikam::TableViewColumns::ColumnItemProperties, [3311](#)
  - Digikam::TableViewColumns::ColumnPhotoProperties, [3316](#)
  - ShowFoto::ShowfotoItemSortSettings, [3703](#)
- compareByOrder
  - Digikam::AlbumFilterModel, [269](#)
  - Digikam::CamItemSortSettings, [559](#)
  - Digikam::ItemSortSettings, [2415](#)
  - ShowFoto::ShowfotoItemSortSettings, [3703](#)
- compareCategories
  - Digikam::CamItemSortSettings, [559](#)
  - Digikam::DCategorizedSortFilterProxyModel, [879](#)
  - Digikam::ImportFilterModel, [2054](#)
  - Digikam::ItemFilterModel, [2244](#)
  - Digikam::ItemSortSettings, [2415](#)
  - ShowFoto::ShowfotoFilterModel, [3676](#)
  - ShowFoto::ShowfotoItemSortSettings, [3703](#)
- compareInfosCategories
  - Digikam::ImportFilterModel, [2055](#)
  - Digikam::ItemAlbumFilterModel, [2172](#)
  - Digikam::ItemFilterModel, [2244](#), [2245](#)
  - ShowFoto::ShowfotoFilterModel, [3676](#)
- compareValue
  - Digikam::AlbumFilterModel, [269](#)
  - Digikam::CamItemSortSettings, [560](#)
  - Digikam::ItemSortSettings, [2415](#)
  - ShowFoto::ShowfotoItemSortSettings, [3703](#)
- Complementary
  - Digikam::SchemeManager, [3023](#)
- CompleteCollectionScan
  - Digikam::NewItemFinder, [2762](#)
- completeCollectionScan
  - Digikam::ScanController, [3011](#)
- completeCollectionScanInBackground
  - Digikam::ScanController, [3012](#)
- completed
  - Digikam::SharedLoadingTask, [3194](#)
- completelyApplied
  - Digikam::FilterActionFilter, [1711](#)
- completeScan
  - Digikam::CollectionScanner, [625](#)
- ComplexFilter
  - Digikam::FilterAction, [1706](#)
- compose
  - Digikam::DColorComposer, [901](#)
- CompositingOperation
  - Digikam::DColorComposer, [901](#)
- computeDestinationPoint
  - Digikam::GeodeticCalculator, [1764](#)
- computeDirection
  - Digikam::GeodeticCalculator, [1764](#)
- configElement
  - Digikam::BdEngineBackend, [483](#)
- configFullScreenHideToolBarsEntry
  - Digikam::DXmlGuiWindow, [1446](#)
- configGroupName
  - Digikam::DPluginLoader, [1337](#)
- confirm
  - Digikam::FacePipeline, [1567](#)
- Confirmed
  - Digikam::FacePipelineFaceTagsIface, [1593](#)
- confirmedEntry
  - Digikam::FaceTagsEditor, [1640](#)
- confirmFaces
  - Digikam::DigikamItemView, [1063](#)
- confirmName
  - Digikam::FaceTagsEditor, [1640](#)
- conflictRule
  - Digikam::VidSlideSettings, [3597](#)
- connectAndSchedule
  - Digikam::WorkerObject, [3618](#)
- connectFinishAndErrorSignals
  - Digikam::DBJobsThread, [863](#)
- ConnectionError
  - Digikam::BdEngineBackend, [482](#)
- connectionError
  - Digikam::DbEngineErrorHandler, [840](#)
- connectionErrorAbortQueries
  - Digikam::BdEngineBackendPrivate, [492](#)
- connectionErrorContinueQueries
  - Digikam::BdEngineBackendPrivate, [492](#)
  - Digikam::DbEngineErrorAnswer, [839](#)
- connectionErrorHandling
  - Digikam::BdEngineBackend, [483](#)
- connectToSignalFileChanged
  - Digikam::LoadingCacheInterface, [2550](#)
- consultUserForError
  - Digikam::DbEngineErrorHandler, [840](#)
- contactInfo
  - Digikam::ItemCopyright, [2205](#)
- contains
  - Digikam::SqueezedComboBox, [3254](#)
- containsItem
  - Digikam::ListItem, [2540](#)
- containsPublicTags

- Digikam::TagsCache, [3401](#)
- contentsMouseEvent
  - Digikam::DMultiTabBarFrame, [1206](#)
- contentsRect
  - Digikam::DPointSelect, [1346](#)
  - Digikam::DSelector, [1400](#)
- contentsSize
  - Digikam::DCategorizedView::Private, [888](#)
- contextMenuEvent
  - Digikam::TagFolderView, [3356](#)
  - Digikam::TagMngrTreeView, [3372](#)
- ContextMenuHelper
  - Digikam::ContextMenuHelper, [661](#)
- contextMenuIcon
  - Digikam::AbstractAlbumTreeView, [152](#)
- contextMenuTitle
  - Digikam::AbstractAlbumTreeView, [152](#)
  - Digikam::EditableSearchTreeView, [1464](#)
  - Digikam::TagFolderView, [3356](#)
- continueQuery
  - Digikam::DImgLoaderObserver, [1106](#)
  - Digikam::LoadingTask, [2560](#)
  - Digikam::SavingTask, [3007](#)
  - Digikam::SimpleCollectionScannerObserver, [3240](#)
- contrast
  - Digikam::SchemeManager, [3026](#)
- contrastF
  - Digikam::SchemeManager, [3026](#)
- convertCommentValue
  - Digikam::MetaEngine::Private, [2705](#)
- convertDegreeAngleToDouble
  - Digikam::MetaEngine, [2676](#)
- convertDepth
  - Digikam::DImg, [1079](#)
- ConvertError
  - Digikam::DNGWriter, [1212](#)
- ConvertForDisplay
  - Digikam::LoadingDescription, [2552](#)
- ConvertForOutput
  - Digikam::LoadingDescription, [2552](#)
- convertFromGPSCoordinateString
  - Digikam::MetaEngine, [2676](#)
- convertOriginalColorDepth
  - Digikam::Imagelface, [1969](#)
- convertRatio
  - Digikam::NamespaceEntry, [2755](#)
- convertToGPSCoordinateString
  - Digikam::MetaEngine, [2676](#), [2677](#)
- ConvertTool
  - Digikam::BatchTool, [465](#)
- convertToPixmap
  - Digikam::EditorCore, [1467](#)
  - Digikam::Imagelface, [1969](#)
- convertToRational
  - Digikam::MetaEngine, [2677](#)
- convertToRationalSmallDenominator
  - Digikam::MetaEngine, [2677](#)
- convertToSixteenBit
  - Digikam::DColor, [898](#)
  - Digikam::DImg, [1079](#)
- convertToUserPresentableNumbers
  - Digikam::MetaEngine, [2677](#), [2678](#)
- convertZoomToBackendZoom
  - Digikam::MapWidget, [2623](#)
- coordinates
  - Digikam::RGInfo, [2979](#)
- coordinatesToClipboard
  - Digikam, [127](#)
- Copied
  - Digikam::CollectionImageChangeset, [605](#)
- copiedFrom
  - Digikam::ItemScanner, [2398](#)
- copy
  - Digikam::DImg, [1079](#), [1080](#)
  - Digikam::DIO, [1134](#)
- COPY\_ALL
  - Digikam::ExifToolProcess, [1527](#)
- COPY\_EXIF
  - Digikam::ExifToolProcess, [1527](#)
- COPY\_ICC
  - Digikam::ExifToolProcess, [1527](#)
- COPY\_IPTC
  - Digikam::ExifToolProcess, [1527](#)
- COPY\_MAKERNOTES
  - Digikam::ExifToolProcess, [1527](#)
- COPY\_NONE
  - Digikam::ExifToolProcess, [1527](#)
- COPY\_TAGS
  - Digikam::ExifToolProcess, [1527](#)
- COPY\_XMP
  - Digikam::ExifToolProcess, [1527](#)
- copyAlbumProperties
  - Digikam::CoreDB, [684](#)
- copyAttributes
  - Digikam::FileActionMngrDatabaseWorker, [1673](#)
- copyFile
  - Digikam::DFileOperations, [1020](#)
- copyFileProperties
  - Digikam::CollectionScanner, [625](#)
- copyFiles
  - Digikam::DFileOperations, [1020](#)
- copyFolderRecursively
  - Digikam::DFileOperations, [1021](#)
- copyImageAttributes
  - Digikam::CoreDB, [684](#)
- copyImageData
  - Digikam::DImg, [1080](#)
- copyImageProperties
  - Digikam::CoreDB, [685](#)
- copyImageTags
  - Digikam::CoreDB, [685](#)
- copyItem
  - Digikam::CoreDB, [685](#)
  - Digikam::ItemInfo, [2297](#)
- copyMetaData
  - Digikam::DImg, [1080](#)

- copyModificationTime
  - Digikam::DFileOperations, [1021](#)
- copyOrMove
  - Digikam::IOJobsThread, [2161](#)
- copyProperties
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1841](#)
- copyQImage
  - Digikam::DImg, [1080](#)
- copyQuery
  - Digikam::BdEngineBackend, [483](#)
- copyrightNotice
  - Digikam::ItemCopyright, [2205](#)
- copySearch
  - Digikam::NormalSearchTreeView, [2780](#)
- copySimilarityAttributes
  - Digikam::SimilarityDb, [3228](#)
- copyTags
  - Digikam::ExifToolParser, [1522](#)
- CopyTagsSource
  - Digikam::ExifToolProcess, [1527](#)
- CoreDB
  - Digikam::CoreDB, [678](#)
- CoreDbAccess
  - Digikam::CoreDbAccess, [722](#)
- CoreDbAccessUnlock
  - Digikam::CoreDbAccessUnlock, [724](#)
- CoreDbNameFilter
  - Digikam::CoreDbNameFilter, [733](#)
- CoreDbOperationGroup
  - Digikam::CoreDbOperationGroup, [734](#)
- CoreDbTransaction
  - Digikam::CoreDbTransaction, [736](#)
- CoreDbUrl
  - Digikam::CoreDbUrl, [739](#)
- CoreDbWatchAdaptor, [139](#)
- cosineDistance
  - Digikam::DNNFaceExtractorBase, [1228](#)
- count
  - Digikam::DPlugin, [1300](#)
  - Digikam::DPluginBqm, [1308](#)
  - Digikam::DPluginConfView, [1311](#)
  - Digikam::DPluginDImg, [1324](#)
  - Digikam::DPluginEditor, [1329](#)
  - Digikam::DPluginGeneric, [1333](#)
  - Digikam::DPluginRawImport, [1342](#)
- countryCodeMap
  - Digikam::DMetadata, [1184](#)
- countryCodeMap2
  - Digikam::DMetadata, [1184](#)
- CR\_basis
  - Digikam, [135](#)
- CREATE\_NEW\_GROUPS
  - Digikam::ExifToolProcess, [1528](#)
- CREATE\_NEW\_TAGS
  - Digikam::ExifToolProcess, [1528](#)
- createAnimation
  - Digikam::ItemVisibilityController, [2473](#)
- createApplicationPalette
  - Digikam::SchemeManager, [3026](#)
- createButton
  - Digikam::ActionVersionsOverlay, [225](#)
  - Digikam::FaceRejectionOverlay, [1619](#)
  - Digikam::HoverButtonDelegateOverlay, [1906](#)
  - Digikam::ImportRotateOverlay, [2102](#)
  - Digikam::ItemFullScreenOverlay, [2265](#)
  - Digikam::ItemRotateOverlay, [2392](#)
  - Digikam::ItemSelectionOverlay, [2406](#)
  - Digikam::ShowHideVersionsOverlay, [3213](#)
- CreateDefaultDelegate
  - Digikam::AbstractAlbumTreeView, [150](#)
- CreateDefaultFilterModel
  - Digikam::AbstractAlbumTreeView, [150](#)
- CreateDefaultModel
  - Digikam::AbstractAlbumTreeView, [150](#)
- createEllipsoid
  - Digikam::Ellipsoid, [1493](#)
- createExifUserStringFromValue
  - Digikam::MetaEngine, [2678](#)
- createField
  - Digikam::SearchField, [3038](#)
- createFilter
  - Digikam::BasicDImgFilterGenerator< T >, [462](#)
  - Digikam::DImgFilterGenerator, [1098](#)
  - Digikam::DImgFilterManager, [1100](#)
- createFilterModel
  - Digikam::ActionItemModel, [210](#)
- createFlattenedSphere
  - Digikam::Ellipsoid, [1493](#)
- createFullScreenAction
  - Digikam::DXmlGuiWindow, [1446](#)
- createFuzzySearchFromDropped
  - Digikam::SearchModificationHelper, [3115](#)
- createFuzzySearchFromImage
  - Digikam::SearchModificationHelper, [3116](#)
- createFuzzySearchFromSketch
  - Digikam::SearchModificationHelper, [3116](#)
- createHelpActions
  - Digikam::DXmlGuiWindow, [1446](#)
- createHintContainer
  - Digikam::CollectionScanner, [625](#)
- createHistoryImageId
  - Digikam::DImg, [1080](#)
- createImageUniqueld
  - Digikam::DImg, [1080](#)
- createItemWidgets
  - Digikam::DWItemDelegate, [1436](#)
  - Digikam::SetupCollectionDelegate, [3167](#)
- createMimeData
  - Digikam::AbstractItemDragDropHandler, [188](#)
  - Digikam::AlbumDragDropHandler, [265](#)
  - Digikam::AlbumModelDragDropHandler, [314](#)
  - Digikam::GPSItemListDragDropHandler, [1812](#)
  - Digikam::ImportDragDropHandler, [2047](#)
  - Digikam::ItemDragDropHandler, [2227](#)
  - Digikam::MapDragDropHandler, [2614](#)

- Digikam::TagDragDropHandler, [3341](#)
- ShowFoto::ShowfotoDragDropHandler, [3671](#)
- CreateNewImageHistoryUUID
  - Digikam::DImg, [1076](#)
- CreateNewMetadataPreview
  - Digikam::DImg, [1076](#)
- createNode
  - Digikam::KDNodeBase, [2481](#)
  - Digikam::KDNodeOpenFace, [2484](#)
  - Digikam::KDNodeSFace, [2486](#)
  - Digikam::KDTreeBase, [2488](#)
- createPALbum
  - Digikam::AlbumManager, [288](#)
- createProgressItem
  - Digikam::PrivateProgressItemCreator, [2865](#)
  - Digikam::ProgressManager, [2877](#), [2878](#)
- createSAlbum
  - Digikam::AlbumManager, [289](#)
- createSearchGroup
  - Digikam::AbstractSearchGroupContainer, [196](#)
  - Digikam::SearchGroup, [3104](#)
  - Digikam::SearchView, [3139](#)
- createSettingsActions
  - Digikam::DXmlGuiWindow, [1446](#)
- createSidebarActions
  - Digikam::DXmlGuiWindow, [1446](#)
- createTag
  - Digikam::TagsCache, [3401](#)
- createTAlbum
  - Digikam::AlbumManager, [290](#)
  - Digikam::TagEditDlg, [3343](#)
- createThreadedFilter
  - Digikam::DImgBuiltinFilter, [1091](#)
- createView
  - Digikam::DConfigDlgView, [939](#)
- createWidget
  - Digikam::AbstractWidgetDelegateOverlay, [202](#)
  - Digikam::AssignNameOverlay, [403](#)
  - Digikam::GroupIndicatorOverlay, [1867](#)
  - Digikam::HoverButtonDelegateOverlay, [1906](#)
  - Digikam::ImportCoordinatesOverlay, [2030](#)
  - Digikam::ImportDownloadOverlay, [2044](#)
  - Digikam::ImportLockOverlay, [2081](#)
  - Digikam::ImportRatingOverlay, [2096](#)
  - Digikam::ItemCoordinatesOverlay, [2201](#)
  - Digikam::ItemRatingOverlay, [2387](#)
  - Digikam::TagsLineEditOverlay, [3414](#)
  - ShowFoto::ShowfotoCoordinatesOverlay, [3659](#)
- creationDateFromFilesystem
  - Digikam::ItemScanner, [2398](#)
- CreationDateRole
  - Digikam::ItemModel, [2348](#)
- creator
  - Digikam::ItemCopyright, [2205](#)
- creatorJobTitle
  - Digikam::ItemCopyright, [2205](#)
- Crop
  - Digikam::DImgBuiltinFilter, [1090](#)
- crop
  - Digikam::DImg, [1080](#)
  - Digikam::Imagelface, [1969](#)
- Current
  - Digikam::HistoryImageId, [1895](#)
- currentActiveItem
  - Digikam::DMetaInfiface, [1194](#)
- currentAlbumChanged
  - Digikam::AbstractAlbumTreeView, [152](#)
- currentAlbumItems
  - Digikam::DBInfiface, [855](#)
  - Digikam::DMetaInfiface, [1194](#)
- currentAlbums
  - Digikam::AlbumManager, [290](#)
  - Digikam::AlbumManager::Private, [306](#)
- currentCamItemInfo
  - Digikam::MapViewWidgetView, [2629](#)
- currentCategorizationSortOrder
  - ShowFoto::ShowfotoItemSortSettings, [3704](#)
- currentFilePath
  - ShowFoto::ShowfotoFolderViewModel, [3685](#)
- currentHistogram
  - Digikam::HistogramWidget, [1893](#)
- currentImage
  - Digikam::Canvas, [565](#)
- currentImageFileFormat
  - Digikam::Canvas, [565](#)
- currentImageFilePath
  - Digikam::Canvas, [565](#)
- currentItemInfo
  - Digikam::MapViewWidgetView, [2629](#)
- currentPage
  - Digikam::DConfigDlg, [914](#)
  - Digikam::DConfigDlgView, [939](#)
  - Digikam::DConfigDlgWdg, [948](#)
- currentPageChanged
  - Digikam::DConfigDlg, [914](#)
  - Digikam::DConfigDlgView, [939](#)
  - Digikam::DConfigDlgWdg, [948](#)
- currentPALbum
  - Digikam::AlbumManager, [290](#)
- currentProfile
  - Digikam::IccProfilesComboBox, [1928](#)
- currentReferenceImage
  - Digikam::ItemInfo, [2297](#)
- currentSeed
  - Digikam::RandomNumberGenerator, [2909](#)
- currentSelectedItems
  - Digikam::DBInfiface, [855](#)
  - Digikam::DInfoInterface, [1128](#)
  - Digikam::DMetaInfiface, [1194](#)
- currentTaggingAction
  - Digikam::AddTagsComboBox, [231](#)
- currentTAlbums
  - Digikam::AlbumManager, [290](#)
- currentValidity
  - Digikam::BdEngineBackendPrivate, [493](#)
- CURVE\_FREE

- Digikam::ImageCurves, [1955](#)
- CURVE\_SMOOTH
  - Digikam::ImageCurves, [1955](#)
- curvesAdjust
  - Digikam::DRawDecoding, [1386](#)
- CurvesContainer
  - Digikam::CurvesContainer, [749](#)
- curvesType
  - Digikam::CurvesContainer, [750](#)
- CurveType
  - Digikam::ImageCurves, [1955](#)
- customIdentifier
  - Digikam::ThumbnailInfo, [3471](#)
- customizedFullScreenMode
  - Digikam::DXmlGuiWindow, [1447](#)
- CustomSettings
  - Digikam::ImageQualityConfSelector, [1979](#)
- CustomStepsDoubleSpinBox
  - Digikam::CustomStepsDoubleSpinBox, [760](#)
- CustomStepsIntSpinBox
  - Digikam::CustomStepsIntSpinBox, [763](#)
- CustomTool
  - Digikam::BatchTool, [465](#)
- customWhiteBalance
  - Digikam::DRawDecoderSettings, [1378](#)
- CVD1
  - Digikam::VidSlideSettings, [3595](#)
- CVD2
  - Digikam::VidSlideSettings, [3595](#)
- DarkShade
  - Digikam::SchemeManager, [3024](#)
- data
  - Digikam::CMat, [604](#)
  - Digikam::IccProfile, [1923](#)
  - Digikam::ImportThumbnailModel, [2132](#)
  - Digikam::ItemFilterModel, [2245](#)
  - Digikam::ItemThumbnailModel, [2439](#)
  - Digikam::Mat, [2630](#)
  - Digikam::MetaEnginePreviews, [2709](#)
  - Digikam::TableViewColumn, [3287](#)
  - Digikam::TableViewColumns::ColumnAudioVideoProperties, [3293](#)
  - Digikam::TableViewColumns::ColumnDigikamProperties, [3297](#)
  - Digikam::TableViewColumns::ColumnFileProperties, [3302](#)
  - Digikam::TableViewColumns::ColumnGeoProperties, [3307](#)
  - Digikam::TableViewColumns::ColumnItemProperties, [3311](#)
  - Digikam::TableViewColumns::ColumnPhotoProperties, [3316](#)
  - Digikam::TableViewColumns::ColumnThumbnail, [3320](#)
  - Digikam::TagMngrListModel, [3364](#)
  - ShowFoto::ShowfotoThumbnailModel, [3763](#)
- databaseChanged
  - Digikam::CoreDbWatch, [745](#)
- databaseEqual
  - Digikam::AlbumManager, [291](#)
- databaseFaces
  - Digikam::FaceTagsEditor, [1641](#)
- databaseForThread
  - Digikam::BdEngineBackendPrivate, [492](#)
- DatabaseImageMetadataFieldsToMetadataInfoField
  - Digikam, [127](#)
- databaseInitialization
  - Digikam::ScanController, [3012](#)
- databaseInitialScanDone
  - Digikam::CollectionScanner, [625](#)
- DatabaseServerErrorEnum
  - Digikam::DatabaseServerError, [788](#)
- databaseType
  - Digikam::BdEngineBackend, [483](#)
- databaseUrl
  - Digikam::Album, [256](#)
  - Digikam::DAlbum, [772](#)
  - Digikam::PAlbum, [2813](#)
  - Digikam::SAlbum, [3004](#)
  - Digikam::TAlbum, [3438](#)
- databaseUuid
  - Digikam::CoreDB, [685](#)
- DatabaseVideoMetadataFieldsToMetadataInfoField
  - Digikam, [128](#)
- databaseWatch
  - Digikam::CoreDbAccess, [723](#)
- dataSize
  - Digikam::MetaEnginePreviews, [2709](#)
- DATE
  - Digikam::Album, [255](#)
- date
  - Digikam::DDateEdit, [968](#)
  - Digikam::DDatePicker, [972](#)
  - Digikam::DDateTable, [981](#)
  - Digikam::DDateTable::Private, [985](#)
- DateAlbumModel
  - Digikam::DateAlbumModel, [806](#)
- dateChanged
  - Digikam::DDateEdit, [968](#)
  - Digikam::DDatePicker, [972](#)
  - Digikam::DDatePickerPopup, [978](#)
  - Digikam::DDateTable, [981](#)
- dateEntered
  - Digikam::DDatePicker, [973](#)
- dateFromPos
  - Digikam::DDateTable, [981](#)
- datePicker
  - Digikam::DDatePickerPopup, [978](#)
- DatePickerYearSelector
  - Digikam::DatePickerYearSelector, [821](#)
- DateRange
  - Digikam, [124](#)
- DateRangeList
  - Digikam, [124](#)
- dateSelected
  - Digikam::DDatePicker, [973](#)

- datesListing
  - Digikam::DatesDBJobsThread, [824](#)
- dateTable
  - Digikam::DDatePicker, [973](#)
- dateTime
  - Digikam::DDateTimeEdit, [988](#)
  - Digikam::DisjointMetadata, [1140](#)
  - Digikam::DRawInfo, [1391](#)
  - Digikam::ItemInfo, [2297](#)
- dateTimeChanged
  - Digikam::DDateTimeEdit, [989](#)
  - Digikam::DisjointMetadata, [1140](#)
- dateTimeInterval
  - Digikam::DisjointMetadata, [1140](#)
- dateTimeStatus
  - Digikam::DisjointMetadata, [1140](#)
- dateUrl
  - Digikam::CoreDbUrl, [740](#)
- daylightMult
  - Digikam::DRawInfo, [1391](#)
- DB
  - Digikam::OpenCVDNNFaceRecognizer, [2800](#)
- db
  - Digikam::CoreDbAccess, [723](#)
- DbEngineParameters
  - Digikam::DbEngineParameters, [844](#)
- DbKeysCollection
  - Digikam::DbKeysCollection, [865](#)
- DCategoryDrawer
  - Digikam::DCategoryDrawer, [892](#)
- dcbEnhanceFl
  - Digikam::DRawDecoderSettings, [1378](#)
- dcbIterations
  - Digikam::DRawDecoderSettings, [1378](#)
- DColor
  - Digikam::DColor, [898](#)
- DConfigDlg
  - Digikam::DConfigDlg, [911](#)
- DConfigDlgMgr
  - Digikam::DConfigDlgMgr, [920](#)
- DConfigDlgModel
  - Digikam::DConfigDlgModel, [926](#)
- DConfigDlgTitle
  - Digikam::DConfigDlgTitle, [931](#)
- DConfigDlgView
  - Digikam::DConfigDlgView, [939](#)
- DConfigDlgWdg
  - Digikam::DConfigDlgWdg, [946](#)
- DConfigDlgWdgItem
  - Digikam::DConfigDlgWdgItem, [952](#), [953](#)
- DConfigDlgWdgModel
  - Digikam::DConfigDlgWdgModel, [958](#)
- DDatePicker
  - Digikam::DDatePicker, [972](#)
- DDatePickerPopup
  - Digikam::DDatePickerPopup, [977](#)
- DDateTimeEdit
  - Digikam::DDateTimeEdit, [988](#)
- deactivate
  - Digikam::WorkerObject, [3618](#)
- DeactivatingMode
  - Digikam::WorkerObject, [3617](#)
- deadPixelMap
  - Digikam::DRawDecoderSettings, [1378](#)
- debugExtractedRAWData
  - Digikam::DNGWriter::Private, [1214](#)
- decodeGPSCoordinate
  - Digikam::MetaEngine::Private, [2705](#)
- decodeHalfRAWImage
  - Digikam::DRawDecoder, [1369](#)
- decodeRAWImage
  - Digikam::DRawDecoder, [1369](#)
- DecodingQuality
  - Digikam::DRawDecoderSettings, [1375](#)
- decodingSettingsToXml
  - Digikam::DRawDecoding, [1386](#)
- DecorateTool
  - Digikam::BatchTool, [465](#)
- decoration
  - Digikam::SchemeManager, [3026](#)
- DecorationRole
  - Digikam::SchemeManager, [3023](#)
- decorationRoleData
  - Digikam::AbstractAlbumModel, [144](#)
  - Digikam::AlbumModel, [312](#)
  - Digikam::DateAlbumModel, [807](#)
  - Digikam::TagModel, [3378](#)
- defaultComment
  - Digikam::ItemComments, [2196](#)
- defaultFieldOperator
  - Digikam::SearchXmlReader, [3148](#)
- defaultLocation
  - Digikam::DNotificationPopup, [1250](#)
- defaultMysqlAdminCmd
  - Digikam::DbEngineParameters, [844](#)
- defaultMysqlInitCmd
  - Digikam::DbEngineParameters, [844](#)
- defaultMysqlServerCmd
  - Digikam::DbEngineParameters, [844](#)
- defaultMysqlUpgradeCmd
  - Digikam::DbEngineParameters, [844](#)
- DefaultOrder
  - Digikam::CamItemSortSettings, [559](#)
  - Digikam::ItemSortSettings, [2414](#)
  - ShowFoto::ShowfotoItemSortSettings, [3702](#)
- defaultParameters
  - Digikam::DbEngineParameters, [844](#)
- defaultProfiles
  - Digikam::lccProfile, [1923](#)
- defaultSearchPaths
  - Digikam::lccProfile, [1924](#)
- defaultSettings
  - Digikam::BatchTool, [466](#)
- defaultThread
  - Digikam::ThumbnailLoadThread, [3480](#)
- defaultUploadUrl

- Digikam::DBInfolface, [855](#)
- Digikam::DInfoInterface, [1128](#)
- Digikam::DMetalInfolface, [1194](#)
- defineShortcut
  - Digikam, [128](#)
- deleteAlbum
  - Digikam::CoreDB, [685](#)
- deleteAlbumRoot
  - Digikam::CoreDB, [686](#)
- deleteAllSpacersOrNewTags
  - Digikam::RGTagModel, [2984](#)
- Deleted
  - Digikam::CollectionImageChangeset, [605](#)
- DeleteDecorationRole
  - Digikam::SetupCollectionModel, [3171](#)
- deleteDirRecursively
  - Digikam::DTrash, [1420](#)
- deleteFiles
  - Digikam::IOJobsThread, [2161](#)
- deleteFilterInstance
  - Digikam::EditorToolThreaded, [1481](#)
- deleteIdentities
  - Digikam::FacialRecognitionWrapper, [1653](#)
  - Digikam::IdentityProvider, [1952](#)
- deleteIdentity
  - Digikam::FacialRecognitionWrapper, [1653](#)
  - Digikam::IdentityProvider, [1952](#)
- deleteImage
  - Digikam::DBInfolface, [855](#)
  - Digikam::DInfoInterface, [1128](#)
  - Digikam::DMetalInfolface, [1194](#)
  - Digikam::DTrash, [1421](#)
- deleteItem
  - Digikam::CoreDB, [686](#)
  - Digikam::GPCamera, [1787](#)
  - Digikam::UMSCamera, [3558](#)
- deleteObsoleteItem
  - Digikam::CoreDB, [687](#)
- deleteRemovedItems
  - Digikam::CoreDB, [687](#)
- deleteSAlbum
  - Digikam::AlbumManager, [291](#)
- deleteSearch
  - Digikam::CoreDB, [687](#)
- deleteSearches
  - Digikam::CoreDB, [687](#)
- deleteSettingsWidget
  - Digikam::BatchTool, [466](#)
- deleteStaleAlbums
  - Digikam::CoreDB, [687](#)
- deleteTab
  - Digikam::Sidebar, [3218](#)
- deleteTag
  - Digikam::CoreDB, [687](#)
  - Digikam::RGTagModel, [2985](#)
- deleteTAlbum
  - Digikam::AlbumManager, [291](#)
- deleteThumbnail
  - Digikam::ThumbnailLoadThread, [3480](#)
- deleteThumbnailsFromDisk
  - Digikam::ThumbnailCreator, [3462](#)
- depth\_first\_search\_sorted
  - Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch, [1846](#)
- description
  - Digikam::DRawInfo, [1391](#)
  - Digikam::FilterAction, [1706](#)
  - Digikam::IccProfile, [1924](#)
  - Digikam::Token, [3511](#)
- deselected
  - Digikam::ImportCategorizedView, [2016](#)
  - ShowFoto::ShowfotoCategorizedView, [3654](#)
- destinationGeographicPoint
  - Digikam::GeodeticCalculator, [1765](#)
- detach
  - Digikam::DImg, [1081](#)
- detAdjustmentByClockPhotoUrl
  - Digikam::TimeAdjustSettings, [3503](#)
- detect
  - Digikam::AestheticDetector, [251](#)
  - Digikam::BlurDetector, [504](#)
  - Digikam::CompressionDetector, [654](#)
  - Digikam::ExposureDetector, [1539](#)
  - Digikam::NoiseDetector, [2769](#)
- detectAccuracy
  - Digikam::FaceScanSettings, [1625](#)
- DetectAndRecognize
  - Digikam::FaceScanSettings, [1625](#)
- detectedFormat
  - Digikam::DImg, [1081](#)
- detectEncodingAndDecode
  - Digikam::MetaEngine::Private, [2705](#)
- detectFaces
  - Digikam::DNNFaceDetectorSSD, [1222](#)
  - Digikam::DNNFaceDetectorYOLO, [1224](#)
  - Digikam::DNNFaceDetectorYuNet, [1226](#)
  - Digikam::FaceDetector, [1553](#)
  - Digikam::OpenCVDNNFaceDetector, [2798](#)
- detectLanguage
  - Digikam::DOnlineTranslator, [1273](#)
- detectLanguageAlt
  - Digikam::MetaEngine, [2678](#)
- detectObjects
  - Digikam::DNNBaseDetectorModel, [1217](#)
- DetectorModel
  - Digikam, [124](#)
- DetectorNNModel
  - Digikam, [125](#)
- DFontProperties
  - Digikam::DFontProperties, [1028](#)
- DHueSaturationSelector
  - Digikam::DHueSaturationSelector, [1040](#)
- Digikam, [89](#)
  - accessCol, [135](#)
  - accessRow, [135](#)
  - ActionJobCollection, [123](#)



- adjustedEnvironmentForApplImage, 127
- asDateTimeLocal, 127
- asDateTimeUTC, 127
- BatchSetList, 123
- BatchToolSettings, 124
- BatchToolsList, 124
- ChildToParent, 126
- coordinatesToClipboard, 127
- CR\_basis, 135
- DatabaseImageMetadataFieldsToMetadataInfoField, 127
- DatabaseVideoMetadataFieldsToMetadataInfoField, 128
- DateRange, 124
- DateRangeList, 124
- defineShortcut, 128
- DetectorModel, 124
- DetectorNNModel, 125
- DImgLoaderPrms, 124
- DItemsListIsLessThanHandler, 124
- DNNDetectorSSD, 125
- DNNDetectorYOLOv3, 125
- DNNDetectorYuNet, 125
- DNotificationWrapper, 128
- faceenum2size, 135
- FS\_ALBUMGUI, 125
- FS\_EDITOR, 125
- FS\_IMPORTUI, 125
- FS\_LIGHTTABLE, 125
- FS\_NONE, 125
- FS\_SIDEBARS, 125
- FS\_STATUSBAR, 125
- FS\_THUMBBAR, 125
- FS\_TOOLBARS, 125
- FullImageHistogram, 126
- FullScreenOptions, 125
- GeoGroupStateEnum, 125
- GeofaceHelperParseLatLonString, 128
- GeofaceMinMarkerGroupingRadius, 136
- GetComponentValue, 128
- HistogramRenderingType, 125
- HistogramScale, 126
- HS\_None, 126
- HudSide, 126
- image2Mat, 129
- image2Mat\_shared, 129
- ImageSelectionHistogram, 126
- installQtTranslationFiles, 130
- isReadableImageFile, 130
- isRunningInApplImageBundle, 130
- isRunningOnNativeKDE, 130
- layoutMargin, 130
- layoutSpacing, 130
- LinScaleHistogram, 126
- loadEcmQtTranslationFiles, 130
- loadStdQtTranslationFiles, 131
- LogScaleHistogram, 126
- macOSBundlePrefix, 131
- mat2Image, 131
- mat2Image\_shared, 131
- MeaningOfDirection, 126
- openOnlineDocumentation, 131
- OperationType, 126
- operator<<, 131
- operator|, 132
- ParentToChild, 126
- qHash, 132
- QPointSquareDistance, 132
- QueuePoolItemsList, 124
- RESNET50, 127
- s\_inlineTranslateString, 132
- s\_metaEngineMutex, 136
- s\_metaEngineSupportBmff, 136
- s\_metaEngineWarnOrError, 136
- s\_rawFileExtensionsdWithDesc, 132
- s\_rawFileExtensionsVersion, 133
- s\_rfc3066ForXMP, 136
- s\_setXmpTagStringFromEntry, 133
- s\_stage, 136
- setExifXmpTagDataVariant, 133
- setMacOSEnvironment, 134
- setOpenCLEnvironment, 134
- setWindowsEnvironment, 134
- showRawCameraList, 134
- startOfDay, 134
- supportedImageMimeTypes, 134
- toolButtonStyleSheet, 134
- unloadQtTranslationFiles, 135
- UnspecifiedOps, 127
- YOLOV5NANO, 125, 127
- YOLOV5XLARGE, 125, 127
- YoloVersions, 127
- digiKam project API reference., 1
- Digikam::AbstractAlbumModel, 140
  - AbstractAlbumModel, 143
  - albumCleared, 143
  - albumData, 143
  - AlbumDataRole, 142
  - albumForIndex, 143
  - AlbumGlobalIdRole, 142
  - AlbumIdRole, 142
  - AlbumPointerRole, 142
  - AlbumSortRole, 142
  - AlbumTitleRole, 142
  - albumType, 143
  - AlbumTypeRole, 142
  - allAlbumsCleared, 143
  - columnHeader, 144
  - decorationRoleData, 144
  - dragDropHandler, 144
  - filterAlbum, 144
  - fontRoleData, 144
  - IgnoreRootAlbum, 142
  - IncludeRootAlbum, 142
  - indexForAlbum, 144
  - isFaceTagModel, 144

- retrieveAlbum, 145
- rootAlbumAvailable, 145
- RootAlbumBehavior, 142
- rootAlbumBehavior, 145
- rootAlbumIndex, 145
- setDragDropHandler, 145
- setDropIndex, 145
- setEnabledDrag, 145
- sortRoleData, 146
- Digikam::AbstractAlbumTreeView, 147
  - AbstractAlbumTreeView, 150
  - adaptColumnsToContent, 150
  - addCustomContextMenuActions, 150
  - AlwaysShowInclusiveCounts, 150
  - contextMenuIcon, 152
  - contextMenuTitle, 152
  - CreateDefaultDelegate, 150
  - CreateDefaultFilterModel, 150
  - CreateDefaultModel, 150
  - currentAlbumChanged, 152
  - doLoadState, 152
  - doSaveState, 152
  - expandEverything, 153
  - expandMatches, 153
  - Flag, 150
  - handleCustomContextMenuAction, 153
  - indexVisuallyAt, 154
  - pixmapForDrag, 154
  - scrollToSelectedAlbum, 154
  - selectedAlbumsChanged, 154
  - setAlbumManagerCurrentAlbum, 154
  - setContextMenuIcon, 154
  - setCurrentAlbums, 154
  - setEnabledContextMenu, 155
  - setExpandNewCurrentItem, 155
  - setExpandOnSingleClick, 155
  - setSelectAlbumOnClick, 155
  - setSelectOnContextMenu, 155
  - showContextMenuAt, 156
  - ShowCountAccordingToSettings, 150
  - slotRootAlbumAvailable, 156
  - viewportEvent, 156
- Digikam::AbstractAlbumTreeView::ContextMenuElement, 157
  - addActions, 157
- Digikam::AbstractAlbumTreeView::Private, 158
- Digikam::AbstractAlbumTreeViewSelectComboBox, 159
  - AbstractAlbumTreeViewSelectComboBox, 161
  - addCheckUncheckContextMenuActions, 162
  - installView, 162
  - sendViewportEventToView, 162
  - setTreeView, 162
- Digikam::AbstractCheckableAlbumModel, 163
  - AbstractCheckableAlbumModel, 168
  - albumCleared, 168
  - albumData, 168
  - allAlbumsCleared, 168
  - checkStateChanged, 168
  - prepareAddExcludeDecoration, 169
  - setAddExcludeTristate, 169
  - setData, 169
  - setRecursive, 169
  - setRootCheckable, 169
  - setTristate, 169
- Digikam::AbstractCheckableAlbumTreeView, 171
  - AbstractCheckableAlbumTreeView, 174
  - albumModel, 175
  - doLoadState, 175
  - doSaveState, 175
  - isRestoreCheckState, 175
  - middleButtonPressed, 175
  - setCheckOnMiddleClick, 175
  - setRestoreCheckState, 176
- Digikam::AbstractCountingAlbumModel, 177
  - albumCleared, 180
  - albumCount, 180
  - albumData, 180
  - albumForId, 181
  - albumName, 181
  - allAlbumsCleared, 181
  - excludeChildrenCount, 181
  - includeChildrenCount, 181
  - setCountHash, 181
  - setup, 182
- Digikam::AbstractCountingAlbumTreeView, 183
- Digikam::AbstractDetector, 186
  - prepareForDetection, 187
- Digikam::AbstractItemDragDropHandler, 187
  - accepts, 188
  - acceptsMimeData, 188
  - createMimeData, 188
  - dropEvent, 188
  - mimeTypes, 188
- Digikam::AbstractMarkerTiler, 190
  - bestRepresentativeIndexFromList, 191
  - getTile, 191
  - getTileGroupState, 192
  - getTileRepresentativeMarker, 192
  - indicesEqual, 192
  - onIndicesClicked, 192
  - pixmapFromRepresentativeIndex, 192
  - prepareTiles, 192
  - setActive, 193
  - tilerFlags, 193
- Digikam::AbstractMarkerTiler::ClickInfo, 193
- Digikam::AbstractMarkerTiler::NonEmptyIterator, 193
- Digikam::AbstractMarkerTiler::Tile, 194
- Digikam::AbstractSearchGroupContainer, 195
  - AbstractSearchGroupContainer, 196
  - addGroupToLayout, 196
  - createSearchGroup, 196
- Digikam::AbstractSpecificAlbumModel, 197
  - columnHeader, 200
- Digikam::AbstractWidgetDelegateOverlay, 200
  - AbstractWidgetDelegateOverlay, 202
  - checkIndex, 202

- checkIndexOnEnter, [202](#)
- createWidget, [202](#)
- hide, [202](#)
- parentWidget, [202](#)
- setActive, [203](#)
- slotEntered, [203](#)
- slotReset, [203](#)
- viewportLeaveEvent, [203](#)
- widgetEnterEvent, [203](#)
- widgetEnterNotifyMultiple, [204](#)
- Digikam::ActionCategorizedView, [205](#)
- Digikam::ActionData, [207](#)
- Digikam::ActionItemModel, [208](#)
  - actionForIndex, [210](#)
  - ActionItemModel, [210](#)
  - createFilterModel, [210](#)
  - hover, [210](#)
  - itemForAction, [210](#)
  - MenuCategoryFlag, [209](#)
  - ParentMenuCategory, [210](#)
  - SortCategoriesAlphabetically, [210](#)
  - SortCategoriesByInsertionOrder, [210](#)
  - ToplevelMenuCategory, [210](#)
- Digikam::ActionJob, [211](#)
  - ~ActionJob, [211](#)
  - ActionJob, [211](#)
  - cancel, [212](#)
  - m\_cancel, [212](#)
  - m\_timer, [212](#)
  - signalDone, [212](#)
  - signalProgress, [212](#)
  - signalStarted, [212](#)
- Digikam::ActionSortFilterProxyModel, [213](#)
- Digikam::ActionTask, [215](#)
- Digikam::ActionThread, [217](#)
  - signalCancelActionTask, [218](#)
  - signalFinished, [218](#)
  - signalQueueProcessed, [218](#)
  - signalStarting, [218](#)
- Digikam::ActionThreadBase, [219](#)
  - appendJobs, [220](#)
  - cancel, [220](#)
  - isEmpty, [220](#)
  - maximumNumberOfThreads, [220](#)
  - pendingCount, [220](#)
  - run, [220](#)
  - setDefaultMaximumNumberOfThreads, [220](#)
  - setMaximumNumberOfThreads, [221](#)
- Digikam::ActionVersionsOverlay, [222](#)
  - checkIndex, [225](#)
  - createButton, [225](#)
  - setActive, [225](#)
  - updateButton, [225](#)
- Digikam::AddBookmarkDialog, [226](#)
- Digikam::AddBookmarkProxyModel, [226](#)
- Digikam::AddTagsComboBox, [228](#)
  - currentTaggingAction, [231](#)
  - setAlbumModels, [231](#)
  - setCurrentTag, [231](#)
  - setParentTag, [232](#)
  - taggingActionActivated, [232](#)
  - taggingActionSelected, [232](#)
- Digikam::AddTagsLineEdit, [233](#)
  - setAlbumModels, [234](#)
  - setCurrentTag, [234](#)
  - setFilterModel, [234](#)
  - setParentTag, [234](#)
  - setSupportingTagModel, [235](#)
  - setTagTreeView, [235](#)
  - slotReturnPressed, [235](#)
  - taggingActionActivated, [235](#)
  - taggingActionSelected, [235](#)
- Digikam::AdvancedMetadataTab, [236](#)
  - AdvancedMetadataTab, [236](#)
- Digikam::AdvancedRenameDialog, [237](#)
- Digikam::AdvancedRenameInput, [238](#)
- Digikam::AdvancedRenameLineEdit, [239](#)
- Digikam::AdvancedRenameListItem, [240](#)
- Digikam::AdvancedRenameManager, [241](#)
- Digikam::AdvancedRenameProcessDialog, [243](#)
- Digikam::AdvancedRenameWidget, [245](#)
  - clear, [246](#)
  - clearParseString, [246](#)
  - focusLineEdit, [246](#)
  - highlightLineEdit, [246](#)
  - parse, [246](#)
  - parser, [247](#)
  - parseString, [247](#)
  - setControlWidgets, [247](#)
  - setLayoutStyle, [247](#)
  - setParser, [248](#)
  - setParseString, [248](#)
- Digikam::AdvancedSettings, [249](#)
- Digikam::AestheticDetector, [250](#)
  - detect, [251](#)
- Digikam::Akonadiface, [251](#)
- Digikam::Album, [252](#)
  - ~Album, [255](#)
  - Album, [255](#)
  - childAlbumIds, [255](#)
  - childAlbums, [255](#)
  - childAtRow, [256](#)
  - childCount, [256](#)
  - clear, [256](#)
  - databaseUrl, [256](#)
  - DATE, [255](#)
  - extraData, [256](#)
  - FACE, [255](#)
  - firstChild, [257](#)
  - globalID, [257](#)
  - id, [258](#)
  - isAncestorOf, [258](#)
  - isRoot, [258](#)
  - isTrashAlbum, [259](#)
  - isUsedByLabelsTree, [259](#)
  - lastChild, [259](#)

- next, [259](#)
- parent, [259](#)
- PHYSICAL, [255](#)
- prepareForDeletion, [259](#)
- prev, [260](#)
- removeExtraData, [260](#)
- rowFromAlbum, [260](#)
- SEARCH, [255](#)
- setExtraData, [260](#)
- setUsedByLabelsTree, [261](#)
- TAG, [255](#)
- title, [261](#)
- Type, [254](#)
- type, [261](#)
- Digikam::AlbumChangeset, [262](#)
- Digikam::AlbumCopyMoveHint, [262](#)
  - AlbumCopyMoveHint, [263](#)
- Digikam::AlbumCustomizer, [263](#)
- Digikam::AlbumDragDropHandler, [264](#)
  - accepts, [265](#)
  - createMimeData, [265](#)
  - dropEvent, [265](#)
  - mimeTypes, [265](#)
- Digikam::AlbumFilterModel, [267](#)
  - ChildMatch, [269](#)
  - compareByOrder, [269](#)
  - compareValue, [269](#)
  - DirectMatch, [269](#)
  - FilterBehavior, [269](#)
  - FullFiltering, [269](#)
  - hasSearchResult, [270](#)
  - isFiltering, [270](#)
  - lessThan, [270](#)
  - matches, [270](#)
  - MatchResult, [269](#)
  - matchResult, [271](#)
  - NoMatch, [269](#)
  - ParentMatch, [269](#)
  - searchTextSettings, [271](#)
  - searchTextSettingsAboutToChange, [271](#)
  - searchTextSettingsChanged, [271](#)
  - setFilterBehavior, [272](#)
  - setSearchTextSettings, [272](#)
  - setSourceAlbumModel, [272](#)
  - setSourceFilterModel, [272](#)
  - setSourceModel, [272](#)
  - signalFilterChanged, [273](#)
  - SimpleFiltering, [269](#)
  - SpecialMatch, [269](#)
  - StrictFiltering, [269](#)
  - updateFilter, [273](#)
- Digikam::AlbumFolderViewSideBarWidget, [274](#)
  - applySettings, [276](#)
  - changeAlbumFromHistory, [276](#)
  - doLoadState, [276](#)
  - doSaveState, [276](#)
  - getCaption, [276](#)
  - getIcon, [276](#)
  - setActive, [277](#)
- Digikam::AlbumHistory, [278](#)
  - addAlbums, [279](#)
- Digikam::AlbumInfo, [280](#)
  - operator<, [280](#)
- Digikam::AlbumIterator, [280](#)
- Digikam::AlbumLabelsSearchHandler, [281](#)
  - albumForSelectedItems, [282](#)
  - generatedName, [282](#)
  - imagesUrls, [282](#)
  - isRestoringSelectionFromHistory, [282](#)
  - restoreSelectionFromHistory, [282](#)
- Digikam::AlbumManager, [283](#)
  - albumTitles, [286](#)
  - allDAAlbums, [286](#)
  - allPAAlbums, [287](#)
  - allSAAlbums, [287](#)
  - allTAAlbums, [287](#)
  - changeDatabase, [287](#)
  - checkDatabaseDirsAfterFirstRun, [287](#)
  - cleanUp, [287](#)
  - clearCurrentAlbums, [288](#)
  - createPAAlbum, [288](#)
  - createSAAlbum, [289](#)
  - createTAAlbum, [290](#)
  - currentAlbums, [290](#)
  - currentPAAlbum, [290](#)
  - currentTAAlbums, [290](#)
  - databaseEqual, [291](#)
  - deleteSAAlbum, [291](#)
  - deleteTAAlbum, [291](#)
  - findAlbum, [292](#)
  - findDAAlbum, [292](#)
  - findOrCreateTAAlbums, [293](#)
  - findPAAlbum, [293](#)
  - findSAAlbum, [294](#)
  - findSAAlbumsBySearchType, [294](#)
  - findTagsWithProperty, [296](#)
  - findTAAlbum, [296](#)
  - getDAAlbumsCount, [296](#)
  - getFaceCount, [296](#)
  - getItemFromAlbum, [297](#)
  - getPAAlbumsCount, [297](#)
  - getRecentlyAssignedTags, [297](#)
  - getTAAlbumsCount, [298](#)
  - getUnconfirmedFaceCount, [298](#)
  - instance, [298](#)
  - isMovingAlbum, [298](#)
  - mergeTAAlbum, [298](#)
  - moveTAAlbum, [299](#)
  - prepareItemCounts, [299](#)
  - refresh, [299](#)
  - renamePAAlbum, [299](#)
  - renameTAAlbum, [300](#)
  - setCurrentAlbums, [300](#)
  - setDatabase, [300](#)
  - signalAlbumAboutToBeAdded, [301](#)
  - signalAlbumAboutToBeDeleted, [301](#)

- signalAlbumAboutToBeMoved, 301
- signalAlbumAdded, 301
- signalAlbumDeleted, 301
- signalAlbumHasBeenDeleted, 301
- signalAlbumMoved, 302
- signalShowOnlyAvailableAlbumsChanged, 302
- startScan, 302
- subTags, 302
- tagNames, 302, 303
- tagPaths, 303
- updatePAlbumIcon, 304
- updateSAlbum, 304
- updateTAlbumIcon, 304
- Digikam::AlbumManager::Private, 305
  - currentAlbums, 306
- Digikam::AlbumManagerCreator, 306
- Digikam::AlbumModel, 307
  - albumData, 312
  - albumForId, 312
  - AlbumModel, 312
  - decorationRoleData, 312
- Digikam::AlbumModelDragDropHandler, 313
  - accepts, 314
  - acceptsMimeData, 314
  - createMimeData, 314
  - dropEvent, 314
  - mimeTypes, 314
- Digikam::AlbumModificationHelper, 315
  - ~AlbumModificationHelper, 316
  - AlbumModificationHelper, 316
  - bindAlbum, 316
  - boundAlbum, 316
  - slotAlbumDelete, 317
  - slotAlbumEdit, 317
  - slotAlbumNew, 317
  - slotAlbumRename, 317
- Digikam::AlbumParser, 319
  - AlbumParser, 321
- Digikam::AlbumPointer< T >, 321
- Digikam::AlbumPointerList< T >, 322
- Digikam::AlbumPropsEdit, 323
- Digikam::AlbumRootChangeset, 324
- Digikam::AlbumRootInfo, 324
- Digikam::AlbumRootLocation, 325
- Digikam::AlbumsDBJobInfo, 327
- Digikam::AlbumsDBJobsThread, 328
  - albumsListing, 329
- Digikam::AlbumSelectComboBox, 330
  - filterModel, 332
  - installView, 332
  - model, 332
  - setAllSelectedText, 333
  - setCheckable, 333
  - setCloseOnActivate, 333
  - setDefaultAlbumModel, 333
  - setNoSelectionText, 333
  - setRecursive, 333
  - setShowCheckStateSummary, 333
  - updateText, 334
- Digikam::AlbumSelectDialog, 334
- Digikam::AlbumSelectionTreeView, 335
  - setEnabledToolTips, 339
  - signalFindDuplicates, 339
- Digikam::AlbumSelectors, 340
  - AlbumSelectors, 341
  - loadState, 341
  - resetPAlbumSelection, 341
  - resetSelection, 341
  - resetTAlbumSelection, 342
  - saveState, 342
  - selectedAlbumIds, 342
  - selectedAlbums, 342
  - selectedAlbumsAndTags, 342
  - selectedTagIds, 342
  - selectedTags, 342
  - setAlbumSelected, 342
  - setTagSelected, 343
  - setTypeSelection, 343
  - typeSelection, 343
  - wholeAlbumsChecked, 343
  - wholeTagsChecked, 343
- Digikam::AlbumSelectTabs, 344
- Digikam::AlbumSelectTreeView, 345
  - ~AlbumSelectTreeView, 349
  - addCustomContextMenuActions, 349
  - AlbumSelectTreeView, 349
  - handleCustomContextMenuAction, 350
  - slotNewAlbum, 350
- Digikam::AlbumSelectWidget, 351
- Digikam::AlbumShortInfo, 351
- Digikam::AlbumSimplified, 352
- Digikam::AlbumsJob, 353
- Digikam::AlbumThumbnailLoader, 355
  - getAlbumThumbnail, 357
  - getAlbumThumbnailDirectly, 357
  - getFaceThumbnailDirectly, 357
  - getStandardTagIcon, 357
  - getTagThumbnail, 357
  - getTagThumbnailDirectly, 357
  - instance, 358
  - RelativeSize, 356
  - setThumbnailSize, 358
  - signalDispatchThumbnailInternal, 358
  - signalFailed, 358
  - signalReloadThumbnails, 358
  - signalThumbnail, 358
  - thumbnailSize, 359
- Digikam::AlbumTreeView, 360
- Digikam::AlbumTreeViewDelegate, 364
- Digikam::AlbumTreeViewSelectComboBox, 365
- Digikam::AlbumWatch, 368
- Digikam::AltLangStrEdit, 369
  - addCurrent, 371
  - allLanguagesRFC3066, 371
  - AltLangStrEdit, 371
  - languageNameRFC3066, 371

- reset, 371
- setLinesVisible, 371
- setTitle, 371
- setTitleWidget, 372
- signalModified, 372
- signalSelectionChanged, 372
- signalValueAdded, 372
- signalValueDeleted, 372
- slotEnabledInternalWidgets, 372
- titleWidget, 373
- Digikam::AltLangStrEdit::Private, 373
- Digikam::AnimatedClearButton, 374
  - setShallBeShown, 375
  - stayVisibleWhenAnimatedOut, 375
- Digikam::AnimatedVisibility, 376
  - AnimatedVisibility, 377
- Digikam::AntiVignettingContainer, 377
- Digikam::AntiVignettingFilter, 378
  - filterAction, 381
  - filterIdentifier, 381
  - readParameters, 381
- Digikam::AntiVignettingSettings, 381
- Digikam::ApplicationSettings, 382
  - askGroupingOperateOnAll, 388
  - getGroupingOperateOnAll, 388
  - getIconShowOverlays, 389
  - getStringComparisonType, 389
    - Natural, 388
    - Normal, 388
  - operationTypeExplanation, 389
  - operationTypeTitle, 389
  - readMsgBoxShouldBeShown, 390
  - saveMsgBoxShouldBeShown, 390
  - setGroupingOperateOnAll, 390
  - setIconShowOverlays, 390
  - setImageSorting, 390
  - setStringComparisonType, 391
  - StringComparisonType, 388
- Digikam::ApplicationSettings::Private, 391
- Digikam::AssignedBatchTools, 396
- Digikam::AssignedListView, 397
- Digikam::AssignedListViewItem, 398
- Digikam::AssignNameOverlay, 400
  - checkIndex, 403
  - createWidget, 403
  - setActive, 403
  - setFocusOnWidget, 403
  - showOnIndex, 403
  - updateFace, 403
  - viewportLeaveEvent, 403
  - visualChange, 404
  - widgetEnterEvent, 404
  - widgetLeaveEvent, 404
- Digikam::AssignNameWidget, 405
  - assigned, 407
  - AssignNameWidget, 407
  - ignoredClicked, 407
  - labelClicked, 407
  - rejected, 408
  - selected, 408
  - setAlbumModels, 408
  - setCurrentTag, 408
  - setMode, 408
  - setParentTag, 408
  - setUserData, 409
- Digikam::AssignNameWidget::Private, 409
  - updateRejectButton, 410
- Digikam::AssignNameWidgetStates, 411
- Digikam::AudPlayerWdg, 414
- Digikam::AutoCrop, 415
  - AutoCrop, 418
  - autoInnerCrop, 418
  - startAnalyse, 418
- Digikam::AutoExpoFilter, 419
  - filterAction, 422
  - filterIdentifier, 422
  - readParameters, 422
- Digikam::AutoLevelsFilter, 423
  - filterAction, 426
  - filterIdentifier, 426
  - readParameters, 426
- Digikam::AutoTagsAssign, 426
  - generateTagsList, 427
- Digikam::AutotagsAssignment, 428
  - AllItems, 431
  - AutotagsAssignment, 431
  - AutotagsAssignmentScanMode, 430
  - NonAssignedItems, 431
  - setUseMultiCoreCPU, 431
- Digikam::AutotagsAssignmentTask, 432
- Digikam::BackendGeonamesRG, 433
  - ~BackendGeonamesRG, 435
  - BackendGeonamesRG, 435
  - backendName, 435
  - callRGBBackend, 435
  - cancelRequests, 436
  - getErrorMessage, 436
  - makeQMapFromXML, 436
- Digikam::BackendGeonamesUSRG, 436
  - ~BackendGeonamesUSRG, 438
  - BackendGeonamesUSRG, 438
  - backendName, 438
  - callRGBBackend, 438
  - cancelRequests, 439
  - getErrorMessage, 439
  - makeQMapFromXML, 439
- Digikam::BackendGoogleMaps, 440
  - ~BackendGoogleMaps, 442
  - addActionToConfigurationMenu, 442
  - backendHumanName, 442
  - backendName, 443
  - geoCoordinates, 443
  - getCenter, 443
  - getMarkerModelLevel, 443
  - getNormalizedBounds, 443
  - getZoom, 443

- isReady, 443
- mapSize, 444
- mapWidget, 444
- mapWidgetDocked, 444
- mouseModeChanged, 444
- readSettingsFromGroup, 444
- regionSelectionChanged, 444
- releaseWidget, 444
- reload, 445
- saveSettingsToGroup, 445
- screenCoordinates, 445
- setActive, 445
- setCenter, 445
- setMarkerPixmap, 445
- setZoom, 445
- updateActionAvailability, 446
- updateClusters, 446
- updateMarkers, 446
- zoomIn, 446
- zoomOut, 446
- Digikam::BackendMarble, 447
  - ~BackendMarble, 450
  - addActionsToConfigurationMenu, 450
  - applyCacheToWidget, 450
  - backendHumanName, 450
  - backendName, 450
  - centerOn, 450
  - eventFilter, 450
  - geoCoordinates, 450
  - GeoPainter\_drawPixmapAtCoordinates, 451
  - getCenter, 451
  - getMarkerModelLevel, 451
  - getNormalizedBounds, 451
  - getProjection, 451
  - getZoom, 451
  - isReady, 452
  - mapSize, 452
  - mapWidget, 452
  - mapWidgetDocked, 452
  - marbleCustomPaint, 452
  - mouseModeChanged, 452
  - readSettingsFromGroup, 452
  - regionSelectionChanged, 452
  - releaseWidget, 453
  - reload, 453
  - saveSettingsToGroup, 453
  - screenCoordinates, 453
  - setActive, 453
  - setCenter, 453
  - setZoom, 453
  - slotScheduleUpdate, 454
  - updateActionAvailability, 454
  - updateClusters, 454
  - updateMarkers, 454
  - zoomIn, 454
  - zoomOut, 454
- Digikam::BackendMarbleLayer, 455
- Digikam::BackendOsmRG, 455
  - ~BackendOsmRG, 457
  - backendName, 457
  - BackendOsmRG, 457
  - callRGBBackend, 457
  - cancelRequests, 458
  - getErrorMessage, 458
  - makeQMapFromXML, 458
- Digikam::BalooInfo, 458
- Digikam::BalooWrap, 459
  - getSemanticInfo, 460
  - setSemanticInfo, 460
- Digikam::BasicDImgFilterGenerator< T >, 461
  - BasicDImgFilterGenerator, 462
  - createFilter, 462
  - displayName, 462
  - supportedFilters, 462
  - supportedVersions, 462
- Digikam::BatchTool, 463
  - apply, 466
  - applyFilter, 466
  - BaseTool, 465
  - BatchTool, 466
  - BatchToolGroup, 465
  - cancel, 466
  - clone, 466
  - ColorTool, 465
  - ConvertTool, 465
  - CustomTool, 465
  - DecorateTool, 465
  - defaultSettings, 466
  - deleteSettingsWidget, 466
  - EnhanceTool, 465
  - errorDescription, 467
  - FiltersTool, 465
  - getNeedResetExifOrientation, 467
  - getResetExifOrientationAllowed, 467
  - image, 467
  - ioFileSettings, 467
  - isCancelled, 467
  - isRawFile, 467
  - loadToDImg, 467
  - m\_settingsWidget, 472
  - MetadataTool, 465
  - outputSuffix, 468
  - rawDecodingSettings, 468
  - registerSettingsWidget, 468
  - savefromDImg, 468
  - setBranchHistory, 468
  - setDRawDecoderSettings, 468
  - setErrorDescription, 469
  - setImageData, 469
  - setInputUrl, 469
  - setIOFileSettings, 469
  - setItemInfo, 469
  - setLastChainedTool, 469
  - setNeedResetExifOrientation, 469
  - setOutputUrl, 470
  - setOutputUrlFromInputUrl, 470

- setRawLoadingRules, [470](#)
- setResetExifOrientationAllowed, [470](#)
- setSaveAsNewVersion, [470](#)
- setSettings, [470](#)
- settingsWidget, [470](#)
- setToolDescription, [471](#)
- setToolIconName, [471](#)
- setToolTitle, [471](#)
- setWorkingUrl, [471](#)
- signalAssignSettings2Widget, [471](#)
- slotAssignSettings2Widget, [471](#)
- toolGroup, [471](#)
- toolGroupToString, [472](#)
- toolOperations, [472](#)
- toolVersion, [472](#)
- TransformTool, [465](#)
- Digikam::BatchToolSet, [472](#)
  - operator==, [473](#)
- Digikam::BatchToolsFactory, [473](#)
- Digikam::BCGContainer, [474](#)
- Digikam::BCGFilter, [475](#)
  - filterAction, [478](#)
  - filterIdentifier, [478](#)
  - readParameters, [478](#)
- Digikam::BCGSettings, [478](#)
- Digikam::BdEngineBackend, [479](#)
  - asDBDateTime, [482](#)
  - BdEngineBackend, [482](#)
  - beginTransaction, [482](#)
  - checkOrSetWALMode, [483](#)
  - close, [483](#)
  - commitTransaction, [483](#)
  - configElement, [483](#)
  - ConnectionError, [482](#)
  - connectionErrorHandling, [483](#)
  - copyQuery, [483](#)
  - databaseType, [483](#)
  - exec, [484](#)
  - execDBAction, [484](#)
  - execDBActionQuery, [484](#)
  - execDirectSql, [484](#)
  - execDirectSqlWithResult, [484](#)
  - execQuery, [485](#)
  - execSql, [485](#)
  - execUpsertDBAction, [486](#)
  - getDBAction, [486](#)
  - getQuery, [486](#)
  - handleQueryResult, [486](#)
  - isCompatible, [486](#)
  - isInTransaction, [487](#)
  - lastError, [487](#)
  - lastSQLError, [487](#)
  - maximumBoundValues, [487](#)
  - NoErrors, [482](#)
  - Open, [482](#)
  - open, [487](#)
  - OpenSchemaChecked, [482](#)
  - prepareQuery, [487](#)
  - queryErrorHandling, [488](#)
  - QueryStateEnum, [482](#)
  - readToList, [488](#)
  - rollbackTransaction, [488](#)
  - setDbEngineErrorHandler, [488](#)
  - setForeignKeyChecks, [488](#)
  - SQLError, [482](#)
  - Status, [482](#)
  - status, [488](#)
  - tables, [489](#)
  - Unavailable, [482](#)
- Digikam::BdEngineBackend::QueryState, [489](#)
- Digikam::BdEngineBackendPrivate, [490](#)
  - connectionErrorAbortQueries, [492](#)
  - connectionErrorContinueQueries, [492](#)
  - currentValidity, [493](#)
  - databaseForThread, [492](#)
  - handleWithErrorHandler, [492](#)
  - queryOperationWakeAll, [492](#)
  - setQueryOperationFlag, [492](#)
- Digikam::BdEngineBackendPrivate::AbstractUnlocker, [493](#)
- Digikam::BdEngineBackendPrivate::AbstractWaitingUnlocker, [494](#)
- Digikam::BdEngineBackendPrivate::BusyWaiter, [495](#)
- Digikam::BdEngineBackendPrivate::ErrorLocker, [497](#)
  - wait, [498](#)
- Digikam::BlackFrameListView, [498](#)
- Digikam::BlackFrameListViewItem, [499](#)
- Digikam::BlackFrameParser, [500](#)
- Digikam::BlackFrameToolTip, [501](#)
  - repositionRect, [502](#)
  - tipContents, [502](#)
- Digikam::BlurDetector, [503](#)
  - detect, [504](#)
- Digikam::BlurFilter, [505](#)
  - BlurFilter, [508](#)
  - filterAction, [508](#)
  - filterIdentifier, [508](#)
  - readParameters, [508](#)
- Digikam::BlurFXFilter, [509](#)
  - filterAction, [512](#)
  - filterIdentifier, [512](#)
  - readParameters, [512](#)
- Digikam::BookmarkNode, [513](#)
- Digikam::BookmarksDialog, [514](#)
- Digikam::BookmarksManager, [515](#)
- Digikam::BookmarksMenu, [517](#)
  - prePopulated, [518](#)
- Digikam::BookmarksModel, [519](#)
- Digikam::BorderContainer, [520](#)
- Digikam::BorderFilter, [522](#)
  - BorderFilter, [525](#)
  - filterAction, [525](#)
  - filterIdentifier, [525](#)
  - readParameters, [525](#)
- Digikam::BorderSettings, [526](#)
- Digikam::BqmlInfolface, [527](#)



- allItemInfoListFromCurrentQueue, [529](#)
- pendingItemInfoListFromCurrentQueue, [529](#)
- selectedItemInfoListFromCurrentQueue, [529](#)
- Digikam::BuildTrashCountersJob, [530](#)
- Digikam::BWSepiaContainer, [531](#)
  - BlackWhiteConversionType, [532](#)
  - BWGeneric, [532](#)
  - BWIlfordSFX200, [532](#)
  - BWKodakHIE, [532](#)
  - BWNoFilter, [532](#)
  - BWNoTone, [532](#)
- Digikam::BWSepiaFilter, [533](#)
  - filterAction, [536](#)
  - filterIdentifier, [536](#)
  - readParameters, [536](#)
- Digikam::BWSepiaSettings, [537](#)
- Digikam::CachedPixmapKey, [538](#)
- Digikam::CachedPixmaps, [538](#)
- Digikam::CameraAutoDetectThread, [539](#)
- Digikam::CameraController, [540](#)
  - getThumbsInfo, [542](#)
- Digikam::CameraFolderDialog, [542](#)
- Digikam::CameraFolderItem, [543](#)
- Digikam::CameraFolderView, [544](#)
- Digikam::CameraHistoryUpdater, [545](#)
- Digikam::CameraInfoDialog, [546](#)
- Digikam::CameraItem, [547](#)
- Digikam::CameraItemList, [548](#)
- Digikam::CameraList, [549](#)
- Digikam::CameraMessageBox, [550](#)
  - informationList, [550](#)
  - warningContinueCancelList, [550](#)
- Digikam::CameraNameHelper, [550](#)
- Digikam::CameraNameOption, [551](#)
  - parseOperation, [552](#)
- Digikam::CameraSelection, [553](#)
- Digikam::CameraThumbsCtrl, [554](#)
  - cameraController, [554](#)
  - getThumbInfo, [554](#)
  - updateThumbInfoFromCache, [555](#)
- Digikam::CameraType, [555](#)
- Digikam::CamItemInfo, [555](#)
  - downloaded, [557](#)
  - DownloadedNo, [557](#)
  - DownloadedYes, [557](#)
  - DownloadFailed, [557](#)
  - DownloadStarted, [557](#)
  - DownloadStatus, [556](#)
  - DownloadUnknown, [557](#)
  - isNull, [557](#)
  - NewPicture, [557](#)
  - operator!=, [557](#)
  - operator==, [557](#)
  - size, [557](#)
  - url, [557](#)
- Digikam::CamItemSortSettings, [558](#)
  - compare, [559](#)
  - compareByOrder, [559](#)
  - compareCategories, [559](#)
  - compareValue, [560](#)
  - DefaultOrder, [559](#)
  - lessThan, [560](#)
  - lessThanByOrder, [560](#)
  - naturalCompare, [560](#)
  - SortOrder, [559](#)
- Digikam::Canvas, [562](#)
  - applyTransform, [565](#)
  - currentImage, [565](#)
  - currentImageFileFormat, [565](#)
  - currentImageFilePath, [565](#)
  - exifRotated, [566](#)
  - fitToSelect, [566](#)
  - getSelectedArea, [566](#)
  - imageHeight, [566](#)
  - imageWidth, [566](#)
  - interface, [566](#)
  - isReadOnly, [566](#)
  - setExifOrient, [566](#)
  - setExposureSettings, [567](#)
  - setICCSettings, [567](#)
  - setSoftProofingEnabled, [567](#)
- Digikam::CaptionEdit, [568](#)
- Digikam::CaptionsMap, [570](#)
  - setAuthorsList, [571](#)
- Digikam::CaptionValues, [571](#)
- Digikam::CaptureDlg, [571](#)
- Digikam::CaptureWidget, [572](#)
- Digikam::CaseModifier, [573](#)
  - parseOperation, [574](#)
- Digikam::CategorizedItemModel, [576](#)
  - ExtraRoles, [577](#)
  - ItemOrderRole, [577](#)
- Digikam::CBContainer, [577](#)
- Digikam::CBFilter, [578](#)
  - filterAction, [581](#)
  - filterIdentifier, [581](#)
  - readParameters, [581](#)
- Digikam::CBSettings, [581](#)
- Digikam::ChangeBookmarkCommand, [582](#)
- Digikam::ChangeFaceRecognitionModelDlg, [583](#)
- Digikam::ChangingDB, [583](#)
- Digikam::CharcoalFilter, [584](#)
  - filterAction, [587](#)
  - filterIdentifier, [587](#)
  - readParameters, [587](#)
- Digikam::CheckableAlbumFilterModel, [588](#)
  - isFiltering, [590](#)
  - matches, [590](#)
- Digikam::ChoiceSearchComboBox, [593](#)
  - ChoiceSearchComboBox, [595](#)
  - installView, [595](#)
  - setLabelText, [595](#)
  - setSearchModel, [595](#)
- Digikam::ChoiceSearchModel, [596](#)
  - checkedDisplayTexts, [597](#)
  - checkedKeys, [597](#)

- resetChecked, [597](#)
- setChecked, [598](#)
- setChoice, [598](#)
- Digikam::ChoiceSearchModel::Entry, [599](#)
  - operator==, [599](#)
- Digikam::CIETongueWidget, [600](#)
- Digikam::ClickDragReleaseItem, [601](#)
  - mousePressEvent, [602](#)
  - started, [602](#)
- Digikam::ClockPhotoDialog, [603](#)
  - setImage, [603](#)
- Digikam::CMat, [604](#)
  - center, [604](#)
  - data, [604](#)
  - radius, [604](#)
  - row\_stride, [604](#)
- Digikam::CollectionImageChangeset, [605](#)
  - Added, [605](#)
  - CollectionImageChangeset, [606](#)
  - Copied, [605](#)
  - Deleted, [605](#)
  - ids, [606](#)
  - Moved, [605](#)
  - Operation, [605](#)
  - operator<<, [606](#)
  - Removed, [605](#)
  - RemovedAll, [605](#)
  - RemovedDeleted, [605](#)
- Digikam::CollectionLocation, [607](#)
  - albumRootPath, [609](#)
  - asQtCaseSensitivity, [609](#)
  - CaseInsensitive, [608](#)
  - CaseSensitive, [608](#)
  - CaseSensitivity, [608](#)
  - caseSensitivity, [609](#)
  - id, [609](#)
  - label, [610](#)
  - LocationAvailable, [609](#)
  - LocationDeleted, [609](#)
  - LocationHidden, [609](#)
  - LocationNull, [609](#)
  - LocationUnavailable, [609](#)
  - Network, [609](#)
  - Status, [608](#)
  - status, [610](#)
  - Type, [609](#)
  - type, [610](#)
  - Undefined, [609](#)
  - UnknownCaseSensitivity, [608](#)
  - VolumeHardWired, [609](#)
  - VolumeRemovable, [609](#)
- Digikam::CollectionManager, [611](#)
  - addLocation, [613](#)
  - album, [614](#)
  - albumRoot, [614](#)
  - albumRootLabel, [614](#)
  - albumRootPath, [614](#)
  - allAvailableAlbumRootPaths, [614](#)
  - allAvailableLocations, [614](#)
  - allLocations, [615](#)
  - changeType, [615](#)
  - checkHardWiredLocations, [615](#)
  - checkLocation, [615](#)
  - isAlbumRoot, [615](#)
  - LocationAllRight, [613](#)
  - LocationCheckResult, [613](#)
  - locationForAlbumRoot, [616](#)
  - locationForAlbumRootId, [616](#)
  - locationForUrl, [616](#)
  - LocationHasProblems, [613](#)
  - LocationInvalidCheck, [613](#)
  - LocationNotAllowed, [613](#)
  - locationPropertiesChanged, [616](#)
  - locationStatusChanged, [616](#)
  - migrateToVolume, [616](#)
  - migrationCandidates, [617](#)
  - oneAlbumRoot, [617](#)
  - refresh, [617](#)
  - removeLocation, [617](#)
  - setLabel, [617](#)
  - setWatchDisabled, [617](#)
- Digikam::CollectionManager::Private, [618](#)
  - findVolumeForLocation, [619](#)
  - findVolumeForUrl, [619](#)
- Digikam::CollectionPage, [620](#)
- Digikam::CollectionScanner, [622](#)
  - cancelled, [625](#)
  - CleanScan, [624](#)
  - completeScan, [625](#)
  - copyFileProperties, [625](#)
  - createHintContainer, [625](#)
  - databaseInitialScanDone, [625](#)
  - FileScanMode, [624](#)
  - finishCompleteScan, [625](#)
  - finishedScanningAlbumRoot, [625](#)
  - getNewIdsList, [626](#)
  - ModifiedScan, [624](#)
  - NormalScan, [624](#)
  - partialScan, [626](#)
  - Rescan, [624](#)
  - safelyRemoveAlbums, [626](#)
  - scanFile, [626](#), [627](#)
  - scannedFiles, [627](#)
  - setNeedFileCount, [627](#)
  - setObserver, [627](#)
  - setPerformFastScan, [627](#)
  - setSignalsEnabled, [627](#)
  - startScanningAlbumRoot, [628](#)
  - totalFilesToScan, [628](#)
- Digikam::CollectionScanner::Private, [628](#)
  - finishScanner, [629](#)
- Digikam::CollectionScannerHintContainer, [630](#)
- Digikam::CollectionScannerHintContainerImplementation, [632](#)
  - clear, [633](#)
  - recordHint, [633](#)

- recordHints, [633](#), [634](#)
- Digikam::CollectionScannerObserver, [635](#)
- Digikam::ColorCorrectionDlg, [636](#)
- Digikam::ColorFXContainer, [636](#)
- Digikam::ColorFXFilter, [637](#)
  - filterAction, [640](#)
  - filterIdentifier, [640](#)
  - readParameters, [640](#)
- Digikam::ColorFXSettings, [640](#)
- Digikam::ColorGradientWidget, [641](#)
- Digikam::ColorLabelFilter, [642](#)
- Digikam::ColorLabelMenuAction, [644](#)
- Digikam::ColorLabelSelector, [645](#)
- Digikam::ColorLabelWidget, [646](#)
  - colorLabels, [647](#)
  - setButtonsExclusive, [647](#)
  - setColorLabels, [648](#)
  - setDescriptionBoxVisible, [648](#)
- Digikam::ComboBoxDelegate, [649](#)
  - paint, [650](#)
  - startEditing, [650](#)
- Digikam::CommentInfo, [650](#)
- Digikam::CommonKeys, [651](#)
  - getDbValue, [652](#)
- Digikam::CompressionDetector, [653](#)
  - detect, [654](#)
- Digikam::ContentAwareContainer, [654](#)
- Digikam::ContentAwareFilter, [655](#)
  - filterAction, [658](#)
  - filterIdentifier, [658](#)
  - readParameters, [658](#)
- Digikam::ContextMenuHelper, [658](#)
  - addAction, [661](#), [662](#)
  - addActionDeleteFaceTag, [662](#)
  - addActionNewAlbum, [662](#)
  - addActionNewTag, [662](#)
  - addActionTagToFaceTag, [663](#)
  - addAlbumCheckUncheckActions, [663](#)
  - addAssignTagsMenu, [663](#)
  - addCreateTagFromAddressbookMenu, [663](#)
  - addExportMenu, [664](#)
  - addGotoMenu, [664](#)
  - addGroupMenu, [664](#)
  - addImportMenu, [664](#)
  - addIQSAction, [665](#)
  - addLabelsAction, [665](#)
  - addOpenAndNavigateActions, [665](#)
  - addQueueManagerMenu, [666](#)
  - addRemoveAllTags, [666](#)
  - addRemoveTagsMenu, [666](#)
  - addSeparator, [666](#)
  - addServicesMenu, [667](#)
  - addStandardActionCopy, [667](#)
  - addStandardActionCut, [667](#)
  - addStandardActionItemDelete, [667](#)
  - addStandardActionLightTable, [668](#)
  - addStandardActionPaste, [668](#)
  - addStandardActionThumbnail, [668](#)
  - addSubMenu, [669](#)
  - ContextMenuHelper, [661](#)
    - exec, [669](#)
    - setAlbumModel, [669](#)
    - setItemFilterModel, [669](#)
  - Digikam::ContextMenuHelper::Private, [670](#)
  - Digikam::CoordinatesOverlayWidget, [670](#)
  - Digikam::CopyOrMoveJob, [671](#)
  - Digikam::CopyrightInfo, [672](#)
  - Digikam::CoreDB, [673](#)
    - ~CoreDB, [678](#)
    - addAlbum, [678](#)
    - addAlbumRoot, [679](#)
    - addImageMetadata, [679](#)
    - addImageRelation, [679](#)
    - addImageRelations, [679](#)
    - addImageTagProperty, [680](#)
    - addItem, [680](#)
    - addItemInformation, [680](#)
    - addItemPosition, [680](#)
    - addItemTag, [681](#)
    - addSearch, [681](#)
    - addTag, [682](#)
    - addTagProperty, [682](#)
    - addTagsToItems, [682](#)
    - addToDownloadHistory, [683](#)
    - addVideoMetadata, [683](#)
    - changeImageComment, [683](#)
    - changeImageMetadata, [683](#)
    - changeItemInformation, [684](#)
    - changeItemPosition, [684](#)
    - changeVideoMetadata, [684](#)
    - copyAlbumProperties, [684](#)
    - copyImageAttributes, [684](#)
    - copyImageProperties, [685](#)
    - copyImageTags, [685](#)
    - copyItem, [685](#)
    - CoreDB, [678](#)
    - databaseUuid, [685](#)
    - deleteAlbum, [685](#)
    - deleteAlbumRoot, [686](#)
    - deleteItem, [686](#)
    - deleteObsoleteItem, [687](#)
    - deleteRemovedItems, [687](#)
    - deleteSearch, [687](#)
    - deleteSearches, [687](#)
    - deleteStaleAlbums, [687](#)
    - deleteTag, [687](#)
    - findByNameAndCreationDate, [688](#)
    - findImageId, [688](#)
    - findInDownloadHistory, [688](#)
    - getAlbumAndSubalbumsForPath, [689](#)
    - getAlbumAverageDate, [689](#)
    - getAlbumForPath, [689](#)
    - getAlbumHighestDate, [690](#)
    - getAlbumLowestDate, [690](#)
    - getAlbumModificationDate, [690](#)
    - getAlbumModificationMap, [691](#)

getAlbumRelativePath, 691  
getAlbumRootId, 691  
getAlbumRoots, 691  
getAlbumShortInfos, 692  
getAlbumsOnAlbumRoot, 692  
getAllCreationDates, 692  
getAllItems, 692  
getAllItemsWithAlbum, 692  
getDatabaseEncoding, 692  
getDirtyOrMissingFacelImageUrls, 693  
getFilterSettings, 693  
getFirstItemWithFaceTag, 693  
getFormatStatistics, 693  
getIdenticalFiles, 693  
getImageld, 693  
getImagelds, 694, 696  
getImageMetadata, 696  
getImageProperty, 696  
getImagesFields, 696  
getImagesRelatedFrom, 697  
getImagesRelatingTo, 697  
getImagesWithImageTagProperty, 697  
getImagesWithProperty, 697  
getImageTagProperties, 697  
getImageUuid, 697  
getItemAlbum, 698  
getItemComments, 698  
getItemCommonTagIds, 698  
getItemCopyright, 698  
getItemFromAlbum, 699  
getItemHistory, 699  
getItemIDsAndURLsInAlbum, 699  
getItemIDsInAlbum, 699  
getItemIDsInTag, 700  
getItemInformation, 700  
getItemName, 700  
getItemNamesInAlbum, 701  
getItemPosition, 701  
getItemScanInfo, 701  
getItemScanInfos, 701  
getItemsForUuid, 701  
getItemShortInfo, 702  
getItemsTagIDs, 702  
getItemsURLsWithTag, 702  
getItemTagIDs, 702  
getItemTagNames, 703  
getItemURLsInAlbum, 703  
getItemURLsInTag, 703  
getListFromImageMetadata, 704  
getNumberOfAllItemsAndAlbums, 704  
getNumberOfImagesInAlbums, 704  
getNumberOfImagesInTagProperties, 704  
getNumberOfImagesInTags, 705  
getNumberOfItemsInAlbum, 705  
getObsoleteItemIds, 705  
getOneRelatedImageEach, 705  
getRecentlyAssignedTags, 705  
getRelatedImagesToByType, 705  
getRelationCloud, 706  
getSearchInfo, 706  
getSearchQuery, 706  
getSetting, 706  
getTagIdsWithProperties, 706  
getTagProperties, 707  
getTagShortInfos, 707  
getTagsWithProperty, 707  
getUniqueHashVersion, 707  
getUserFilterSettings, 707  
getVideoMetadata, 708  
hasImageHistory, 708  
hasTags, 708  
integrityCheck, 708  
makeStaleAlbum, 708  
migrateAlbumRoot, 709  
moveItem, 709  
removeAllImageComments, 709  
removeAllItemCopyrightProperties, 709  
removeImageComment, 709  
removeImageRelation, 710  
removeImageTagProperties, 710  
removeItemAllTags, 710  
removeItemCopyrightProperties, 710  
removeItemPosition, 711  
removeItemPositionAltitude, 711  
removeItems, 711  
removeItemsFromAlbum, 711  
removeItemsPermanently, 712  
removeItemTag, 712  
removeTagProperties, 712  
removeTagsFromItems, 712  
renameAlbum, 713  
renameItem, 713  
scanAlbums, 713  
scanSearches, 713  
scanTags, 713  
setAlbumCaption, 714  
setAlbumCategory, 715  
setAlbumDate, 715  
setAlbumIcon, 715  
setAlbumModificationDate, 715  
setAlbumRootCaseSensitivity, 716  
setAlbumRootLabel, 716  
setAlbumRootPath, 716  
setAlbumRootType, 716  
setFilterSettings, 717  
setImageComment, 717  
setImageProperty, 717  
setItemAlbum, 718  
setItemCopyrightProperty, 718  
setItemHistory, 718  
setItemManualOrder, 718  
setItemModificationDate, 718  
setItemStatus, 718  
setSetting, 719  
setTagIcon, 719  
setTagName, 719

- setTagParentID, [720](#)
  - setUserFilterSettings, [720](#)
  - updateItem, [720](#)
  - updateSearch, [720](#)
  - vacuum, [721](#)
- Digikam::CoreDbAccess, [721](#)
  - backend, [722](#)
  - checkReadyForUse, [722](#)
  - cleanUpDatabase, [722](#)
  - CoreDbAccess, [722](#)
  - databaseWatch, [723](#)
  - db, [723](#)
  - initDbEngineErrorHandler, [723](#)
  - lastError, [723](#)
  - parameters, [723](#)
  - setLastError, [723](#)
  - setParameters, [723](#)
- Digikam::CoreDbAccessUnlock, [724](#)
  - CoreDbAccessUnlock, [724](#)
- Digikam::CoreDbBackend, [725](#)
  - initSchema, [728](#)
  - recordChangeset, [728](#)
  - setCoreDbWatch, [728](#)
- Digikam::CoreDbBackendPrivate, [729](#)
  - transactionFinished, [731](#)
- Digikam::CoreDbBackendPrivate::ChangesetContainer< T >, [731](#)
- Digikam::CoreDbCopyManager, [732](#)
- Digikam::CoreDbDownloadHistory, [732](#)
  - setDownloaded, [733](#)
  - status, [733](#)
- Digikam::CoreDbNameFilter, [733](#)
  - CoreDbNameFilter, [733](#)
  - matches, [734](#)
- Digikam::CoreDbOperationGroup, [734](#)
  - allowLift, [734](#)
  - CoreDbOperationGroup, [734](#)
  - lift, [734](#)
  - resetTime, [735](#)
- Digikam::CoreDbPrivilegesChecker, [735](#)
- Digikam::CoreDbSchemaUpdater, [735](#)
- Digikam::CoreDbTransaction, [735](#)
  - CoreDbTransaction, [736](#)
- Digikam::CoreDbUrl, [737](#)
  - album, [739](#)
  - albumRoot, [739](#)
  - albumRootId, [739](#)
  - albumRootPath, [739](#)
  - albumUrl, [739](#)
  - areaCoordinates, [740](#)
  - CoreDbUrl, [739](#)
  - dateUrl, [740](#)
  - endDate, [740](#)
  - fileUrl, [740](#)
  - fromAlbumAndName, [740](#)
  - fromDateForMonth, [740](#)
  - fromDateForYear, [741](#)
  - fromDateRange, [741](#)
  - fromFileUrl, [741](#)
  - fromTagIds, [741](#)
  - isAlbumUrl, [742](#)
  - mapImagesUrl, [742](#)
  - name, [742](#)
  - parameters, [742](#)
  - searchId, [742](#)
  - searchUrl, [742](#)
  - setParameters, [742](#)
  - startDate, [743](#)
  - tagId, [743](#)
  - tagIds, [743](#)
- Digikam::CoreDbWatch, [744](#)
  - databaseChanged, [745](#)
  - imageChange, [745](#)
- Digikam::CountrySelector, [746](#)
- Digikam::CtrlButton, [747](#)
- Digikam::CurvesBox, [748](#)
- Digikam::CurvesContainer, [749](#)
  - CurvesContainer, [749](#)
  - curvesType, [750](#)
  - initialize, [750](#)
  - isEmpty, [750](#)
  - isStoredLosslessly, [750](#)
- Digikam::CurvesFilter, [751](#)
  - filterAction, [754](#)
  - filterIdentifier, [754](#)
  - readParameters, [754](#)
- Digikam::CurvesSettings, [755](#)
- Digikam::CurvesWidget, [757](#)
  - reset, [758](#)
  - resetUI, [758](#)
  - restoreCurve, [758](#)
  - saveCurve, [759](#)
  - stopHistogramComputation, [759](#)
  - updateData, [759](#)
- Digikam::CustomStepsDoubleSpinBox, [760](#)
  - CustomStepsDoubleSpinBox, [760](#)
  - reset, [761](#)
  - setSingleSteps, [761](#)
  - setSuggestedInitialValue, [761](#)
  - setSuggestedValues, [761](#)
- Digikam::CustomStepsIntSpinBox, [762](#)
  - CustomStepsIntSpinBox, [763](#)
  - enableFractionMagic, [763](#)
  - fractionMagicValue, [763](#)
  - reset, [763](#)
  - setSingleSteps, [763](#)
  - setSuggestedInitialValue, [763](#)
  - setSuggestedValues, [764](#)
- Digikam::DAboutData, [764](#)
- Digikam::DAbstractSliderSpinBox, [766](#)
  - setBlockUpdateSignalOnDrag, [767](#)
  - setInternalValue, [767](#)
- Digikam::DActiveLabel, [768](#)
- Digikam::DAdjustableLabel, [769](#)
- Digikam::DAlbum, [770](#)
  - databaseUrl, [772](#)

- Digikam::DAlbumDrag, [772](#)
- Digikam::DAlbumInfo, [773](#)
- Digikam::DArrowClickLabel, [774](#)
- Digikam::DatabaseBlob, [775](#)
  - read, [775](#)
- Digikam::DatabaseCopyThread, [776](#)
- Digikam::DatabaseFields::DatabaseFieldsEnumIteator<
  - FieldName >, [776](#)
- Digikam::DatabaseFields::DatabaseFieldsEnumIteatorSet<
  - FieldName >, [777](#)
- Digikam::DatabaseFields::FieldMetaInfo<
  - FieldName >, [778](#)
- Digikam::DatabaseFields::Hash< T >, [778](#)
- Digikam::DatabaseFields::Set, [779](#)
- Digikam::DatabaseLoadSaveFileInfoProvider, [780](#)
  - dimensionsHint, [780](#)
  - orientationHint, [780](#)
- Digikam::DatabaseMigrationDialog, [781](#)
- Digikam::DatabaseOption, [782](#)
  - parseOperation, [783](#)
- Digikam::DatabaseOptionDialog, [784](#)
- Digikam::DatabasePage, [786](#)
- Digikam::DatabaseServer, [787](#)
  - isRunning, [788](#)
  - startDatabaseProcess, [788](#)
  - stopDatabaseProcess, [788](#)
- Digikam::DatabaseServerError, [788](#)
  - DatabaseServerErrorEnum, [788](#)
  - NoErrors, [789](#)
  - NotSupported, [789](#)
  - StartError, [789](#)
- Digikam::DatabaseServerStarter, [789](#)
  - instance, [789](#)
- Digikam::DatabaseSettingsWidget, [790](#)
  - checkDatabaseSettings, [791](#)
- Digikam::DatabaseSettingsWidget::Private, [791](#)
- Digikam::DatabaseTask, [792](#)
  - signalAddItemsToProcess, [793](#)
- Digikam::DatabaseVersionManager, [794](#)
  - toplevelDirectory, [795](#)
- Digikam::DatabaseWorkerInterface, [796](#)
- Digikam::DatabaseWriter, [799](#)
- Digikam::DateAlbumModel, [802](#)
  - albumForId, [807](#)
  - albumName, [807](#)
  - DateAlbumModel, [806](#)
  - decorationRoleData, [807](#)
  - monthIndexForDate, [807](#)
  - setPixmaps, [807](#)
  - sortRoleData, [808](#)
- Digikam::DateFolderView, [809](#)
  - doLoadState, [811](#)
  - doSaveState, [811](#)
  - setConfigGroup, [811](#)
- Digikam::DateFolderViewSideBarWidget, [812](#)
  - applySettings, [813](#)
  - changeAlbumFromHistory, [813](#)
  - doLoadState, [814](#)
  - doSaveState, [814](#)
  - getCaption, [814](#)
  - getIcon, [814](#)
  - setActive, [814](#)
- Digikam::DateFormat, [815](#)
- Digikam::DateOption, [816](#)
  - parseOperation, [817](#)
- Digikam::DateOptionDialog, [818](#)
- Digikam::DatePickerValidator, [819](#)
- Digikam::DatePickerYearSelector, [820](#)
  - DatePickerYearSelector, [821](#)
- Digikam::DatesDBJobInfo, [821](#)
- Digikam::DatesDBJobsThread, [823](#)
  - datesListing, [824](#)
- Digikam::DatesJob, [825](#)
- Digikam::DateTreeView, [827](#)
- Digikam::DbCleaner, [831](#)
  - setUseMultiCoreCPU, [833](#)
- Digikam::DbEngineAccess, [834](#)
  - checkReadyForUse, [834](#)
- Digikam::DbEngineAction, [834](#)
- Digikam::DbEngineActionElement, [834](#)
- Digikam::DbEngineActionType, [834](#)
  - getActionValue, [835](#)
  - isValue, [835](#)
  - setActionValue, [835](#)
  - setValue, [835](#)
- Digikam::DbEngineConfig, [835](#)
- Digikam::DbEngineConfigSettings, [836](#)
- Digikam::DbEngineConfigSettingsLoader, [836](#)
- Digikam::DbEngineConnectionChecker, [837](#)
- Digikam::DbEngineErrorAnswer, [838](#)
  - connectionErrorContinueQueries, [839](#)
- Digikam::DbEngineErrorHandler, [839](#)
  - connectionError, [840](#)
  - consultUserForError, [840](#)
- Digikam::DbEngineGuiErrorHandler, [841](#)
- Digikam::DbEngineLocking, [842](#)
- Digikam::DbEngineParameters, [842](#)
  - DbEngineParameters, [844](#)
  - defaultMysqlAdminCmd, [844](#)
  - defaultMysqlInitCmd, [844](#)
  - defaultMysqlServerCmd, [844](#)
  - defaultMysqlUpgradeCmd, [844](#)
  - defaultParameters, [844](#)
  - faceParameters, [844](#)
  - getCoreDatabaseNameOrDir, [844](#)
  - hash, [845](#)
  - internalServerMysqlInitCmd, [846](#)
  - isValid, [845](#)
  - parametersForSQLite, [845](#)
  - readFromConfig, [845](#)
  - serverPrivatePath, [845](#)
  - setCoreDatabasePath, [845](#)
  - setInternalServerPath, [845](#)
  - similarityParameters, [846](#)
  - SQLiteDatabaseType, [846](#)
  - thumbnailParameters, [846](#)

- Digikam::DbEngineSqlQuery, [847](#)
- Digikam::DbEngineThreadData, [847](#)
- Digikam::DbHeaderListItem, [848](#)
- Digikam::DBinaryIface, [849](#)
- Digikam::DBinarySearch, [851](#)
- Digikam::DBInfolface, [852](#)
  - albumChooser, [854](#)
  - albumChooserItems, [854](#)
  - albumInfo, [854](#)
  - albumItems, [854](#)
  - albumItems, [854](#)
  - allAlbumItems, [855](#)
  - currentAlbumItems, [855](#)
  - currentSelectedItems, [855](#)
  - defaultUploadUrl, [855](#)
  - deleteImage, [855](#)
  - itemInfo, [855](#)
  - openSetupPage, [855](#)
  - parseAlbumItemsRecursive, [856](#)
  - passShortcutActionsToWidget, [856](#)
  - setItemInfo, [856](#)
  - supportAlbums, [856](#)
  - tagFilterModel, [856](#)
  - uploadUrl, [856](#)
  - uploadWidget, [856](#)
- Digikam::DBJob, [857](#)
- Digikam::DBJobInfo, [858](#)
- Digikam::DBJobsManager, [859](#)
  - instance, [860](#)
  - startAlbumsJobThread, [860](#)
  - startDatesJobThread, [860](#)
  - startGPSJobThread, [860](#)
  - startSearchesJobThread, [861](#)
  - startTagsJobThread, [861](#)
- Digikam::DBJobsThread, [862](#)
  - connectFinishAndErrorSignals, [863](#)
  - error, [863](#)
  - errorsList, [863](#)
  - hasErrors, [863](#)
- Digikam::DbKeysCollection, [864](#)
  - addId, [865](#)
  - collectionName, [865](#)
  - DbKeysCollection, [865](#)
  - getDbValue, [865](#)
  - getValue, [867](#)
  - ids, [867](#)
- Digikam::DbKeySelector, [868](#)
- Digikam::DbKeySelectorItem, [869](#)
- Digikam::DbKeySelectorView, [870](#)
- Digikam::DbShrinkDialog, [871](#)
- Digikam::DBStatDlg, [872](#)
- Digikam::DBusyDlg, [873](#)
- Digikam::DBusyThread, [874](#)
- Digikam::DCameraDragObject, [875](#)
- Digikam::DCameraItemListAdapter, [876](#)
- Digikam::DCategorizedSortFilterProxyModel, [877](#)
  - AdditionalRoles, [878](#)
  - CategoryDisplayRole, [878](#)
  - CategorySortRole, [878](#)
  - compareCategories, [879](#)
  - isCategorizedModel, [879](#)
  - lessThan, [879](#)
  - setCategorizedModel, [880](#)
  - setSortCategoriesByNaturalComparison, [880](#)
  - sort, [880](#)
  - sortCategoriesByNaturalComparison, [881](#)
  - sortColumn, [881](#)
  - sortOrder, [881](#)
  - subSortLessThan, [881](#)
- Digikam::DCategorizedSortFilterProxyModel::Private, [882](#)
- Digikam::DCategorizedView, [882](#)
  - categorizedIndexesIn, [884](#)
  - categoryAt, [885](#)
  - categoryRange, [885](#)
  - categoryVisualRect, [885](#)
  - setDrawDraggedItems, [885](#)
- Digikam::DCategorizedView::Private, [886](#)
  - cacheCategory, [887](#)
  - cachedRectCategory, [887](#)
  - cachedRectIndex, [887](#)
  - cacheIndex, [887](#)
  - categoryUpperBound, [887](#)
  - categoryVisualRect, [888](#)
  - contentsSize, [888](#)
  - drawDraggedItems, [888](#)
  - drawNewCategory, [888](#)
  - elementsInfo, [890](#)
  - intersectionSet, [888](#)
  - selectionForRect, [889](#)
  - updateScrollbars, [889](#)
  - visualCategoryRectInViewport, [889](#)
  - visualRect, [889](#)
  - visualRectInViewport, [889](#)
- Digikam::DCategorizedView::Private::ElementInfo, [890](#)
- Digikam::DCategoryDrawer, [891](#)
  - actionRequested, [892](#)
  - categoryHeight, [892](#)
  - collapseOrExpandClicked, [893](#)
  - DCategoryDrawer, [892](#)
  - drawCategory, [893](#)
  - leftMargin, [893](#)
  - mouseButtonDoubleClicked, [893](#)
  - mouseButtonPressed, [894](#)
  - mouseButtonReleased, [894](#)
  - mouseLeft, [895](#)
  - mouseMoved, [895](#)
  - rightMargin, [895](#)
  - view, [895](#)
- Digikam::DClickLabel, [896](#)
- Digikam::DColor, [897](#)
  - blendZero, [898](#)
  - convertToSixteenBit, [898](#)
  - DColor, [898](#)
  - getHSL, [898](#)
  - getYCbCr, [899](#)

- premultiply, 899
- setColor, 899
- setHSL, 899
- setPixel, 899
- setYCbCr, 900
- Digikam::DColorComposer, 900
  - compose, 901
  - CompositingOperation, 901
  - getComposer, 901
- Digikam::DColorSelector, 902
- Digikam::DColorValueSelector, 903
  - chooserMode, 905
  - colorValue, 905
  - drawContents, 905
  - drawPalette, 905
  - hue, 905
  - saturation, 906
  - setChooserMode, 906
  - setColorValue, 906
  - setHue, 906
  - setSaturation, 907
  - updateContents, 907
- Digikam::DComboBox, 908
- Digikam::DConfigDlg, 909
  - ~DConfigDlg, 911
  - addActionButton, 912
  - addPage, 912
  - addSubPage, 913
  - button, 914
  - buttonBox, 914
  - currentPage, 914
  - currentPageChanged, 914
  - DConfigDlg, 911
  - FaceType, 911
  - insertPage, 915
  - pageRemoved, 916
  - pageWidget, 916
  - removePage, 916
  - setButtonBox, 916
  - setConfigGroup, 917
  - setcurrentPage, 917
  - setFaceType, 917
  - setPageWidget, 917
  - setStandardButtons, 918
- Digikam::DConfigDlgMngr, 919
  - ~DConfigDlgMngr, 921
  - addWidget, 921
  - changedMap, 921
  - DConfigDlgMngr, 920
  - getCustomProperty, 921
  - getCustomPropertyChangedSignal, 921
  - getUserProperty, 921
  - getUserPropertyChangedSignal, 922
  - hasChanged, 922
  - init, 922
  - initMaps, 922
  - isDefault, 922
  - parseChildren, 922
  - property, 923
  - propertyMap, 923
  - setProperty, 923
  - settingsChanged, 923
  - setupWidget, 924
  - updateSettings, 924
  - updateWidgets, 924
  - updateWidgetsDefault, 924
  - widgetModified, 924
- Digikam::DConfigDlgModel, 925
  - ~DConfigDlgModel, 926
  - DConfigDlgModel, 926
  - HeaderRole, 926
  - Role, 926
  - WidgetRole, 926
- Digikam::DConfigDlgModelPrivate, 927
- Digikam::DConfigDlgStackedWidget, 928
- Digikam::DConfigDlgTitle, 929
  - autoHideTimeout, 931
  - comment, 931
  - DConfigDlgTitle, 931
  - ErrorMessage, 931
  - ImageAlignment, 930
  - ImageLeft, 931
  - ImageRight, 931
  - InfoMessage, 931
  - MessageType, 931
  - pixmap, 932
  - PlainTextMessage, 931
  - setAutoHideTimeout, 932
  - setBuddy, 932
  - setComment, 933
  - setPixmap, 933, 934
  - setText, 934
  - setWidget, 935
  - text, 935
  - WarningMessage, 931
- Digikam::DConfigDlgTitle::Private, 935
  - iconTypeToIconName, 936
- Digikam::DConfigDlgView, 936
  - ~DConfigDlgView, 939
  - createView, 939
  - currentPage, 939
  - currentPageChanged, 939
  - DConfigDlgView, 939
  - FaceType, 938
  - faceType, 939
  - itemDelegate, 939
  - model, 940
  - setCurrentPage, 940
  - setDefaultWidget, 940
  - setFaceType, 940
  - setItemDelegate, 940
  - setModel, 941
  - showPageHeader, 941
  - viewPosition, 941
- Digikam::DConfigDlgViewPrivate, 942
  - \_k\_dataChanged, 943



- [\\_k\\_modelChanged](#), 943
- [Digikam::DConfigDlgWdg](#), 943
  - [~DConfigDlgWdg](#), 946
  - [addPage](#), 946, 947
  - [addSubPage](#), 947, 948
  - [currentPage](#), 948
  - [currentPageChanged](#), 948
  - [DConfigDlgWdg](#), 946
  - [insertPage](#), 949
  - [pageRemoved](#), 950
  - [pageToggled](#), 950
  - [removePage](#), 950
  - [setCurrentPage](#), 950
- [Digikam::DConfigDlgWdgItem](#), 951
  - [~DConfigDlgWdgItem](#), 953
  - [changed](#), 953
  - [DConfigDlgWdgItem](#), 952, 953
  - [enabled](#), 956
  - [header](#), 953
  - [icon](#), 953
  - [isCheckedable](#), 953
  - [isChecked](#), 954
  - [isEnabled](#), 954
  - [name](#), 954
  - [setCheckable](#), 954
  - [setChecked](#), 954
  - [setEnabled](#), 954
  - [setHeader](#), 954
  - [setIcon](#), 955
  - [setName](#), 955
  - [toggled](#), 955
  - [widget](#), 955
- [Digikam::DConfigDlgWdgModel](#), 956
  - [~DConfigDlgWdgModel](#), 958
  - [addPage](#), 958
  - [addSubPage](#), 959
  - [columnCount](#), 960
  - [DConfigDlgWdgModel](#), 958
  - [index](#), 960
  - [insertPage](#), 960, 961
  - [item](#), 961
  - [removePage](#), 961
  - [toggled](#), 962
- [Digikam::DConfigDlgWdgModelPrivate](#), 963
- [Digikam::DConfigDlgWdgPrivate](#), 964
- [Digikam::DCursorTracker](#), 965
  - [setText](#), 966
- [Digikam::DDateEdit](#), 967
  - [assignDate](#), 968
  - [date](#), 968
  - [dateChanged](#), 968
  - [isReadOnly](#), 969
  - [setDate](#), 969
  - [setReadOnly](#), 969
  - [setupKeywords](#), 969
- [Digikam::DDatePicker](#), 970
  - [~DDatePicker](#), 972
  - [date](#), 972
  - [dateChanged](#), 972
  - [dateEntered](#), 973
  - [dateSelected](#), 973
  - [dateTable](#), 973
  - [DDatePicker](#), 972
  - [fontSize](#), 973
  - [hasCloseButton](#), 973
  - [setCloseButton](#), 974
  - [setDate](#), 974
  - [setFontSize](#), 974
  - [sizeHint](#), 974
  - [tableClicked](#), 974
- [Digikam::DDatePicker::Private](#), 975
  - [fillWeeksCombo](#), 976
- [Digikam::DDatePickerPopup](#), 976
  - [dateChanged](#), 978
  - [datePicker](#), 978
  - [DDatePickerPopup](#), 977
  - [items](#), 978
- [Digikam::DDateTable](#), 979
  - [aboutToShowContextMenu](#), 981
  - [date](#), 981
  - [dateChanged](#), 981
  - [dateFromPos](#), 981
  - [event](#), 981
  - [mousePressEvent](#), 982
  - [popupMenuEnabled](#), 982
  - [posFromDate](#), 982
  - [setCustomDatePainting](#), 982
  - [setDate](#), 982
  - [setFontSize](#), 982
  - [setPopupMenuEnabled](#), 983
  - [sizeHint](#), 983
  - [tableClicked](#), 983
  - [unsetCustomDatePainting](#), 983
- [Digikam::DDateTable::Private](#), 984
  - [date](#), 985
  - [fontsize](#), 985
  - [maxCell](#), 985
  - [numDayColumns](#), 985
  - [numDaysThisMonth](#), 986
  - [numWeekRows](#), 986
  - [weekDayFirstOfMonth](#), 986
- [Digikam::DDateTable::Private::DatePaintingMode](#), 986
- [Digikam::DDateTimeEdit](#), 987
  - [~DDateTimeEdit](#), 988
  - [dateTime](#), 988
  - [dateTimeChanged](#), 989
  - [DDateTimeEdit](#), 988
  - [setDateTime](#), 989
- [Digikam::DDoubleNumInput](#), 989
- [Digikam::DDoubleSliderSpinBox](#), 991
  - [setInternalValue](#), 993
  - [valueString](#), 993
- [Digikam::DefaultRenameParser](#), 994
- [Digikam::DefaultValueDialog](#), 995
- [Digikam::DefaultValueModifier](#), 996
  - [parseOperation](#), 998

- Digikam::DefaultVersionNamingScheme, 999
  - baseName, 1000
  - directory, 1000
  - incrementedCounter, 1000
  - initialCounter, 1000
  - intermediateDirectory, 1000
  - intermediateFileName, 1001
  - versionFileName, 1001
- Digikam::DeleteDialog, 1002
- Digikam::DeleteItem, 1003
- Digikam::DeleteItemList, 1004
- Digikam::DeleteJob, 1005
- Digikam::DeleteWidget, 1006
- Digikam::DeltaTime, 1007
  - isNull, 1007
- Digikam::DetByClockPhotoButton, 1008
- Digikam::DetectionBenchmark, 1009
  - result, 1011
- Digikam::DetectionWorker, 1012
- Digikam::DExpanderBox, 1014
  - addItem, 1015
  - insertItem, 1015
- Digikam::DExpanderBoxExclusive, 1017
  - setIsToolBox, 1018
- Digikam::DFileDialog, 1019
- Digikam::DFileOperations, 1020
  - copyFile, 1020
  - copyFiles, 1020
  - copyFolderRecursively, 1021
  - copyModificationTime, 1021
  - findExecutable, 1021
  - getUniqueFileUrl, 1021
  - getUniqueFolderUrl, 1021
  - localFileRename, 1021
  - openFilesWithDefaultApplication, 1022
  - openInFileManager, 1022
  - removeAndCopyFile, 1022
  - renameFile, 1022
  - setModificationTime, 1022
  - sidecarFiles, 1022
- Digikam::DFileSelector, 1024
- Digikam::DFontProperties, 1026
  - ~DFontProperties, 1029
  - backgroundColor, 1029
  - color, 1029
  - DFontProperties, 1028
  - DisplayFlag, 1027
  - enableColumn, 1029
  - font, 1030
  - FontColumn, 1027
  - FontDiff, 1028
  - fontDiffFlags, 1030
  - FontListCriteria, 1028
  - fontSelected, 1030
  - getFontList, 1030
  - makeColumnVisible, 1030
  - sampleText, 1031
  - setBackgroundColor, 1031
  - setColor, 1031
  - setFont, 1031
  - setSampleBoxVisible, 1032
  - setSampleText, 1032
  - setSizesRelative, 1032
  - sizeHint, 1032
  - sizeRelative, 1032
- Digikam::DFontSelect, 1033
- Digikam::DGradientSlider, 1035
- Digikam::DHBox, 1036
- Digikam::DHistoryView, 1037
- Digikam::DHueSaturationSelector, 1038
  - ~DHueSaturationSelector, 1040
  - chooserMode, 1040
  - colorValue, 1040
  - DHueSaturationSelector, 1040
  - drawContents, 1040
  - drawPalette, 1041
  - hue, 1041
  - saturation, 1041
  - setChooserMode, 1041
  - setColorValue, 1042
  - setHue, 1042
  - setSaturation, 1042
  - updateContents, 1042
- Digikam::DigikamApp, 1043
  - infolface, 1045
- Digikam::DigikamApp::Private, 1046
- Digikam::DigikamItemDelegate, 1049
  - updateRects, 1053
- Digikam::DigikamItemDelegatePrivate, 1054
- Digikam::DigikamItemView, 1057
  - activated, 1063
  - confirmFaces, 1063
  - hasHiddenGroupedImages, 1064
  - ignoreFaces, 1064
  - rejectFaces, 1064
  - removeFaces, 1064
  - setThumbnailSize, 1064
  - showContextMenu, 1064
  - showContextMenuOnInfo, 1065
  - slotSetupChanged, 1065
  - unknownFaces, 1065
- Digikam::DigikamItemView::Private, 1066
- Digikam::DImageHistory, 1067
  - adjustReferredImages, 1068
  - clearReferredImages, 1068
  - entries, 1068
  - hasActions, 1068
  - isEmpty, 1068
  - isNull, 1068
  - isValid, 1068
  - moveCurrentReferredImage, 1069
  - operator<<, 1069
  - purgePathFromReferredImages, 1069
  - referredImages, 1069
  - toXml, 1069
- Digikam::DImageHistory::Entry, 1070

- action, [1070](#)
- Digikam::DImg, [1071](#)
  - addAsReferredImage, [1078](#)
  - addCurrentUniqueImageId, [1078](#)
  - bitBlendImage, [1078](#)
  - bitBlendImageOnColor, [1078](#)
  - bitBlitImage, [1078](#)
  - bitsDepth, [1079](#)
  - bytesDepth, [1079](#)
  - colorModelToString, [1079](#)
  - convertDepth, [1079](#)
  - convertToSixteenBit, [1079](#)
  - copy, [1079](#), [1080](#)
  - copyImageData, [1080](#)
  - copyMetaData, [1080](#)
  - copyQImage, [1080](#)
  - createHistoryImageId, [1080](#)
  - createImageUniqueid, [1080](#)
  - CreateNewImageHistoryUUID, [1076](#)
  - CreateNewMetadataPreview, [1076](#)
  - crop, [1080](#)
  - detach, [1081](#)
  - detectedFormat, [1081](#)
  - DImg, [1076](#), [1077](#)
  - exifOrientation, [1081](#)
  - fileFormat, [1081](#)
  - fileOriginData, [1081](#)
  - fill, [1082](#)
  - FORMAT, [1075](#)
  - format, [1082](#)
  - getMetadata, [1082](#)
  - getPixelColor, [1082](#)
  - getUniqueHash, [1083](#)
  - getUniqueHashVersion, [1083](#)
  - hasTransparentPixels, [1083](#)
  - imageSavedAs, [1083](#)
  - isAnimatedImage, [1084](#)
  - isReadOnly, [1084](#)
  - lastSavedFilePath, [1084](#)
  - loadItemInfo, [1084](#)
  - operator=, [1084](#)
  - operator==, [1084](#)
  - orientation, [1085](#)
  - originalBitDepth, [1085](#)
  - originalColorModel, [1085](#)
  - originalFilePath, [1085](#)
  - originalRatioSize, [1085](#)
  - originalSize, [1085](#)
  - PrepareMetadataFlag, [1075](#)
  - prepareMetadataToSave, [1085](#)
  - pureColorMask, [1086](#)
  - putImageData, [1086](#)
  - QIMAGE, [1075](#)
  - rawDecodingSettings, [1086](#)
  - removeAlphaChannel, [1086](#)
  - RemoveOldMetadataPreviews, [1076](#)
  - reset, [1087](#)
  - ResetExifOrientationTag, [1076](#)
  - resetMetaData, [1087](#)
  - resize, [1087](#)
  - reverseExifRotate, [1087](#)
  - reverseRotateAndFlip, [1087](#)
  - rotateAndFlip, [1087](#)
  - savedFormat, [1087](#)
  - setHistoryBranchAfter, [1088](#)
  - smoothScale, [1088](#)
  - smoothScaleClipped, [1088](#)
  - smoothScaleSection, [1088](#)
  - striplImageData, [1089](#)
  - transform, [1089](#)
  - wasExifRotated, [1089](#)
- Digikam::DImgBuiltinFilter, [1089](#)
  - apply, [1091](#)
  - createThreadedFilter, [1091](#)
  - Crop, [1090](#)
  - DImgBuiltinFilter, [1090](#), [1091](#)
  - displayName, [1091](#)
  - filterAction, [1091](#)
  - isSupported, [1091](#)
  - isValid, [1092](#)
  - Resize, [1090](#)
  - reverseFilter, [1092](#)
  - setAction, [1092](#)
  - supportedVersions, [1092](#)
  - Type, [1090](#)
- Digikam::DImgChildItem, [1093](#)
  - boundingRect, [1095](#)
  - DImgChildItem, [1095](#)
  - originalRect, [1095](#)
  - parentDImgItem, [1095](#)
  - positionChanged, [1095](#)
  - positionOnImageChanged, [1095](#)
  - rect, [1095](#)
  - relativeRect, [1096](#)
  - setOriginalPos, [1096](#)
  - setPos, [1096](#)
  - setRectInSceneCoordinates, [1096](#)
  - setRelativePos, [1096](#)
- Digikam::DImgFilterGenerator, [1097](#)
  - createFilter, [1098](#)
  - displayName, [1098](#)
  - isSupported, [1098](#)
  - supportedFilters, [1098](#)
  - supportedVersions, [1098](#)
- Digikam::DImgFilterManager, [1099](#)
  - addGenerator, [1100](#)
  - createFilter, [1100](#)
  - displayName, [1100](#)
  - filterIcon, [1100](#)
  - i18nDisplayName, [1101](#)
  - isRawConversion, [1101](#)
  - isSupported, [1101](#)
  - supportedFilters, [1101](#)
  - supportedVersions, [1101](#)
- Digikam::DImgLoader, [1102](#)
  - checkAllocation, [1104](#)

- LoadAll, 1103
- LoadFlag, 1103
- LoadICCDData, 1103
- LoadImageData, 1103
- LoadImageHistory, 1103
- LoadItemInfo, 1103
- LoadMetadata, 1103
- LoadPreview, 1103
- LoadUniqueHash, 1103
- new\_failureTolerant, 1104
- Digikam::DImgLoaderObserver, 1105
  - continueQuery, 1106
  - granularity, 1106
  - progressInfo, 1106
- Digikam::DImgLoaderSettings, 1106
  - parameters, 1107
  - setSettings, 1107
  - settings, 1107
  - signalSettingsChanged, 1107
- Digikam::DImgPreviewItem, 1108
  - userLoadingHint, 1110
- Digikam::DImgStaticPriv, 1110
- Digikam::DImgThreadedAnalyser, 1111
  - DImgThreadedAnalyser, 1114
  - startAnalyse, 1114
- Digikam::DImgThreadedFilter, 1114
  - cancelFilter, 1117
  - cleanupFilter, 1117
  - DImgThreadedFilter, 1116, 1117
  - filterAction, 1117
  - filterIdentifier, 1118
  - filterImage, 1118
  - finished, 1118
  - initFilter, 1119
  - initSlave, 1119
  - m\_destImage, 1121
  - m\_master, 1121
  - m\_name, 1121
  - m\_orgImage, 1121
  - m\_progressBegin, 1122
  - m\_slave, 1122
  - modulateProgress, 1119
  - multithreadedSteps, 1119
  - parametersSuccessfullyRead, 1119
  - postProgress, 1119
  - progress, 1120
  - run, 1120
  - setFilterVersion, 1120
  - setSlave, 1120
  - setupAndStartDirectly, 1120
  - setupFilter, 1120
  - started, 1121
  - startFilter, 1121
  - startFilterDirectly, 1121
- Digikam::DImgThreadedFilter::DefaultFilterAction < Filter >, 1123
  - supportOlderVersionIf, 1125
- Digikam::DInfoInterface, 1126
  - albumChooser, 1128
  - currentSelectedItems, 1128
  - defaultUploadUrl, 1128
  - deleteImage, 1128
  - openSetupPage, 1128
  - passShortcutActionsToWidget, 1128
  - slotDateTimeForUrl, 1128
  - slotMetadataChangedForUrl, 1129
  - tagFilterModel, 1129
  - uploadWidget, 1129
- Digikam::DIntNumInput, 1130
- Digikam::DIntRangeBox, 1131
  - maxValue, 1131
  - minValue, 1131
  - setEnabled, 1132
  - setInterval, 1132
  - setRange, 1132
  - setSuffix, 1132
- Digikam::DIO, 1133
  - copy, 1134
- Digikam::DirectoryNameOption, 1135
  - parseOperation, 1136
- Digikam::DisjointMetadata, 1137
  - changedFlags, 1139
  - colorLabel, 1139
  - colorLabelInterval, 1139
  - comments, 1139
  - dateTime, 1140
  - dateTimeChanged, 1140
  - dateTimeInterval, 1140
  - dateTimeStatus, 1140
  - FullWrite, 1139
  - FullWriteIfChanged, 1139
  - keywords, 1140
  - metadataTemplate, 1140
  - PartialWrite, 1139
  - pickLabel, 1140
  - pickLabelInterval, 1141
  - rating, 1141
  - ratingInterval, 1141
  - replaceColorLabel, 1141
  - setDateTime, 1141
  - tags, 1141
  - titles, 1142
  - willWriteMetadata, 1142
  - write, 1142
  - WriteMode, 1139
- Digikam::DisjointMetadata::Private, 1143
- Digikam::DisjointMetadataDataFields, 1145
  - MetadataAvailable, 1146
  - MetadataDisjoint, 1146
  - MetadataInvalid, 1146
  - Status, 1146
- Digikam::DistortionFXFilter, 1147
  - filterAction, 1150
  - filterIdentifier, 1150
  - readParameters, 1150
- Digikam::DItemDelegate, 1151

- acceptsToolTip, 1152
- gridSize, 1152
- mouseMoved, 1152
- setDefaultViewOptions, 1153
- setThumbnailSize, 1153
- Digikam::DItemDrag, 1154
- Digikam::DItemInfo, 1155
- Digikam::DItemsList, 1157
  - appendControlButtonsWidget, 1159
  - checkSelection, 1159
  - loadImagesFromCurrentAlbum, 1159
  - setControlButtonsPlacement, 1159
  - setIsLessThanHandler, 1159
- Digikam::DItemsListView, 1160
- Digikam::DItemsListViewItem, 1161
  - updateItemWidgets, 1162
- Digikam::DItemToolTip, 1162
  - tipContents, 1163
- Digikam::DKCamera, 1164
  - capture, 1166
  - getFreeSpace, 1166
  - getItemsInfoList, 1166
  - getPreview, 1166
- Digikam::DLabelExpander, 1167
- Digikam::DLineWidget, 1168
- Digikam::DLogoAction, 1169
- Digikam::DMessageBox, 1169
  - readMsgBoxShouldBeShown, 1170
  - saveMsgBoxShouldBeShown, 1170
  - showContinueCancel, 1170
  - showContinueCancelList, 1171
  - showContinueCancelWidget, 1171
  - showInformationList, 1171
  - showInformationWidget, 1171
  - showYesNo, 1171
  - showYesNoList, 1172
  - showYesNoWidget, 1172
- Digikam::DMetadata, 1173
  - addToXmpTagStringBag, 1184
  - countryCodeMap, 1184
  - countryCodeMap2, 1184
  - getCameraSerialNumber, 1184
  - getCopyrightInformation, 1184
  - getIccProfile, 1184
  - getItemFacesMap, 1184
  - getLensDescription, 1185
  - getMetadataField, 1185
  - getMsecsInfo, 1186
  - getVideoInformation, 1186
  - getXmpKeywords, 1186
  - getXmpSubCategories, 1186
  - getXmpSubjects, 1186
  - load, 1186
  - loadUsingFFmpeg, 1186
  - loadUsingRawEngine, 1187
  - mSecTimeStamp, 1187
  - possibleValuesForEnumField, 1187
  - removeExifColorSpace, 1187
  - removeFromXmpTagStringBag, 1187
  - removeItemFacesMap, 1188
  - removeXmpKeywords, 1188
  - removeXmpSubCategories, 1188
  - removeXmpSubjects, 1188
  - setIccProfile, 1188
  - setItemFacesMap, 1188
  - setXmpKeywords, 1189
  - setXmpSubCategories, 1189
  - setXmpSubjects, 1189
  - valueToString, 1189
  - VIDEOCOLORMODEL, 1183
  - videoColorModelToString, 1189
- Digikam::DMetadataSettings, 1190
  - instance, 1190
  - setSettings, 1190
  - settings, 1191
- Digikam::DMetadataSettingsContainer, 1191
- Digikam::DMetaInfoface, 1192
  - allAlbumItems, 1194
  - currentActiveItem, 1194
  - currentAlbumItems, 1194
  - currentSelectedItems, 1194
  - defaultUploadUrl, 1194
  - deleteImage, 1194
  - itemInfo, 1195
  - parseAlbumItemsRecursive, 1195
  - setItemInfo, 1195
  - slotDateTimeForUrl, 1195
  - slotMetadataChangedForUrl, 1195
  - supportAlbums, 1195
  - uploadUrl, 1195
  - uploadWidget, 1196
- Digikam::DModelFactory, 1196
- Digikam::DMultiTabBar, 1198
  - ActiveIconText, 1199
  - AllIconsText, 1199
  - appendButton, 1200
  - appendTab, 1200
  - button, 1200
  - isTabRaised, 1200
  - position, 1200
  - removeButton, 1201
  - removeTab, 1201
  - setPosition, 1201
  - setStyle, 1201
  - setTab, 1201
  - tab, 1202
  - tabStyle, 1202
  - TextStyle, 1199
- Digikam::DMultiTabBar::Private, 1202
- Digikam::DMultiTabBarButton, 1203
  - signalClicked, 1204
- Digikam::DMultiTabBarFrame, 1205
  - contentsMouseEvent, 1206
- Digikam::DMultiTabBarFrame::Private, 1206
- Digikam::DMultiTabBarTab, 1207
  - DMultiTabBarTab, 1209

- setPosition, 1209
- setState, 1209
- setStyle, 1209
- Digikam::DMultiTabBarTab::Private, 1210
- Digikam::DNGConvertSettings, 1210
- Digikam::DNGSettings, 1211
- Digikam::DNGWriter, 1212
  - ConvertError, 1212
  - DNG\_SDK\_INTERNAL\_ERROR, 1212
  - FILE\_NOT\_SUPPORTED, 1212
  - FULL\_SIZE, 1213
  - JPEGPreview, 1213
  - MEDIUM, 1213
  - NONE, 1213
  - PROCESS\_CANCELED, 1212
  - PROCESS\_COMPLETE, 1212
  - PROCESS\_CONTINUE, 1212
  - PROCESS\_FAILED, 1212
- Digikam::DNGWriter::Private, 1213
  - debugExtractedRAWData, 1214
- Digikam::DNGWriterHost, 1215
- Digikam::DNNBaseDetectorModel, 1216
  - detectObjects, 1217
  - generateObjects, 1217, 1218
  - getInputImageSize, 1218
  - getPredefinedClasses, 1218
  - uiConfidenceThreshold, 1218
- Digikam::DNNFaceDetectorBase, 1219
  - selectBbox, 1220
- Digikam::DNNFaceDetectorSSD, 1221
  - detectFaces, 1222
- Digikam::DNNFaceDetectorYOLO, 1223
  - detectFaces, 1224
- Digikam::DNNFaceDetectorYuNet, 1225
  - detectFaces, 1226
  - setFaceDetectionSize, 1226
- Digikam::DNNFaceExtractorBase, 1227
  - cosineDistance, 1228
  - getThreshold, 1228
  - loadModels, 1228
  - vectortomat, 1228
- Digikam::DNNModelBase, 1229
  - getModelPath, 1230
  - getThreshold, 1230
- Digikam::DNNModelConfig, 1230
- Digikam::DNNModelInfoContainer, 1231
- Digikam::DNNModelManager, 1233
  - getDownloadInformation, 1233
  - getModel, 1233
  - instance, 1234
- Digikam::DNNModelNet, 1234
- Digikam::DNNModelSFace, 1236
- Digikam::DNNModelYuNet, 1237
- Digikam::DNNOpenFaceExtractor, 1239
  - alignFace, 1240
  - getFaceEmbedding, 1240
  - getThreshold, 1240
  - loadModels, 1240
- Digikam::DNNResnetDetector, 1241
  - loadModels, 1242
- Digikam::DNNSFaceExtractor, 1243
  - alignFace, 1244
  - getFaceEmbedding, 1244
  - getThreshold, 1244
  - loadModels, 1244
- Digikam::DNNYoloDetector, 1245
  - loadModels, 1246
- Digikam::DNotificationPopup, 1247
  - ~DNotificationPopup, 1249
  - anchor, 1250
  - autoDelete, 1250
  - Balloon, 1249
  - Boxed, 1249
  - clicked, 1250
  - defaultLocation, 1250
  - DNotificationPopup, 1249
  - message, 1250–1254
  - moveNear, 1255
  - PopupStyle, 1249
  - positionSelf, 1255
  - setAnchor, 1255
  - setAutoDelete, 1255
  - setPopupStyle, 1256
  - setTimeout, 1256
  - setView, 1256
  - show, 1257
  - standardView, 1257
  - timeout, 1257
  - view, 1257
- Digikam::DNotificationWidget, 1258
  - ~DNotificationWidget, 1260
  - addAction, 1260
  - animatedHide, 1261
  - animatedShow, 1261
  - animatedShowTemporized, 1261
  - clearAllActions, 1261
  - DNotificationWidget, 1260
  - heightForWidth, 1261
  - hideAnimationFinished, 1262
  - icon, 1262
  - isCloseButtonVisible, 1262
  - isHideAnimationRunning, 1262
  - isShowAnimationRunning, 1262
  - linkActivated, 1263
  - linkHovered, 1263
  - MessageType, 1260
  - messageType, 1263
  - minimumSizeHint, 1264
  - removeAction, 1264
  - setCloseButtonVisible, 1264
  - setIcon, 1264
  - setMessageType, 1264
  - setText, 1265
  - setWordWrap, 1265
  - showAnimationFinished, 1265
  - sizeHint, 1266

- text, [1266](#)
- wordWrap, [1266](#)
- Digikam::DNotificationWidget::Private, [1267](#)
- Digikam::DOnlineTranslator, [1268](#)
  - detectLanguage, [1273](#)
  - DOnlineTranslator, [1272](#)
  - engineName, [1273](#)
  - error, [1273](#)
  - errorString, [1273](#)
  - fromRFC3066, [1273](#)
  - isRunning, [1273](#)
  - isSourceTranscriptionEnabled, [1274](#)
  - isSourceTranslitEnabled, [1274](#)
  - isSupportTranslation, [1274](#)
  - isTranslationOptionsEnabled, [1274](#)
  - isTranslationTranslitEnabled, [1275](#)
  - language, [1275](#)
  - languageCode, [1275](#)
  - languageName, [1276](#)
  - NetworkError, [1272](#)
  - NoError, [1272](#)
  - ParametersError, [1272](#)
  - ParsingError, [1272](#)
  - ServiceError, [1272](#)
  - setEngineApiKey, [1276](#)
  - setEngineUrl, [1276](#)
  - setSourceTranscriptionEnabled, [1277](#)
  - setSourceTranslitEnabled, [1277](#)
  - setTranslationOptionsEnabled, [1277](#)
  - setTranslationTranslitEnabled, [1277](#)
  - signalFinished, [1277](#)
  - source, [1277](#)
  - sourceLanguage, [1278](#)
  - sourceLanguageName, [1278](#)
  - sourceTranscription, [1278](#)
  - sourceTranslit, [1278](#)
  - supportedRFC3066, [1278](#)
  - toJson, [1278](#)
  - translate, [1279](#)
  - translation, [1279](#)
  - TranslationError, [1272](#)
  - translationLanguage, [1279](#)
  - translationLanguageName, [1279](#)
  - translationOptions, [1279](#)
  - translationTranslit, [1280](#)
- Digikam::DOnlineTranslator::Private, [1280](#)
  - s\_bingLanguageCodes, [1281](#)
  - s\_googleLanguageCodes, [1281](#)
  - s\_lingvaLanguageCodes, [1281](#)
  - s\_rfc3066LanguageCodesBing, [1282](#)
  - s\_rfc3066LanguageCodesGoogle, [1282](#)
  - s\_rfc3066LanguageCodesLingva, [1282](#)
  - s\_rfc3066LanguageCodesYandex, [1282](#)
  - s\_yandexLanguageCodes, [1282](#)
- Digikam::DOnlineTranslatorOption, [1283](#)
  - toJson, [1284](#)
- Digikam::DOnlineTts, [1284](#)
  - DOnlineTts, [1287](#)
    - Emotion, [1285](#)
    - emotion, [1287](#)
    - emotionCode, [1287](#)
    - error, [1288](#)
    - errorString, [1288](#)
    - generateUrls, [1288](#)
    - media, [1289](#)
    - NoError, [1287](#)
    - TtsError, [1285](#)
    - UnsupportedEmotion, [1287](#)
    - UnsupportedEngine, [1287](#)
    - UnsupportedLanguage, [1287](#)
    - UnsupportedVoice, [1287](#)
    - Voice, [1287](#)
    - voice, [1289](#)
    - voiceCode, [1289](#)
- Digikam::DOnlineTts::Private, [1290](#)
  - s\_emotionCodes, [1290](#)
  - s\_voiceCodes, [1290](#)
- Digikam::DownloadInfo, [1290](#)
  - hash, [1291](#)
  - name, [1291](#)
  - path, [1291](#)
  - size, [1291](#)
- Digikam::DownloadSettings, [1291](#)
- Digikam::DPixelsAliasFilter, [1292](#)
  - pixelAntiAliasing, [1292](#)
  - pixelAntiAliasing16, [1292](#)
- Digikam::DPlainTextEdit, [1293](#)
  - ~DPlainTextEdit, [1295](#)
  - acceptedCharacters, [1295](#)
  - DPlainTextEdit, [1295](#)
  - ignoredCharacters, [1295](#)
  - isClearButtonEnabled, [1295](#)
  - leftCharacters, [1296](#)
  - returnPressed, [1296](#)
  - setCurrentLanguage, [1296](#)
  - setLinesVisible, [1296](#)
  - setMaxLength, [1296](#)
  - spellCheckSettings, [1296](#)
  - text, [1297](#)
- Digikam::DPlainTextEdit::Private, [1297](#)
  - init, [1297](#)
- Digikam::DPlugin, [1298](#)
  - ~DPlugin, [1299](#)
  - categories, [1299](#)
  - cleanUp, [1299](#)
  - count, [1300](#)
  - DPlugin, [1299](#)
  - extraAboutData, [1300](#)
  - extraAboutDataRowTitles, [1300](#)
  - extraAboutDataTitle, [1300](#)
  - handbookChapter, [1300](#)
  - handbookReference, [1300](#)
  - handbookSection, [1300](#)
  - hasVisibilityProperty, [1301](#)
  - ifacelid, [1301](#)
  - name, [1301](#)

- pluginAuthors, [1301](#)
- setShouldLoaded, [1301](#)
- setup, [1301](#)
- setVisible, [1301](#)
- shouldLoaded, [1302](#)
- version, [1302](#)
- Digikam::DPluginAboutDlg, [1302](#)
- Digikam::DPluginAction, [1303](#)
  - ActionCategory, [1304](#)
  - ActionType, [1304](#)
  - actionType, [1304](#)
  - Editor, [1304](#)
  - EditorColors, [1304](#)
  - EditorDecorate, [1304](#)
  - EditorEnhance, [1304](#)
  - EditorFile, [1304](#)
  - EditorFilters, [1304](#)
  - EditorTransform, [1304](#)
  - Generic, [1304](#)
  - GenericExport, [1304](#)
  - GenericImport, [1304](#)
  - GenericMetadata, [1304](#)
  - GenericTool, [1304](#)
  - GenericView, [1304](#)
  - InvalidType, [1304](#)
  - pluginId, [1304](#)
  - setActionCategory, [1304](#)
  - toString, [1305](#)
  - xmlSection, [1305](#)
- Digikam::DPluginAuthor, [1305](#)
  - toString, [1305](#)
- Digikam::DPluginBqm, [1306](#)
  - ~DPluginBqm, [1308](#)
  - categories, [1308](#)
  - count, [1308](#)
  - DPluginBqm, [1308](#)
  - findToolByName, [1308](#)
  - hasVisibilityProperty, [1308](#)
  - ifacelid, [1309](#)
  - infoface, [1309](#)
  - setVisible, [1309](#)
  - tools, [1309](#)
- Digikam::DPluginConfView, [1310](#)
  - activated, [1311](#)
  - apply, [1311](#)
  - clearAll, [1311](#)
  - count, [1311](#)
  - DPluginConfView, [1311](#)
  - filter, [1311](#)
  - itemsVisible, [1311](#)
  - itemsWithVisiblyProperty, [1311](#)
  - selectAll, [1312](#)
  - setFilter, [1312](#)
  - signalSearchResult, [1312](#)
- Digikam::DPluginConfViewBqm, [1313](#)
  - loadPlugins, [1314](#)
- Digikam::DPluginConfViewDImg, [1315](#)
  - loadPlugins, [1316](#)
- Digikam::DPluginConfViewEditor, [1317](#)
  - loadPlugins, [1318](#)
- Digikam::DPluginConfViewGeneric, [1319](#)
  - loadPlugins, [1320](#)
- Digikam::DPluginDialog, [1321](#)
- Digikam::DPluginDImg, [1322](#)
  - ~DPluginDImg, [1324](#)
  - canRead, [1324](#)
  - canWrite, [1324](#)
  - categories, [1324](#)
  - count, [1324](#)
  - DPluginDImg, [1324](#)
  - exportWidget, [1324](#)
  - extraAboutData, [1325](#)
  - extraAboutDataRowTitles, [1325](#)
  - extraAboutDataTitle, [1325](#)
  - hasVisibilityProperty, [1325](#)
  - ifacelid, [1325](#)
  - loader, [1325](#)
  - loaderName, [1326](#)
  - previewSupported, [1326](#)
  - setVisible, [1326](#)
  - typeMimes, [1326](#)
- Digikam::DPluginEditor, [1327](#)
  - ~DPluginEditor, [1329](#)
  - actions, [1329](#)
  - categories, [1329](#)
  - count, [1329](#)
  - DPluginEditor, [1329](#)
  - findActionByName, [1329](#)
  - ifacelid, [1329](#)
  - infoface, [1330](#)
  - setVisible, [1330](#)
- Digikam::DPluginGeneric, [1331](#)
  - ~DPluginGeneric, [1333](#)
  - actions, [1333](#)
  - categories, [1333](#)
  - count, [1333](#)
  - DPluginGeneric, [1333](#)
  - findActionByName, [1333](#)
  - ifacelid, [1333](#)
  - infoface, [1334](#)
  - reactivateToolDialog, [1334](#)
  - setVisible, [1334](#)
- Digikam::DPluginLoader, [1334](#)
  - appendPluginToBlackList, [1336](#)
  - appendPluginToWhiteList, [1336](#)
  - canExport, [1337](#)
  - canImport, [1337](#)
  - cleanUp, [1337](#)
  - configGroupName, [1337](#)
  - exportWidget, [1337](#)
  - init, [1337](#)
  - instance, [1338](#)
  - registerEditorPlugins, [1338](#)
  - registerGenericPlugins, [1338](#)
  - registerRawImportPlugins, [1338](#)
- Digikam::DPluginLoader::Private, [1338](#)



- appendPlugin, 1339
- loadPlugins, 1339
- pluginEntriesList, 1339
- Digikam::DPluginRawImport, 1340
  - ~DPluginRawImport, 1342
  - categories, 1342
  - count, 1342
  - DPluginRawImport, 1342
  - getRawProgram, 1342
  - ifacelid, 1342
  - run, 1342
  - setVisible, 1343
  - signalDecodedImage, 1343
  - signalLoadRaw, 1343
- Digikam::DPluginSetup, 1343
- Digikam::DPointSelect, 1344
  - contentsRect, 1346
  - DPointSelect, 1345
  - drawContents, 1346
  - drawMarker, 1346
  - minimumSizeHint, 1346
  - setMarkerColor, 1346
  - setRange, 1346
  - setValues, 1347
  - setXValue, 1347
  - setYValue, 1347
  - valueChanged, 1347
  - valuesFromPosition, 1348
  - xValue, 1348
  - yValue, 1348
- Digikam::DPopupFrame, 1349
  - ~DPopupFrame, 1350
  - close, 1350
  - DPopupFrame, 1350
  - exec, 1350
  - hideEvent, 1350
  - keyPressEvent, 1351
  - popup, 1351
  - resizeEvent, 1351
  - setMainWidget, 1351
- Digikam::DPreviewImage, 1352
  - setSelectionArea, 1353
  - slotClearHighlight, 1353
  - slotSetHighlightArea, 1354
  - slotSetHighlightShown, 1354
  - slotSetSelection, 1354
- Digikam::DPreviewManager, 1356
  - setSelectionArea, 1357
- Digikam::DProgressDlg, 1358
- Digikam::DProgressWdg, 1359
  - progressCompleted, 1360
  - progressScheduled, 1360
  - progressStatusChanged, 1360
  - progressThumbnailChanged, 1360
  - signalProgressCanceled, 1360
- Digikam::DragDropModelImplementation, 1361
  - dragDropFlags, 1362
  - dragDropFlagsV2, 1362
  - DragDropModelImplementation, 1362
  - setDragDropHandler, 1362
  - supportedDropActions, 1363
- Digikam::DragDropViewImplementation, 1364
  - asView, 1365
  - dragDropHandler, 1365
  - dragEnterEvent, 1365
  - mapIndexForDragDrop, 1365
  - pixmapForDrag, 1365
- Digikam::DragHandle, 1366
- Digikam::DRawDecoder, 1367
  - ~DRawDecoder, 1368
  - cancel, 1369
  - checkToCancelWaitingData, 1369
  - decodeHalfRAWImage, 1369
  - decodeRAWImage, 1369
  - DRawDecoder, 1368
  - extractRAWData, 1370
  - librawUseGomp, 1370
  - librawVersion, 1370
  - loadEmbeddedPreview, 1370, 1371
  - loadFullImage, 1371
  - loadHalfPreview, 1371, 1372
  - loadRawPreview, 1372
  - m\_cancel, 1374
  - m\_decoderSettings, 1374
  - rawFileIdentify, 1372
  - rawFiles, 1373
  - rawFilesList, 1373
  - rawFilesVersion, 1373
  - setWaitingDataProgress, 1373
  - supportedCamera, 1373
- Digikam::DRawDecoder::Private, 1374
- Digikam::DRawDecoderSettings, 1374
  - ~DRawDecoderSettings, 1377
  - autoBrightness, 1377
  - blackPoint, 1377
  - brightness, 1378
  - customWhiteBalance, 1378
  - dcbEnhanceFI, 1378
  - dcbIterations, 1378
  - deadPixelMap, 1378
  - DecodingQuality, 1375
  - DontStretchPixels, 1378
  - DRawDecoderSettings, 1377
  - enableBlackPoint, 1378
  - enableWhitePoint, 1379
  - expoCorrection, 1379
  - expoCorrectionHighlight, 1379
  - expoCorrectionShift, 1379
  - fixColorsHighlights, 1379
  - halfSizeColorImage, 1379
  - InputColorSpace, 1376
  - inputColorSpace, 1379
  - inputProfile, 1380
  - medianFilterPasses, 1380
  - NoiseReduction, 1376
  - NRThreshold, 1380

- NRType, [1380](#)
- operator==, [1377](#)
- optimizeTimeLoading, [1377](#)
- OutputColorSpace, [1376](#)
- outputColorSpace, [1380](#)
- outputProfile, [1380](#)
- RAWQuality, [1380](#)
- RGBInterpolate4Colors, [1381](#)
- sixteenBitsImage, [1381](#)
- unclipColors, [1381](#)
- WhiteBalance, [1376](#)
- whiteBalance, [1381](#)
- whiteBalanceArea, [1381](#)
- whitePoint, [1381](#)
- Digikam::DRawDecoderWidget, [1382](#)
  - DRawDecoderWidget, [1384](#)
  - readSettings, [1384](#)
  - writeSettings, [1384](#)
- Digikam::DRawDecoding, [1385](#)
  - ~DRawDecoding, [1385](#)
  - bcg, [1386](#)
  - curvesAdjust, [1386](#)
  - decodingSettingsToXml, [1386](#)
  - DRawDecoding, [1385](#)
  - operator==, [1386](#)
  - optimizeTimeLoading, [1386](#)
  - postProcessingSettingsIsDirty, [1386](#)
  - rawPrm, [1387](#)
  - resetPostProcessingSettings, [1386](#)
  - wb, [1387](#)
- Digikam::DRawInfo, [1387](#)
  - ~DRawInfo, [1389](#)
  - ambientAcceleration, [1389](#)
  - ambientElevationAngle, [1389](#)
  - ambientHumidity, [1389](#)
  - ambientPressure, [1389](#)
  - ambientTemperature, [1389](#)
  - ambientWaterDepth, [1390](#)
  - aperture, [1390](#)
  - baselineExposure, [1390](#)
  - blackPoint, [1390](#)
  - blackPointCh, [1390](#)
  - cameraColorMatrix1, [1390](#)
  - cameraMult, [1390](#)
  - colorKeys, [1390](#)
  - dateTime, [1391](#)
  - daylightMult, [1391](#)
  - description, [1391](#)
  - DNGVersion, [1391](#)
  - DRawInfo, [1389](#)
  - exposureIndex, [1391](#)
  - exposureProgram, [1391](#)
  - exposureTime, [1391](#)
  - filterPattern, [1391](#)
  - firmware, [1392](#)
  - flashUsed, [1392](#)
  - focalLength, [1392](#)
  - fullSize, [1392](#)
  - hasIccProfile, [1392](#)
  - iccData, [1392](#)
  - imageID, [1392](#)
  - ImageOrientation, [1389](#)
  - imageSize, [1392](#)
  - isDecodable, [1393](#)
  - latitude, [1393](#)
  - leftMargin, [1393](#)
  - lensModel, [1393](#)
  - localizedCameraModel, [1393](#)
  - make, [1393](#)
  - meteringMode, [1393](#)
  - model, [1393](#)
  - orientation, [1394](#)
  - originalRawFileName, [1394](#)
  - outputSize, [1394](#)
  - owner, [1394](#)
  - pixelAspectRatio, [1394](#)
  - rawColors, [1394](#)
  - rawDataUniqueID, [1394](#)
  - rawImages, [1395](#)
  - sensitivity, [1395](#)
  - serialNumber, [1395](#)
  - software, [1395](#)
  - thumbnail, [1395](#)
  - thumbSize, [1395](#)
  - topMargin, [1395](#)
  - uniqueCameraModel, [1395](#)
  - whitePoint, [1396](#)
  - xmpData, [1396](#)
- Digikam::DSaveSettingsWidget, [1396](#)
- Digikam::DSelectedItem, [1398](#)
- Digikam::DSelector, [1399](#)
  - arrowDirection, [1400](#)
  - contentsRect, [1400](#)
  - drawArrow, [1401](#)
  - drawContents, [1401](#)
  - indent, [1401](#)
  - setArrowDirection, [1401](#)
  - setIndent, [1401](#)
- Digikam::DServiceInfo, [1402](#)
- Digikam::DServiceMenu, [1402](#)
  - getIconFromService, [1402](#)
  - runFiles, [1402](#)
  - servicesForOpenWith, [1403](#)
- Digikam::DSliderSpinBox, [1404](#)
  - setInternalValue, [1406](#)
  - valueString, [1406](#)
- Digikam::DSplashScreen, [1407](#)
- Digikam::DSqueezedClickLabel, [1408](#)
- Digikam::DTagListDrag, [1409](#)
- Digikam::DTextBrowser, [1410](#)
- Digikam::DTextEdit, [1411](#)
  - ~DTextEdit, [1413](#)
  - acceptedCharacters, [1413](#)
  - DTextEdit, [1412](#)
  - ignoredCharacters, [1413](#)
  - isClearButtonEnabled, [1413](#)

- leftCharacters, 1413
- returnPressed, 1413
- setCurrentLanguage, 1414
- setLinesVisible, 1414
- setMaxLength, 1414
- spellCheckSettings, 1414
- text, 1414
- Digikam::DTextEdit::Private, 1415
  - init, 1415
- Digikam::DTextEditClearButton, 1415
- Digikam::DTextLabelName, 1416
- Digikam::DTextLabelValue, 1418
- Digikam::DTextList, 1419
- Digikam::DToolTipStyleSheet, 1419
- Digikam::DTrash, 1420
  - deleteDirRecursively, 1420
  - deleteImage, 1421
  - extractJsonForItem, 1421
- Digikam::DTrashItemInfo, 1421
- Digikam::DTrashItemModel, 1422
  - append, 1423
  - changeThumbSize, 1423
  - isEmpty, 1424
  - loadItemsForCollection, 1424
  - pixmapForItem, 1424
  - refreshThumbnails, 1424
  - removeItems, 1425
- Digikam::DTrashItemsListingJob, 1426
- Digikam::DuplicatesFinder, 1428
  - DuplicatesFinder, 1431
- Digikam::DuplicatesProgressObserver, 1431
  - imageProcessed, 1432
  - isCanceled, 1432
- Digikam::DVBox, 1432
- Digikam::DWItemDelegate, 1434
  - blockedEventTypes, 1435
  - createItemWidgets, 1436
  - DWItemDelegate, 1435
  - focusedIndex, 1436
  - itemView, 1436
  - setBlockedEventTypes, 1437
  - updateItemWidgets, 1437
- Digikam::DWItemDelegatePool, 1438
  - DWItemDelegatePool, 1438
  - findWidgets, 1438
- Digikam::DWItemDelegatePoolPrivate, 1439
- Digikam::DWItemDelegatePrivate, 1440
- Digikam::DWizardDlg, 1441
- Digikam::DWizardPage, 1442
- Digikam::DWorkingPixmap, 1443
- Digikam::DXmlGuiWindow, 1444
  - allActions, 1446
  - cleanupActions, 1446
  - configFullScreenHideToolBarsEntry, 1446
  - createFullScreenAction, 1446
  - createHelpActions, 1446
  - createSettingsActions, 1446
  - createSidebarActions, 1446
  - customizedFullScreenMode, 1447
  - editKeyboardShortcuts, 1447
  - fullScreenIsActive, 1447
  - infoface, 1447
  - readFullScreenSettings, 1447
  - registerPluginsActions, 1447
  - setConfigGroupName, 1448
  - setFullScreenOptions, 1448
  - setUpIconTheme, 1448
  - showSideBars, 1448
  - showThumbBar, 1448
  - thumbbarVisibility, 1448
- Digikam::DXmlGuiWindow::Private, 1449
  - dirtyMainToolBar, 1449
  - fsOptions, 1449
  - fullScreenAction, 1449
  - fullScreenBtn, 1449
  - fullScreenHideSideBars, 1450
  - fullScreenHideStatusBar, 1450
  - fullScreenHideThumbBar, 1450
  - fullScreenHideToolBars, 1450
  - fullScreenParent, 1450
  - menubarVisibility, 1450
  - statusbarVisibility, 1450
  - thumbbarVisibility, 1450
  - toolbarsVisibility, 1451
- Digikam::DynamicLayout, 1451
- Digikam::DynamicThread, 1452
  - ~DynamicThread, 1453
  - DynamicThread, 1453
  - run, 1453
  - runningFlag, 1453
  - setPriority, 1454
  - shutDown, 1454
  - start, 1454
  - starting, 1454
  - stop, 1454
  - threadMutex, 1454
  - wait, 1455
- Digikam::DZoomBar, 1456
  - BarMode, 1457
  - NoPreviewZoomCtrl, 1458
  - PreviewZoomCtrl, 1458
  - ThumbsSizeCtrl, 1458
- Digikam::EditableSearchTreeView, 1459
  - ~EditableSearchTreeView, 1463
  - addCustomContextMenuActions, 1463
  - contextMenuTitle, 1464
  - EditableSearchTreeView, 1463
  - handleCustomContextMenuAction, 1464
- Digikam::EditorCore, 1465
  - convertToPixmap, 1467
  - getImgSelection, 1467
  - rotate90, 1468
  - setUndoImg, 1468
- Digikam::EditorCore::Private, 1468
- Digikam::EditorCore::Private::FileToSave, 1469
- Digikam::EditorStackView, 1470

- Digikam::EditorTool, 1472
  - init, 1474
  - setInitPreview, 1474
- Digikam::EditorTooliface, 1475
- Digikam::EditorToolSettings, 1476
- Digikam::EditorToolThreaded, 1478
  - analyser, 1481
  - deleteFilterInstance, 1481
  - filter, 1481
  - renderingMode, 1481
  - setProgressMessage, 1481
  - slotAnalyserStarted, 1481
  - slotFilterStarted, 1481
  - slotProgress, 1481
- Digikam::EditorWindow, 1482
  - registerExtraPluginsActions, 1487
  - saveDestinationUrl, 1487
  - toggleZoomActions, 1487
- Digikam::EditorWindow::Private, 1488
- Digikam::EffectMgr, 1489
  - EffectType, 1490
  - None, 1490
- Digikam::EffectMgr::Private, 1490
- Digikam::EffectPreview, 1491
- Digikam::Ellipsoid, 1491
  - CLARKE\_1866, 1493
  - createEllipsoid, 1493
  - createFlattenedSphere, 1493
  - eccentricity, 1494
  - Ellipsoid, 1492
  - GRS80, 1494
  - INTERNATIONAL\_1924, 1494
  - inverseFlattening, 1494
  - isIvfDefinitive, 1494
  - isSphere, 1494
  - m\_inverseFlattening, 1496
  - m\_ivfDefinitive, 1496
  - m\_semiMajorAxis, 1497
  - m\_semiMinorAxis, 1497
  - orthodromicDistance, 1495
  - radiusOfCurvature, 1495
  - semiMajorAxis, 1495
  - semiMinorAxis, 1496
  - SPHERE, 1496
  - WGS84, 1496
- Digikam::EmbossFilter, 1498
  - filterAction, 1501
  - filterIdentifier, 1501
  - readParameters, 1501
- Digikam::EmptyDTrashItemsJob, 1502
- Digikam::EmptyImageListProvider, 1504
  - atEnd, 1505
  - image, 1505
  - images, 1505
  - proceed, 1505
  - setImages, 1505
  - setUnpairedImages, 1505
  - size, 1505
- Digikam::EqualizeFilter, 1506
  - filterAction, 1509
  - filterIdentifier, 1509
  - readParameters, 1509
- Digikam::ExifMetaEngineMergeHelper, 1509
- Digikam::ExifToolBinary, 1511
- Digikam::ExifToolConfPanel, 1513
- Digikam::ExifToolErrorView, 1514
- Digikam::ExifToolListView, 1515
  - setGroupList, 1516
- Digikam::ExifToolListViewGroup, 1516
- Digikam::ExifToolListViewItem, 1517
- Digikam::ExifToolLoadingView, 1518
- Digikam::ExifToolParser, 1519
  - applyChanges, 1521
  - applyMetadataFile, 1522
  - changeTimestamps, 1522
  - copyTags, 1522
  - exifToolAvailable, 1522
  - ExifToolData, 1520
  - load, 1522
  - loadChunk, 1523
  - readableFormats, 1523
  - setOutputStream, 1523
  - tagsDatabase, 1523
  - tagsDbToOrderedMap, 1523
  - translateTags, 1523
  - translationsList, 1524
  - version, 1524
  - writableFormats, 1524
- Digikam::ExifToolParser::Private, 1524
  - actionString, 1525
- Digikam::ExifToolProcess, 1525
  - ~ExifToolProcess, 1528
  - Action, 1526
  - APPLY\_CHANGES, 1527
  - APPLY\_CHANGES\_EXV, 1527
  - APPLY\_METADATA\_FILE, 1527
  - CHANGE\_TIMESTAMP, 1527
  - command, 1528
  - COPY\_ALL, 1527
  - COPY\_EXIF, 1527
  - COPY\_ICC, 1527
  - COPY\_IPTC, 1527
  - COPY\_MAKERNOTES, 1527
  - COPY\_NONE, 1527
  - COPY\_TAGS, 1527
  - COPY\_XMP, 1527
  - CopyTagsSource, 1527
  - CREATE\_NEW\_GROUPS, 1528
  - CREATE\_NEW\_TAGS, 1528
  - exifToolAvailable, 1528
  - exifToolError, 1529
  - exifToolErrorString, 1529
  - exifToolsBusy, 1529
  - ExifToolProcess, 1528
  - getExifToolResult, 1529
  - initExifTool, 1529

- LOAD\_CHUNKS, [1527](#)
- LOAD\_METADATA, [1527](#)
- NO\_ACTION, [1527](#)
- READ\_FORMATS, [1527](#)
- RESTORE\_PREVIEW, [1527](#)
- ResultStatus, [1527](#)
- setExifToolProgram, [1529](#)
- shutDownExifTool, [1529](#)
- TAGS\_DATABASE, [1527](#)
- TRANS\_ALL\_EXIF, [1528](#)
- TRANS\_ALL\_IPTC, [1528](#)
- TRANS\_ALL\_XMP, [1528](#)
- TRANS\_TAGS, [1527](#)
- TranslateTagsOps, [1527](#)
- TRANSLATIONS\_LIST, [1527](#)
- VERSION\_STRING, [1527](#)
- waitForExifToolResult, [1530](#)
- WRITE\_EXISTING\_TAGS, [1528](#)
- WRITE\_FORMATS, [1527](#)
- WritingTagsMode, [1528](#)
- Digikam::ExifToolProcess::Private, [1530](#)
- Digikam::ExifToolProcess::Private::Command, [1531](#)
- Digikam::ExifToolProcess::Result, [1532](#)
- Digikam::ExifToolThread, [1532](#)
  - run, [1533](#)
- Digikam::ExifToolWidget, [1533](#)
- Digikam::ExifWidget, [1535](#)
  - getMetadataTitle, [1537](#)
  - getTagDescription, [1537](#)
  - getTagTitle, [1537](#)
  - loadFromURL, [1537](#)
- Digikam::ExposureDetector, [1538](#)
  - detect, [1539](#)
- Digikam::ExposureSettingsContainer, [1539](#)
  - exposureIndicatorMode, [1539](#)
- Digikam::FaceClassifier, [1540](#)
  - loadTrainingData, [1541](#)
  - predict, [1541](#)
  - retrain, [1541](#)
- Digikam::FaceClassifierBase, [1542](#)
- Digikam::FaceDb, [1543](#)
  - clearDNNTraining, [1544](#)
  - insertFaceVector, [1544](#)
  - integrityCheck, [1544](#)
  - removeFaceVector, [1544](#), [1546](#)
  - trainData, [1546](#)
  - vacuum, [1546](#)
- Digikam::FaceDb::Private, [1546](#)
- Digikam::FaceDbAccess, [1547](#)
  - FaceDbAccess, [1547](#)
  - setLastError, [1547](#)
- Digikam::FaceDbAccessUnlock, [1547](#)
  - FaceDbAccessUnlock, [1548](#)
- Digikam::FaceDbBackend, [1548](#)
  - initSchema, [1551](#)
- Digikam::FaceDbOperationGroup, [1551](#)
  - allowLift, [1552](#)
  - FaceDbOperationGroup, [1551](#)
  - lift, [1552](#)
  - resetTime, [1552](#)
- Digikam::FaceDbSchemaUpdater, [1552](#)
- Digikam::FaceDetector, [1553](#)
  - detectFaces, [1553](#)
  - FaceDetector, [1553](#)
  - recommendedImageSize, [1554](#)
  - setParameter, [1554](#)
- Digikam::FaceGroup, [1555](#)
  - aboutToSetInfo, [1557](#)
  - addFace, [1557](#)
  - closestItem, [1557](#)
  - FaceGroup, [1557](#)
  - hasUnconfirmed, [1557](#)
  - markAllAsIgnored, [1557](#)
  - rejectAll, [1558](#)
  - setAutoSuggest, [1558](#)
  - setInfo, [1558](#)
  - setShowOnHover, [1558](#)
  - setVisible, [1558](#)
- Digikam::FaceGroup::Private, [1559](#)
  - MaxFaceListSize, [1559](#)
  - MaxMouseDistance, [1559](#)
- Digikam::FaceItem, [1560](#)
- Digikam::FaceItemRetriever, [1563](#)
- Digikam::FacePipeline, [1564](#)
  - addManually, [1567](#)
  - cancel, [1567](#)
  - confirm, [1567](#)
  - editRegion, [1567](#)
  - editTag, [1567](#)
  - FilterMode, [1566](#)
  - NormalWrite, [1566](#)
  - OverwriteAllFaces, [1566](#)
  - OverwriteUnconfirmed, [1566](#)
  - plugDatabaseFilter, [1567](#)
  - process, [1568](#)
  - ReadConfirmedFaces, [1566](#)
  - ReadFacesForTraining, [1566](#)
  - ReadUnconfirmedFaces, [1566](#)
  - remove, [1568](#)
  - ScanAll, [1566](#)
  - setPriority, [1568](#)
  - shutDown, [1569](#)
  - SkipAlreadyScanned, [1566](#)
  - train, [1569](#)
  - WriteMode, [1566](#)
- Digikam::FacePipeline::Private, [1570](#)
- Digikam::FacePipelineBase, [1572](#)
  - enqueue, [1575](#)
  - FilterMode, [1574](#)
  - NormalWrite, [1575](#)
  - OverwriteAllFaces, [1575](#)
  - OverwriteUnconfirmed, [1575](#)
  - ScanAll, [1574](#)
  - ScanNew, [1574](#)
  - TrainAll, [1574](#)
  - TrainNew, [1574](#)

- TrainRemove, [1574](#)
- TrainReset, [1574](#)
- WriteMode, [1575](#)
- Digikam::FacePipelineDetect, [1576](#)
  - addMoreWorkers, [1579](#)
  - classifier, [1579](#)
  - extractor, [1579](#)
  - finder, [1579](#)
  - loader, [1579](#)
  - start, [1579](#)
  - trainer, [1579](#)
  - writer, [1579](#)
- Digikam::FacePipelineDetectRecognize, [1580](#)
  - addMoreWorkers, [1583](#)
  - classifier, [1583](#)
  - extractor, [1583](#)
  - finder, [1583](#)
  - loader, [1583](#)
  - start, [1583](#)
  - trainer, [1583](#)
  - writer, [1583](#)
- Digikam::FacePipelineEdit, [1584](#)
  - addMoreWorkers, [1587](#)
  - classifier, [1587](#)
  - extractor, [1587](#)
  - finder, [1587](#)
  - loader, [1588](#)
  - start, [1588](#)
  - trainer, [1588](#)
  - writer, [1588](#)
- Digikam::FacePipelineExtendedPackage, [1589](#)
- Digikam::FacePipelineFaceTagsIface, [1591](#)
  - Confirmed, [1593](#)
  - ForRecognition, [1593](#)
  - GivenAsArgument, [1593](#)
  - Role, [1593](#)
- Digikam::FacePipelineFaceTagsIfaceList, [1594](#)
- Digikam::FacePipelinePackage, [1595](#)
- Digikam::FacePipelinePackageBase, [1596](#)
- Digikam::FacePipelineRecognize, [1598](#)
  - addMoreWorkers, [1601](#)
  - classifier, [1601](#)
  - extractor, [1601](#)
  - finder, [1601](#)
  - loader, [1601](#)
  - start, [1601](#)
  - trainer, [1601](#)
  - writer, [1601](#)
- Digikam::FacePipelineReset, [1602](#)
  - addMoreWorkers, [1605](#)
  - classifier, [1605](#)
  - extractor, [1605](#)
  - finder, [1605](#)
  - loader, [1605](#)
  - start, [1605](#)
  - trainer, [1605](#)
  - writer, [1605](#)
- Digikam::FacePipelineRetrain, [1606](#)
  - addMoreWorkers, [1609](#)
  - classifier, [1609](#)
  - extractor, [1609](#)
  - finder, [1609](#)
  - loader, [1609](#)
  - start, [1609](#)
  - trainer, [1609](#)
  - writer, [1609](#)
- Digikam::FacePreprocessor, [1610](#)
- Digikam::FacePreviewLoader, [1611](#)
- Digikam::FaceRejectionOverlay, [1616](#)
  - checkIndex, [1619](#)
  - createButton, [1619](#)
  - setActive, [1619](#)
  - updateButton, [1619](#)
  - widgetEnterEvent, [1619](#)
  - widgetLeaveEvent, [1619](#)
- Digikam::FaceRejectionOverlayButton, [1620](#)
  - icon, [1621](#)
  - sizeHint, [1621](#)
  - updateToolTip, [1622](#)
- Digikam::FaceScanSettings, [1622](#)
  - AlreadyScannedHandling, [1623](#)
  - ClearAll, [1624](#)
  - detectAccuracy, [1625](#)
  - DetectAndRecognize, [1625](#)
  - FaceDetectionModel, [1624](#)
  - FaceDetectionSize, [1624](#)
  - FaceRecognitionModel, [1624](#)
  - OpenFace, [1624](#)
  - recognizeAccuracy, [1625](#)
  - RecognizeMarkedFaces, [1625](#)
  - RecognizeOnly, [1624](#)
  - Rescan, [1624](#)
  - RetrainAll, [1625](#)
  - ScanTask, [1624](#)
  - SFace, [1624](#)
  - Skip, [1624](#)
  - SSDMOBILENET, [1624](#)
  - YOLOv3, [1624](#)
  - YuNet, [1624](#)
- Digikam::FaceScanWidget, [1626](#)
  - doLoadState, [1627](#)
  - doSaveState, [1627](#)
- Digikam::FaceScanWidget::Private, [1628](#)
- Digikam::FacesDetector, [1629](#)
- Digikam::FacesEngine, [1632](#)
- Digikam::FaceTags, [1635](#)
  - allPersonNames, [1635](#)
  - allPersonPaths, [1635](#)
  - allPersonTags, [1635](#)
  - applyTagIdentityMapping, [1636](#)
  - ensureIsPerson, [1636](#)
  - faceNameForTag, [1636](#)
  - getOrCreateTagForIdentity, [1636](#)
  - getOrCreateTagForPerson, [1636](#)
  - isPerson, [1636](#)
  - personParentTag, [1637](#)

- tagForPerson, 1637
- Digikam::FaceTagsEditor, 1638
  - add, 1639
  - addNormalTag, 1639
  - changeRegion, 1640
  - changeSuggestedName, 1640
  - changeTag, 1640
  - confirmedEntry, 1640
  - confirmName, 1640
  - databaseFaces, 1641
  - faceCountForPersonInImage, 1641
  - getSuggestedNames, 1641
  - getTagRects, 1641
  - numberOfFaces, 1641
  - removeAllFaces, 1641
  - removeFace, 1642
  - removeNormalTag, 1642
  - rotateFaces, 1642
  - unconfirmedEntry, 1642
  - unconfirmedFaceTagsIfaces, 1642
  - unconfirmedNameFaceTagsIfaces, 1643
- Digikam::FaceTagsIface, 1644
  - attributeForType, 1646
  - attributesForFlags, 1646
  - fromListing, 1646
  - fromVariant, 1646
  - getAutodetectedPersonString, 1646
  - hash, 1646
  - removeFaceTraining, 1646
  - typeForAttribute, 1646
  - typeForId, 1647
- Digikam::FaceUtils, 1648
  - addNormalTag, 1650
  - faceRectToDisplayRect, 1650
  - hasBeenScanned, 1650
  - markAsScanned, 1650
  - removeNormalTag, 1650
  - removeNormalTags, 1651
  - storeThumbnails, 1651
  - toFaceTagsIfaces, 1651
  - writeUnconfirmedResults, 1651
- Digikam::FacialRecognitionWrapper, 1652
  - addIdentity, 1652
  - addIdentityAttributes, 1652
  - addIdentityDebug, 1653
  - allIdentities, 1653
  - clearAllTraining, 1653
  - clearTraining, 1653
  - deleteIdentities, 1653
  - deleteIdentity, 1653
  - findIdentity, 1654
  - integrityCheck, 1654
  - recognizeFaces, 1654
  - setParameter, 1654
  - train, 1654, 1655
  - vacuum, 1655
- Digikam::FacialRecognitionWrapper::Private, 1655
  - findByAttributes, 1656
  - identityContains, 1656
- Digikam::FFmpegBinary, 1657
- Digikam::FFmpegConfigHelper, 1659
  - getAudioCodecsProperties, 1659
  - getExtensionsProperties, 1659
  - getVideoCodecsProperties, 1660
- Digikam::FFmpegLauncher, 1661
  - encodeFrames, 1662
  - setSettings, 1662
  - soundTrackLength, 1662
  - supportedCodecs, 1662
  - supportedFormats, 1663
- Digikam::FieldQueryBuilder, 1663
- Digikam::FileActionItemInfoList, 1664
- Digikam::FileActionMngr, 1666
  - transform, 1667
- Digikam::FileActionMngr::Private, 1668
- Digikam::FileActionMngrDatabaseWorker, 1670
  - applyMetadata, 1672
  - assignColorLabel, 1672
  - assignPickLabel, 1672
  - assignRating, 1672
  - assignTags, 1672
  - copyAttributes, 1673
  - editGroup, 1673
  - removeTags, 1673
  - setExifOrientation, 1673
- Digikam::FileActionMngrFileWorker, 1674
  - transform, 1676
  - writeMetadata, 1676
  - writeMetadataToFiles, 1676
  - writeOrientationToFiles, 1676
- Digikam::FileActionProgress, 1677
- Digikam::FileActionProgressItemContainer, 1680
- Digikam::FileActionProgressItemCreator, 1681
- Digikam::FilePropertiesOption, 1682
  - parseOperation, 1683
- Digikam::FileReadLocker, 1684
- Digikam::FileReadWriteLockKey, 1684
- Digikam::FileSaveConflictBox, 1685
- Digikam::FileSaveOptionsBox, 1686
  - ~FileSaveOptionsBox, 1687
  - discoverFormat, 1687
  - FileSaveOptionsBox, 1687
  - FORMAT, 1686
  - NONE, 1687
- Digikam::FileSaveOptionsDlg, 1688
- Digikam::FilesDownloader, 1688
- Digikam::FileWorkerInterface, 1689
- Digikam::FileWriteLocker, 1691
- Digikam::FilmContainer, 1691
- Digikam::FilmContainer::ListItem, 1692
- Digikam::FilmContainer::Private, 1693
- Digikam::FilmFilter, 1694
  - filterAction, 1697
  - filterIdentifier, 1697
  - readParameters, 1697
- Digikam::FilmFilter::Private, 1697

- Digikam::FilmGrainContainer, [1697](#)
- Digikam::FilmGrainFilter, [1698](#)
  - FilmGrainFilter, [1701](#)
  - filterAction, [1701](#)
  - filterIdentifier, [1701](#)
  - readParameters, [1701](#)
- Digikam::FilmGrainSettings, [1702](#)
- Digikam::FilmProfile, [1702](#)
- Digikam::Filter, [1703](#)
- Digikam::FilterAction, [1704](#)
  - Category, [1705](#)
  - ComplexFilter, [1706](#)
  - description, [1706](#)
  - DocumentedHistory, [1706](#)
  - ExplicitBranch, [1706](#)
  - Flag, [1706](#)
  - hasParameters, [1706](#)
  - identifier, [1706](#)
  - parameter, [1706](#), [1707](#)
  - ReproducibleFilter, [1706](#)
  - version, [1707](#)
- Digikam::FilterActionFilter, [1708](#)
  - appliedFilterActions, [1711](#)
  - completelyApplied, [1711](#)
  - filterAction, [1711](#)
  - FilterActionFilter, [1711](#)
  - filterIdentifier, [1712](#)
  - filterImage, [1712](#)
  - isComplexAction, [1712](#)
  - isReproducible, [1712](#)
  - isSupported, [1712](#)
  - readParameters, [1712](#)
  - setContinueOnError, [1712](#)
  - setFilterActions, [1713](#)
- Digikam::FiltersHistoryWidget, [1713](#)
- Digikam::FilterSideBarWidget, [1714](#)
  - ~FilterSideBarWidget, [1716](#)
  - doLoadState, [1717](#)
  - doSaveState, [1717](#)
  - FilterSideBarWidget, [1716](#)
  - setConfigGroup, [1717](#)
  - signalTagFilterChanged, [1717](#)
  - slotResetFilters, [1718](#)
- Digikam::FilterStatusBar, [1718](#)
- Digikam::FindDuplicatesAlbum, [1719](#)
- Digikam::FindDuplicatesAlbumItem, [1720](#)
  - calculateInfos, [1721](#)
  - itemCount, [1721](#)
- Digikam::FindDuplicatesView, [1721](#)
- Digikam::FingerPrintsGenerator, [1723](#)
  - FingerPrintsGenerator, [1726](#)
  - setUseMultiCoreCPU, [1726](#)
- Digikam::FingerprintsTask, [1727](#)
- Digikam::FirstRunDlg, [1728](#)
- Digikam::FocusPoint, [1729](#)
  - FocusPoint, [1730](#)
  - getRectBySize, [1730](#)
  - Inactive, [1729](#)
  - InFocus, [1729](#)
  - operator=, [1730](#)
  - Selected, [1729](#)
  - SelectedInFocus, [1729](#)
  - setCenterPosition, [1730](#)
  - setType, [1730](#)
  - TypePoint, [1729](#)
- Digikam::FocusPointGroup, [1731](#)
  - setInfo, [1732](#)
  - setVisible, [1732](#)
- Digikam::FocusPointGroup::Private, [1733](#)
- Digikam::FocusPointItem, [1734](#)
- Digikam::FocusPointsExtractor, [1737](#)
  - ListAFPoints, [1738](#)
- Digikam::FocusPointsWriter, [1738](#)
- Digikam::FrameOsd, [1738](#)
  - insertMessageOsdToFrame, [1739](#)
  - insertOsdToFrame, [1739](#)
  - populateOSD, [1739](#)
  - printComments, [1739](#)
  - printTags, [1739](#)
- Digikam::FrameOsdSettings, [1740](#)
  - osdFont, [1740](#)
  - readSettings, [1740](#)
- Digikam::FrameOsdWidget, [1741](#)
- Digikam::FrameUtils, [1741](#)
- Digikam::FreeRotationContainer, [1741](#)
- Digikam::FreeRotationFilter, [1743](#)
  - filterAction, [1746](#)
  - filterIdentifier, [1746](#)
  - readParameters, [1746](#)
- Digikam::FreeRotationSettings, [1746](#)
- Digikam::FreeSpaceToolTip, [1748](#)
  - repositionRect, [1749](#)
  - tipContents, [1749](#)
- Digikam::FreeSpaceWidget, [1750](#)
- Digikam::FullObjectDetection, [1751](#)
- Digikam::FullScreenSettings, [1752](#)
- Digikam::FuzzySearchSideBarWidget, [1753](#)
  - applySettings, [1755](#)
  - changeAlbumFromHistory, [1755](#)
  - doLoadState, [1755](#)
  - doSaveState, [1755](#)
  - getCaption, [1755](#)
  - getIcon, [1755](#)
  - setActive, [1756](#)
- Digikam::FuzzySearchView, [1757](#)
  - doLoadState, [1759](#)
  - doSaveState, [1759](#)
  - setConfigGroup, [1759](#)
- Digikam::FuzzySearchView::Private, [1759](#)
- Digikam::GeoCoordinates, [1760](#)
- Digikam::GeodeticCalculator, [1761](#)
  - a01, [1767](#)
  - azimuth, [1763](#)
  - checkAzimuth, [1763](#)
  - checkLatitude, [1763](#)
  - checkLongitude, [1764](#)



- checkOrthodromicDistance, [1764](#)
- computeDestinationPoint, [1764](#)
- computeDirection, [1764](#)
- destinationGeographicPoint, [1765](#)
- ellipsoid, [1765](#)
- fo, [1767](#)
- GeodeticCalculator, [1762](#)
- m\_A, [1767](#)
- m\_destinationValid, [1767](#)
- m\_directionValid, [1767](#)
- m\_distance, [1768](#)
- m\_eccentricitySquared, [1768](#)
- m\_ellipsoid, [1768](#)
- m\_lat1, [1768](#)
- m\_lat2, [1768](#)
- m\_maxOrthodromicDistance, [1768](#)
- m\_semiMajorAxis, [1768](#)
- m\_semiMinorAxis, [1769](#)
- m\_TOLERANCE\_0, [1769](#)
- m\_TOLERANCE\_CHECK, [1769](#)
- meridianArcLength, [1765](#)
- meridianArcLengthRadians, [1765](#)
- orthodromicDistance, [1766](#)
- setDestinationGeographicPoint, [1766](#)
- setDirection, [1766](#)
- setStartingGeographicPoint, [1767](#)
- T1, [1769](#)
- Digikam::GeoDragDropHandler, [1770](#)
- Digikam::GeofaceCluster, [1770](#)
- Digikam::GeofaceGlobalObject, [1771](#)
- Digikam::GeofaceInternalWidgetInfo, [1773](#)
- Digikam::GeofaceSharedData, [1774](#)
  - hasRegionSelection, [1775](#)
- Digikam::GeolocationFilter, [1776](#)
- Digikam::GeolocationSettings, [1777](#)
  - applySettingsToWidget, [1778](#)
  - instance, [1778](#)
  - mainMarbleWidget, [1778](#)
  - registerWidget, [1778](#)
  - setSettings, [1778](#)
  - settings, [1778](#)
  - unregisterWidget, [1778](#)
- Digikam::GeolocationSettingsContainer, [1779](#)
- Digikam::GeoModelHelper, [1780](#)
  - bestRepresentativeIndexFromList, [1781](#)
  - itemCoordinates, [1781](#)
  - itemIcon, [1782](#)
  - model, [1782](#)
  - onIndicesClicked, [1782](#)
  - pixmapFromRepresentativeIndex, [1782](#)
  - selectionModel, [1783](#)
- Digikam::GeoPluginAboutDlg, [1783](#)
- Digikam::GPCamera, [1784](#)
  - cameraAbout, [1786](#)
  - cameraDriverType, [1786](#)
  - cameraManual, [1787](#)
  - cameraMD5ID, [1787](#)
  - cameraSummary, [1787](#)
- cancel, [1787](#)
- capture, [1787](#)
- deleteItem, [1787](#)
- doConnect, [1787](#)
- downloadItem, [1788](#)
- getFolders, [1788](#)
- getFreeSpace, [1788](#)
- getItemInfo, [1788](#)
- getItemsInfoList, [1788](#)
- getMetadata, [1788](#)
- getPreview, [1789](#)
- getThumbnail, [1789](#)
- setLockItem, [1789](#)
- uploadItem, [1789](#)
- Digikam::GPSBookmarkModelHelper, [1790](#)
  - itemCoordinates, [1791](#)
  - itemFlags, [1791](#)
  - itemIcon, [1792](#)
  - model, [1792](#)
  - modelFlags, [1792](#)
  - selectionModel, [1792](#)
  - snapItemsTo, [1792](#)
- Digikam::GPSBookmarkOwner, [1793](#)
- Digikam::GPSCorrelatorWidget, [1794](#)
- Digikam::GPSDataContainer, [1795](#)
- Digikam::GPSDBJobInfo, [1796](#)
- Digikam::GPSDBJobsThread, [1798](#)
  - GPSTListing, [1799](#)
- Digikam::GPSGeofaceModelHelper, [1800](#)
  - bestRepresentativeIndexFromList, [1802](#)
  - itemCoordinates, [1802](#)
  - model, [1802](#)
  - modelFlags, [1802](#)
  - onIndicesMoved, [1802](#)
  - pixmapFromRepresentativeIndex, [1802](#)
  - selectionModel, [1802](#)
- Digikam::GPSItemContainer, [1803](#)
  - getTagList, [1805](#)
  - isTagListDirty, [1805](#)
  - loadImageData, [1805](#)
  - restoreRGTagList, [1805](#)
  - saveChanges, [1805](#)
  - setTagList, [1805](#)
  - writeLocations, [1806](#)
  - writeTagsToXmp, [1806](#)
- Digikam::GPSItemDelegate, [1806](#)
- Digikam::GPSItemInfo, [1807](#)
- Digikam::GPSItemInfoSorter, [1808](#)
- Digikam::GPSItemList, [1809](#)
- Digikam::GPSItemListContextMenu, [1810](#)
- Digikam::GPSItemListDragDropHandler, [1812](#)
  - createMimeData, [1812](#)
- Digikam::GPSItemModel, [1813](#)
- Digikam::GPSItemSortProxyModel, [1814](#)
- Digikam::GPSJob, [1815](#)
- Digikam::GPSTLinkItemSelectionModel, [1817](#)
- Digikam::GPSMarkerTiler, [1818](#)
  - bestRepresentativeIndexFromList, [1821](#)

- getGlobalGroupState, [1821](#)
- getTile, [1821](#)
- getTileGroupState, [1822](#)
- getTileMarkerCount, [1822](#)
- getTileRepresentativeMarker, [1822](#)
- getTileSelectedCount, [1822](#)
- GPSTiler, [1821](#)
- indicesEqual, [1822](#)
- onIndicesClicked, [1823](#)
- pixmapFromRepresentativeIndex, [1823](#)
- prepareTiles, [1823](#)
- regenerateTiles, [1824](#)
- setActive, [1824](#)
- setPositiveFilterIsActive, [1824](#)
- slotNewModelData, [1824](#)
- tileNew, [1824](#)
- Digikam::GPSModelIndexProxyMapper, [1825](#)
- isConnected, [1827](#)
- mapLeftToRight, [1826](#)
- mapRightToLeft, [1826](#)
- mapSelectionLeftToRight, [1826](#)
- mapSelectionRightToLeft, [1826](#)
- Digikam::GPSSearchSideBarWidget, [1828](#)
- applySettings, [1830](#)
- changeAlbumFromHistory, [1830](#)
- doLoadState, [1830](#)
- doSaveState, [1830](#)
- getCaption, [1830](#)
- getIcon, [1830](#)
- setActive, [1831](#)
- Digikam::GPSSearchView, [1832](#)
- doLoadState, [1834](#)
- doSaveState, [1834](#)
- GPSSearchView, [1833](#)
- setActive, [1834](#)
- setConfigGroup, [1834](#)
- Digikam::GPSUndoCommand, [1835](#)
- Digikam::GPSUndoCommand::UndoInfo, [1836](#)
- Digikam::Graph< VertexProperties, EdgeProperties >, [1837](#)
- AdjacencyFlags, [1841](#)
- copyProperties, [1841](#)
- edgeDifference, [1841](#)
- EdgesToLeaf, [1841](#)
- findZeroDegree, [1841](#)
- getGraph, [1841](#)
- graph\_traits, [1841](#)
- leaves, [1842](#)
- listPath, [1842](#)
- longestPathTouching, [1842](#)
- mostRemoteNodes, [1842](#)
- roots, [1842](#)
- rootsOf, [1842](#)
- shortestDistancesFrom, [1843](#)
- shortestPath, [1843](#)
- toList, [1843](#)
- topologicalSort, [1843](#)
- transitiveClosure, [1843](#)
- transitiveReduction, [1844](#)
- verticesBreadthFirst, [1844](#)
- verticesDepthFirstSorted, [1844](#)
- verticesDominatedBy, [1844](#)
- verticesDominatedByDepthFirstSorted, [1845](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::DominatorTree, [1845](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::Edge, [1845](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch, [1846](#)
- depth\_first\_search\_sorted, [1846](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::BreadthFirstSearchVisitor, [1847](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::CommonVisitor, [1848](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::DepthFirstSearchVisitor, [1849](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::GraphSearch::lessThanMapEdgeToTarget< GraphType, VertexLessThan >, [1850](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::Path, [1850](#)
- longestPath, [1850](#)
- shortestPath, [1850](#)
- Digikam::Graph< VertexProperties, EdgeProperties >::Vertex, [1851](#)
- Digikam::GraphicsDImgItem, [1852](#)
- setImage, [1853](#)
- Digikam::GraphicsDImgView, [1854](#)
- item, [1856](#)
- previewItem, [1856](#)
- scrollPointOnPoint, [1856](#)
- setItem, [1856](#)
- Digikam::GreycstorationContainer, [1856](#)
- Digikam::GreycstorationFilter, [1858](#)
- cancelFilter, [1862](#)
- filterAction, [1862](#)
- filterIdentifier, [1862](#)
- GreycstorationFilter, [1861](#)
- MODE, [1861](#)
- readParameters, [1862](#)
- SimpleResize, [1861](#)
- Digikam::GreycstorationSettings, [1863](#)
- Digikam::GroupedImagesFinder, [1863](#)
- GroupedImagesFinder, [1864](#)
- Digikam::GroupIndicatorOverlay, [1865](#)
- checkIndex, [1867](#)
- createWidget, [1867](#)
- setActive, [1868](#)
- slotEntered, [1868](#)
- visualChange, [1868](#)
- Digikam::GroupIndicatorOverlayWidget, [1869](#)
- Digikam::GroupingViewImplementation, [1870](#)
- hasHiddenGroupedImages, [1871](#)
- Digikam::GroupItemFilterSettings, [1871](#)

- matches, [1871](#)
- setAllOpen, [1871](#)
- setOpen, [1871](#)
- Digikam::GroupStateComputer, [1872](#)
- Digikam::Haar::Calculator, [1872](#)
  - calcHaar, [1872](#)
  - transform, [1872](#)
- Digikam::Haar::ImageData, [1873](#)
  - fillImageData, [1873](#)
- Digikam::Haar::SignatureData, [1873](#)
  - avg, [1873](#)
  - sig, [1873](#)
- Digikam::Haar::SignatureMap, [1874](#)
- Digikam::Haar::WeightBin, [1874](#)
  - m\_bin, [1875](#)
  - WeightBin, [1874](#)
- Digikam::Haar::Weights, [1875](#)
- Digikam::HaarIface, [1875](#)
  - bestMatchesForImageWithThreshold, [1877](#)
  - ExcludeFolder, [1876](#)
  - findDuplicates, [1877](#)
  - fulfillsRestrictions, [1877](#)
  - getBestAndWorstPossibleScore, [1878](#)
  - imagesFromAlbumsAndTags, [1878](#)
  - indexImage, [1878](#)
  - loadQImage, [1878](#)
  - NewerCreationDate, [1876](#)
  - NewerModificationDate, [1876](#)
  - OlderOrLarger, [1876](#)
  - PreferFolder, [1876](#)
  - rebuildDuplicatesAlbums, [1879](#)
  - ReflImageSelMethod, [1876](#)
  - retrieveSignatureFromDB, [1879](#)
  - setAlbumRootsToSearch, [1879](#)
  - signatureAsText, [1879](#)
- Digikam::HaarIface::Private, [1879](#)
- Digikam::HaarProgressObserver, [1880](#)
- Digikam::HidingStateChanger, [1881](#)
  - finished, [1884](#)
  - HidingStateChanger, [1883](#)
  - stateChanged, [1884](#)
- Digikam::Highlighter, [1884](#)
- Digikam::HistogramBox, [1885](#)
- Digikam::HistogramPainter, [1886](#)
  - ~HistogramPainter, [1887](#)
  - disableHistogramGuide, [1887](#)
  - enableHistogramGuideByColor, [1887](#)
  - HistogramPainter, [1887](#)
  - initFrom, [1888](#)
  - render, [1888](#)
  - setChannelType, [1888](#)
  - setHighlightSelection, [1889](#)
  - setHistogram, [1889](#)
  - setRenderXGrid, [1889](#)
  - setScale, [1889](#)
  - setSelection, [1889](#)
- Digikam::HistogramWidget, [1891](#)
  - currentHistogram, [1893](#)
  - HistogramWidget, [1892](#)
  - stopHistogramComputation, [1893](#)
  - updateData, [1893](#)
  - updateSelectionData, [1893](#)
- Digikam::HistoryEdgeProperties, [1893](#)
- Digikam::HistoryImageId, [1894](#)
  - Current, [1895](#)
  - HistoryImageId, [1895](#)
  - Intermediate, [1895](#)
  - m\_originalUUID, [1895](#)
  - m\_uuid, [1895](#)
  - Original, [1895](#)
  - Source, [1895](#)
  - Type, [1895](#)
- Digikam::HistoryVertexProperties, [1896](#)
- Digikam::HotPixelContainer, [1896](#)
- Digikam::HotPixelFixer, [1898](#)
  - filterAction, [1901](#)
  - filterIdentifier, [1901](#)
  - readParameters, [1901](#)
- Digikam::HotPixelProps, [1901](#)
  - operator==, [1902](#)
- Digikam::HotPixelSettings, [1902](#)
- Digikam::HotPixelsWeights, [1903](#)
- Digikam::HoverButtonDelegateOverlay, [1904](#)
  - createButton, [1906](#)
  - createWidget, [1906](#)
  - setActive, [1906](#)
  - updateButton, [1906](#)
  - visualChange, [1906](#)
- Digikam::HSLContainer, [1907](#)
- Digikam::HSLFilter, [1908](#)
  - filterAction, [1911](#)
  - filterIdentifier, [1911](#)
  - readParameters, [1911](#)
- Digikam::HSLSettings, [1911](#)
- Digikam::HSPreviewWidget, [1912](#)
- Digikam::HTMLWidget, [1913](#)
- Digikam::HTMLWidgetPage, [1914](#)
- Digikam::IccManager, [1915](#)
  - displaySoftProofingTransform, [1917](#)
  - IccManager, [1916](#)
  - imageProfile, [1917](#)
  - isSRGB, [1917](#)
  - needsPostLoadingManagement, [1917](#)
  - transform, [1917](#)
  - transformDefault, [1917](#)
  - transformForDisplay, [1917](#), [1918](#)
  - transformForOutput, [1918](#)
  - transformToSRGB, [1918](#)
- Digikam::IccPostLoadingManager, [1919](#)
  - IccPostLoadingManager, [1921](#)
  - postLoadingManage, [1921](#)
- Digikam::ICCPreviewWidget, [1921](#)
- Digikam::IccProfile, [1922](#)
  - close, [1923](#)
  - data, [1923](#)
  - defaultProfiles, [1923](#)

- defaultSearchPaths, [1924](#)
- description, [1924](#)
- Display, [1923](#)
- filePath, [1924](#)
- handle, [1924](#)
- lccProfile, [1923](#)
- Input, [1923](#)
- InvalidType, [1923](#)
- isOpen, [1924](#)
- isSameProfileAs, [1924](#)
- open, [1924](#)
- operator==, [1925](#)
- Output, [1923](#)
- ProfileType, [1922](#)
- sRGB, [1925](#)
- type, [1925](#)
- writeToFile, [1925](#)
- Digikam::ICCPProfileInfoDlg, [1925](#)
- Digikam::lccProfilesComboBox, [1926](#)
  - addProfileSqueezed, [1927](#)
  - addProfilesSqueezed, [1927](#)
  - currentProfile, [1928](#)
  - lccProfilesComboBox, [1927](#)
  - replaceProfilesSqueezed, [1928](#)
  - setCurrentProfile, [1928](#)
  - setNoProfileIfEmpty, [1928](#)
- Digikam::lccProfilesMenuAction, [1929](#)
  - addProfile, [1930](#)
  - addProfiles, [1930](#)
  - disableIfEmpty, [1930](#)
  - parentObject, [1930](#)
  - replaceProfiles, [1930](#)
- Digikam::lccProfilesSettings, [1931](#)
- Digikam::ICCPProfileWidget, [1933](#)
  - getMetadataTitle, [1935](#)
  - getTagDescription, [1935](#)
  - getTagTitle, [1935](#)
  - loadFromURL, [1935](#)
- Digikam::lccRenderingIntentComboBox, [1936](#)
- Digikam::lccSettings, [1937](#)
  - displayProfiles, [1938](#)
  - inputProfiles, [1938](#)
  - instance, [1938](#)
  - isEnabled, [1938](#)
  - loadAllProfilesProperties, [1938](#)
  - monitorProfile, [1939](#)
  - monitorProfileFromSystem, [1939](#)
  - outputProfiles, [1939](#)
  - profilesForDescription, [1939](#)
  - setSettings, [1939](#)
  - settings, [1939](#)
  - setUseManagedView, [1939](#)
  - useManagedPreviews, [1940](#)
  - workspaceProfiles, [1940](#)
- Digikam::lccSettings::Private, [1940](#)
- Digikam::ICCSettingsContainer, [1940](#)
  - BehaviorEnum, [1941](#)
  - InvalidBehavior, [1941](#)
  - KeepProfile, [1941](#)
  - LeaveFileUntagged, [1941](#)
  - PreserveEmbeddedProfile, [1941](#)
  - UseEmbeddedProfile, [1941](#)
- Digikam::lccTransform, [1942](#)
  - apply, [1942](#)
  - close, [1943](#)
  - effectiveInputProfile, [1943](#)
  - embeddedProfile, [1943](#)
  - init, [1943](#)
  - setDoNotEmbedOutputProfile, [1943](#)
  - setEmbeddedProfile, [1943](#)
  - setIntent, [1943](#)
  - setOutputProfile, [1944](#)
  - setProofProfile, [1944](#)
  - willHaveEffect, [1944](#)
- Digikam::lccTransformFilter, [1945](#)
  - filterAction, [1948](#)
  - filterIdentifier, [1948](#)
  - filterImage, [1948](#)
  - parametersSuccessfullyRead, [1948](#)
  - progressInfo, [1949](#)
  - readParameters, [1949](#)
  - readParametersError, [1949](#)
- Digikam::Identity, [1949](#)
  - attribute, [1950](#)
  - attributesMap, [1950](#)
  - id, [1950](#)
  - Identity, [1950](#)
- Digikam::IdentityProvider, [1950](#)
  - addIdentity, [1951](#)
  - addIdentityDebug, [1951](#)
  - addTraining, [1951](#)
  - allIdentities, [1951](#)
  - clearAllTraining, [1951](#)
  - clearTraining, [1952](#)
  - deleteIdentities, [1952](#)
  - deleteIdentity, [1952](#)
  - findIdentity, [1952](#)
  - getTrainingData, [1952](#)
  - integrityCheck, [1952](#)
  - isValidId, [1953](#)
  - vacuum, [1953](#)
- Digikam::ImageChangeset, [1953](#)
  - ImageChangeset, [1953](#)
- Digikam::ImageCommonContainer, [1954](#)
- Digikam::ImageCurves, [1954](#)
  - channelToBinary, [1955](#)
  - CURVE\_FREE, [1955](#)
  - CURVE\_SMOOTH, [1955](#)
  - CurveType, [1955](#)
  - fillFromOtherCurves, [1956](#)
  - getContainer, [1956](#)
  - isLinear, [1956](#)
  - MULTIPLIER\_16BIT, [1957](#)
  - NUM\_CHANNELS, [1957](#)
  - NUMBER\_OF\_POINTS, [1957](#)
  - setChannelFromBinary, [1956](#)

- Digikam::ImageDialog, 1958
- Digikam::ImageDialog::Private, 1959
- Digikam::ImageDialogIconProvider, 1959
- Digikam::ImageDialogPreview, 1960
- Digikam::ImageDialogToolTip, 1961
- Digikam::ImageGuideWidget, 1963
- Digikam::ImageHistogram, 1965
  - calculate, 1967
  - calculationAboutToStart, 1967
  - calculationStarted, 1967
  - isSixteenBit, 1967
  - run, 1967
  - stopCalculation, 1967
- Digikam::ImageHistoryEntry, 1968
- Digikam::ImageIface, 1968
  - colorInfoFromOriginal, 1969
  - convertOriginalColorDepth, 1969
  - convertToPixmap, 1969
  - crop, 1969
  - FullImage, 1969
  - ImageIface, 1969
  - ImageSelection, 1969
  - original, 1970
  - originalIccProfile, 1970
  - originalSize, 1970
  - paint, 1970
  - preview, 1970
  - previewReference, 1970
  - previewSize, 1970
  - PreviewType, 1969
  - selection, 1971
  - selectionRect, 1971
  - setOriginal, 1971
  - setOriginalIccProfile, 1971
  - setPreview, 1971
  - setPreviewIccProfile, 1971
  - setPreviewSize, 1972
  - setPreviewType, 1972
  - setSelection, 1972
- Digikam::ImageLevels, 1972
  - levelsChannelReset, 1973
  - saveLevelsToGimpLevelsFile, 1973
  - setLevelGammaValue, 1973
- Digikam::ImageListProvider, 1974
- Digikam::ImageMetadataContainer, 1975
- Digikam::ImagePreviewItem, 1976
- Digikam::ImageQualityCalculator, 1977
- Digikam::ImageQualityCalculator::ResultDetection, 1978
- Digikam::ImageQualityConfSelector, 1978
  - CustomSettings, 1979
  - GlobalSettings, 1979
  - SettingsType, 1979
- Digikam::ImageQualityContainer, 1979
- Digikam::ImageQualityParser, 1980
  - ImageQualityParser, 1981
  - startAnalyse, 1981
- Digikam::ImageQualityParser::Private, 1981
- Digikam::ImageQualitySettings, 1982
- Digikam::ImageQualitySorter, 1983
  - AllItems, 1986
  - ImageQualitySorter, 1986
  - NonAssignedItems, 1986
  - QualityScanMode, 1985
  - setUseMultiCoreCPU, 1986
- Digikam::ImageQualityTask, 1987
- Digikam::ImageQualityThread, 1988
- Digikam::ImageQualityThreadPool, 1989
- Digikam::ImageRegionItem, 1990
- Digikam::ImageRegionWidget, 1992
  - getOriginalImageRegionToRender, 1994
  - getOriginalRegionImage, 1994
- Digikam::ImageRelation, 1994
- Digikam::ImageSortFilterModel, 1995
  - imageFilterModel, 1997
  - imageInfosSorted, 1997
  - mapListToSource, 1997
  - setDirectSourceItemModel, 1997
- Digikam::ImageTagChangeset, 1997
  - Operation, 1998
  - operator<<, 1998
- Digikam::ImageTagProperty, 1998
- Digikam::ImageTagPropertyName, 1999
- Digikam::ImageWindow, 2000
  - infoIface, 2005
  - versionManager, 2005
- Digikam::ImageWindow::Private, 2006
- Digikam::ImageZoomSettings, 2006
  - fitToSize, 2007
  - fitToSizeZoomFactor, 2007
  - imageSize, 2007
  - mapImageToZoom, 2008
  - mapZoomToImage, 2008
  - originalImageSize, 2008
  - realZoomFactor, 2008
  - setDisplayWidget, 2008
  - setImageSize, 2008
  - setImageSmoothScale, 2009
  - setZoomFactor, 2009
  - snappedZoomFactor, 2009
  - snappedZoomStep, 2009
  - sourceRect, 2009
  - zoomedSize, 2009
  - zoomFactor, 2010
- Digikam::ImportCategorizedView, 2011
  - activated, 2016
  - addOverlay, 2016
  - camItemInfoActivated, 2016
  - deselected, 2016
  - dragDropHandler, 2016
  - filterModel, 2016
  - hintAt, 2016
  - importFilterModel, 2017
  - importThumbnailModel, 2017
  - indexActivated, 2017
  - modelChanged, 2017

- nextIndexHint, 2017
- nextInOrder, 2017
- selected, 2018
- setCurrentInfo, 2018
- setCurrentUrl, 2018
- setCurrentWhenAvailable, 2018
- setSelectedCamItemInfos, 2018
- setSelectedUrls, 2018
- showContextMenuOnIndex, 2019
- toIndex, 2019
- Digikam::ImportCategoryDrawer, 2020
  - categoryHeight, 2021
  - drawCategory, 2021
- Digikam::ImportContextMenuHelper, 2022
  - addAction, 2023, 2024
  - addAssignTagsMenu, 2024
  - addGroupMenu, 2025
  - addLabelsAction, 2025
  - addRemoveTagsMenu, 2025
  - addRotateMenu, 2026
  - addSeparator, 2026
  - addServicesMenu, 2026
  - addSubMenu, 2026
  - exec, 2027
  - ImportContextMenuHelper, 2023
  - setImportFilterModel, 2027
- Digikam::ImportCoordinatesOverlay, 2028
  - checkIndex, 2030
  - createWidget, 2030
  - setActive, 2030
  - slotEntered, 2031
  - visualChange, 2031
- Digikam::ImportDelegate, 2032
  - acceptsActivation, 2036
  - acceptsToolTip, 2036
  - clearCaches, 2036
  - clearModelDataCaches, 2036
  - imageInformationRect, 2037
  - invalidatePaintingCache, 2037
  - pixmapForDrag, 2037
  - pixmapRect, 2037
  - retrieveThumbnailPixmap, 2037
  - setDefaultViewOptions, 2037
  - setSpacing, 2038
  - updateContentWidth, 2038
  - updateRects, 2038
  - updateSizeRectsAndPixmaps, 2038
- Digikam::ImportDelegate::ImportDelegatePrivate, 2039
  - clearRects, 2041
- Digikam::ImportDownloadOverlay, 2042
  - checkIndex, 2044
  - createWidget, 2044
  - setActive, 2044
  - slotEntered, 2045
  - visualChange, 2045
- Digikam::ImportDragDropHandler, 2046
  - accepts, 2047
  - createMimeData, 2047
- dropEvent, 2047
- mimeTypes, 2047
- Digikam::ImportFilterComboBox, 2048
- Digikam::ImportFilterDlg, 2049
- Digikam::ImportFilterModel, 2051
  - camItemInfosAdded, 2054
  - CategorizationModeRole, 2054
  - CategoryDateRole, 2054
  - CategoryFormatRole, 2054
  - categoryIdentifier, 2054
  - compareCategories, 2054
  - compareInfosCategories, 2055
  - importFilterModel, 2055
  - ImportFilterModelPointerRole, 2054
  - ImportFilterModelRoles, 2053
  - infosLessThan, 2055
  - setDirectSourceImportModel, 2055
  - SortOrderRole, 2054
  - subSortLessThan, 2055
- Digikam::ImportIconView, 2057
  - activated, 2063
  - setThumbnailSize, 2063
  - showContextMenu, 2063
  - showContextMenuOnInfo, 2063
  - slotSetupChanged, 2063
- Digikam::ImportIconView::Private, 2064
- Digikam::ImportItemModel, 2065
  - addCamItemInfoSynchronously, 2068
  - allRefreshingFinished, 2068
  - camItemInfo, 2068
  - camItemInfosCleared, 2068
  - clearCamItemInfos, 2069
  - ExtraDataDuplicateCount, 2068
  - ExtraDataRole, 2068
  - ImportItemModelPointerRole, 2068
  - ImportItemModelRoles, 2068
  - indexForCamItemInfo, 2069
  - indexForUrl, 2069
  - isRefreshing, 2069
  - itemInfosAboutToBeAdded, 2069
  - itemInfosAboutToBeRemoved, 2069
  - itemInfosAdded, 2070
  - itemInfosRemoved, 2070
  - preprocess, 2070
  - readyForIncrementalRefresh, 2070
  - removeIndex, 2070
  - requestIncrementalRefresh, 2070
  - retrieveCamItemInfo, 2071
  - setCameraThumbsController, 2071
  - setCamItemInfos, 2071
  - setKeepsFileUrlCache, 2071
  - setSendRemovalSignals, 2071
  - startIncrementalRefresh, 2071
  - startRefresh, 2072
  - ThumbnailRole, 2068
- Digikam::ImportItemPropertiesSideBarImport, 2073
  - applySettings, 2075
  - doLoadState, 2075

- doSaveState, [2076](#)
- Digikam::ImportItemPropertiesTab, [2077](#)
- Digikam::ImportLockOverlay, [2079](#)
  - checkIndex, [2081](#)
  - createWidget, [2081](#)
  - setActive, [2081](#)
  - slotEntered, [2082](#)
  - visualChange, [2082](#)
- Digikam::ImportNormalDelegate, [2083](#)
  - updateRects, [2087](#)
- Digikam::ImportNormalDelegatePrivate, [2088](#)
- Digikam::ImportOverlayWidget, [2090](#)
- Digikam::ImportPreviewView, [2091](#)
  - acceptsMouseClicked, [2093](#)
- Digikam::ImportRatingOverlay, [2094](#)
  - createWidget, [2096](#)
  - hide, [2096](#)
  - setActive, [2096](#)
  - slotEntered, [2097](#)
  - visualChange, [2097](#)
  - widgetEnterEvent, [2097](#)
  - widgetLeaveEvent, [2097](#)
- Digikam::ImportRenameParser, [2098](#)
- Digikam::ImportRotateOverlay, [2099](#)
  - checkIndex, [2102](#)
  - createButton, [2102](#)
  - setActive, [2102](#)
  - updateButton, [2102](#)
  - widgetEnterEvent, [2102](#)
  - widgetLeaveEvent, [2102](#)
- Digikam::ImportRotateOverlayButton, [2103](#)
  - icon, [2105](#)
  - sizeHint, [2105](#)
  - updateToolTip, [2105](#)
- Digikam::ImportSettings, [2106](#)
- Digikam::ImportSortFilterModel, [2109](#)
  - camItemInfosSorted, [2111](#)
  - importFilterModel, [2111](#)
  - mapToSourceImportModel, [2111](#)
  - setDirectSourceImportModel, [2111](#)
- Digikam::ImportStackedView, [2112](#)
  - PreviewCameraMode, [2113](#)
  - StackedViewMode, [2113](#)
- Digikam::ImportThumbnailBar, [2114](#)
  - setModelsFiltered, [2119](#)
  - setScrollBarPolicy, [2119](#)
  - slotSetupChanged, [2119](#)
- Digikam::ImportThumbnailDelegate, [2120](#)
  - acceptsActivation, [2124](#)
  - maximumSize, [2124](#)
  - setDefaultViewOptions, [2124](#)
  - updateContentWidth, [2124](#)
  - updateRects, [2125](#)
- Digikam::ImportThumbnailDelegatePrivate, [2126](#)
  - ImportThumbnailDelegatePrivate, [2128](#)
- Digikam::ImportThumbnailModel, [2129](#)
  - data, [2132](#)
  - ImportThumbnailModel, [2132](#)
  - setCameraThumbsController, [2132](#)
  - setData, [2132](#)
  - setEmitDataChanged, [2133](#)
- Digikam::ImportUI, [2134](#)
  - infoface, [2136](#)
- Digikam::ImportUI::Private, [2137](#)
- Digikam::ImportView, [2139](#)
- Digikam::InfoDlg, [2141](#)
- Digikam::InfraredContainer, [2142](#)
- Digikam::InfraredFilter, [2143](#)
  - filterAction, [2146](#)
  - filterIdentifier, [2146](#)
  - readParameters, [2146](#)
- Digikam::InitializationObserver, [2147](#)
- Digikam::InsertBookmarksCommand, [2148](#)
- Digikam::InternalTagName, [2149](#)
- Digikam::InvertFilter, [2150](#)
  - filterAction, [2153](#)
  - filterIdentifier, [2153](#)
  - readParameters, [2153](#)
- Digikam::IOFileSettings, [2153](#)
  - JPEGSubSampling, [2154](#)
- Digikam::IOJob, [2154](#)
- Digikam::IOJobData, [2155](#)
- Digikam::IOJobsManager, [2157](#)
  - buildCollectionTrashCounters, [2157](#)
  - instance, [2157](#)
  - startDTrashItemsListingForCollection, [2158](#)
  - startIOJobs, [2158](#)
- Digikam::IOJobsThread, [2159](#)
  - copyOrMove, [2161](#)
  - deleteFiles, [2161](#)
  - emptyDTrashItems, [2161](#)
  - errorsList, [2161](#)
  - hasErrors, [2161](#)
  - isCanceled, [2162](#)
  - jobData, [2162](#)
  - listDTrashItems, [2162](#)
  - renameFile, [2162](#)
  - restoreDTrashItems, [2162](#)
- Digikam::IptcCoreContactInfo, [2163](#)
- Digikam::IptcCoreLocationInfo, [2163](#)
- Digikam::IptcMetaEngineMergeHelper, [2164](#)
- Digikam::IptcWidget, [2165](#)
  - getMetadataTitle, [2167](#)
  - getTagDescription, [2167](#)
  - getTagTitle, [2167](#)
  - loadFromURL, [2167](#)
- Digikam::ItemAlbumFilterModel, [2168](#)
  - compareInfosCategories, [2172](#)
  - setItemFilterSettings, [2173](#)
- Digikam::ItemAlbumModel, [2174](#)
  - openAlbum, [2179](#)
  - refresh, [2179](#)
  - slotImageChange, [2179](#)
- Digikam::ItemAttributesWatch, [2180](#)
  - signalFileMetadataChanged, [2180](#)
  - signalImageRatingChanged, [2180](#)

- signalImagesChanged, [2181](#)
  - signalImageTagsChanged, [2181](#)
- Digikam::ItemCategorizedView, [2182](#)
  - activated, [2187](#)
  - albumAt, [2187](#)
  - dragDropHandler, [2187](#)
  - filterModel, [2187](#)
  - hintAt, [2188](#)
  - indexActivated, [2188](#)
  - nextIndexHint, [2188](#)
  - nextInOrder, [2188](#)
  - setCurrentInfo, [2188](#)
  - setCurrentUrl, [2188](#)
  - setCurrentUrlWhenAvailable, [2189](#)
  - setCurrentWhenAvailable, [2189](#)
  - setSelectedItemInfos, [2189](#)
  - setSelectedUrls, [2189](#)
  - showContextMenuOnIndex, [2189](#)
  - toIndex, [2189](#)
- Digikam::ItemCategoryDrawer, [2190](#)
  - categoryHeight, [2191](#)
  - drawCategory, [2191](#)
- Digikam::ItemChangeHint, [2192](#)
  - ChangeType, [2192](#)
  - ItemModified, [2192](#)
  - ItemRescan, [2192](#)
- Digikam::ItemComments, [2193](#)
  - addComment, [2195](#)
  - addHeadline, [2195](#)
  - addTitle, [2195](#)
  - apply, [2195](#)
  - changeComment, [2196](#)
  - commentForLanguage, [2196](#)
  - defaultComment, [2196](#)
  - ItemComments, [2194](#), [2195](#)
  - language, [2196](#)
  - LanguageChoiceBehavior, [2194](#)
  - numberOfComments, [2196](#)
  - remove, [2196](#)
  - removeAll, [2197](#)
  - removeAllComments, [2197](#)
  - replaceComments, [2197](#)
  - replaceFrom, [2197](#)
  - ReturnMatchingDefaultOrFirstLanguage, [2194](#)
  - ReturnMatchingLanguageOnly, [2194](#)
  - ReturnMatchingOrDefaultLanguage, [2194](#)
  - setUniqueBehavior, [2197](#)
  - toCaptionsMap, [2197](#)
  - type, [2198](#)
  - UniqueBehavior, [2194](#)
  - UniquePerLanguage, [2194](#)
  - UniquePerLanguageAndAuthor, [2194](#)
- Digikam::ItemCoordinatesOverlay, [2199](#)
  - checkIndex, [2201](#)
  - createWidget, [2201](#)
  - setActive, [2201](#)
  - slotEntered, [2202](#)
  - visualChange, [2202](#)
- Digikam::ItemCopyMoveHint, [2202](#)
  - ItemCopyMoveHint, [2203](#)
- Digikam::ItemCopyright, [2203](#)
  - AddEntryToExisting, [2204](#)
  - contactInfo, [2205](#)
  - copyrightNotice, [2205](#)
  - creator, [2205](#)
  - creatorJobTitle, [2205](#)
  - fillTemplate, [2205](#)
  - instructions, [2206](#)
  - ItemCopyright, [2205](#)
  - provider, [2206](#)
  - removeAll, [2206](#)
  - ReplaceAllEntries, [2204](#)
  - replaceFrom, [2206](#)
  - ReplaceLanguageEntry, [2204](#)
  - ReplaceMode, [2204](#)
  - rightsUsageTerms, [2206](#)
  - setCopyrightNotice, [2206](#)
  - setCreator, [2207](#)
  - setFromTemplate, [2207](#)
  - source, [2207](#)
- Digikam::ItemDelegate, [2208](#)
  - acceptsActivation, [2212](#)
  - acceptsToolTip, [2212](#)
  - clearCaches, [2212](#)
  - clearModelDataCaches, [2212](#)
  - imageInformationRect, [2213](#)
  - invalidatePaintingCache, [2213](#)
  - pixmapForDrag, [2213](#)
  - pixmapRect, [2213](#)
  - retrieveThumbnailPixmap, [2213](#)
  - setDefaultViewOptions, [2213](#)
  - setSpacing, [2214](#)
  - updateContentWidth, [2214](#)
  - updateRects, [2214](#)
  - updateSizeRectsAndPixmaps, [2214](#)
- Digikam::ItemDelegate::ItemDelegatePrivate, [2215](#)
  - clearRects, [2217](#)
- Digikam::ItemDelegateOverlay, [2217](#)
  - affectsMultiple, [2218](#)
  - mouseMoved, [2218](#)
  - setActive, [2218](#)
  - viewHasMultiSelection, [2218](#)
  - visualChange, [2219](#)
- Digikam::ItemDelegateOverlayContainer, [2220](#)
  - asDelegate, [2221](#)
  - ItemDelegateOverlayContainer, [2221](#)
- Digikam::ItemDescEditTab, [2222](#)
- Digikam::ItemDescEditTab::Private, [2224](#)
- Digikam::ItemDragDropHandler, [2226](#)
  - accepts, [2227](#)
  - createMimeData, [2227](#)
  - dropEvent, [2227](#)
  - mimeTypes, [2228](#)
  - setReadOnlyDrop, [2228](#)
- Digikam::ItemExtendedProperties, [2228](#)
  - intellectualGenre, [2229](#)



- ItemExtendedProperties, [2229](#)
- jobId, [2229](#)
- location, [2229](#)
- scene, [2229](#)
- similarityTo, [2230](#)
- subjectCode, [2230](#)
- Digikam::ItemFaceDelegate, [2231](#)
  - thumbnailPixmap, [2235](#)
  - updateRects, [2235](#)
- Digikam::ItemFaceDelegatePrivate, [2236](#)
- Digikam::ItemFilterModel, [2239](#)
  - addPrepareHook, [2243](#)
  - CategorizationModeRole, [2243](#)
  - CategoryAlbumIdRole, [2243](#)
  - CategoryDateRole, [2243](#)
  - CategoryFaceRole, [2243](#)
  - CategoryFormatRole, [2243](#)
  - categoryIdentifier, [2243](#)
  - compareCategories, [2244](#)
  - compareInfosCategories, [2244](#), [2245](#)
  - data, [2245](#)
  - filterMatches, [2245](#)
  - filterMatchesForText, [2245](#)
  - filterSettingsChanged, [2245](#)
  - GroupsOpenRole, [2243](#)
  - imageFilterModel, [2246](#)
  - imageInfosAdded, [2246](#)
  - infosLessThan, [2246](#)
  - isGroupOpen, [2246](#)
  - ItemFilterModelRoles, [2243](#)
  - setDayFilter, [2246](#)
  - setDirectSourceItemModel, [2246](#)
  - setGroupItemFilterSettings, [2247](#)
  - setItemFilterSettings, [2247](#)
  - setItemSortSettings, [2247](#)
  - setSendItemInfoSignals, [2247](#)
  - setVersionItemFilterSettings, [2247](#)
  - SortOrderRole, [2243](#)
  - subSortLessThan, [2247](#)
  - suggestedWatchFlags, [2248](#)
- Digikam::ItemFilterModelFilterer, [2249](#)
  - process, [2251](#)
- Digikam::ItemFilterModelPrepareHook, [2251](#)
- Digikam::ItemFilterModelPreparer, [2252](#)
  - process, [2254](#)
- Digikam::ItemFilterModelTodoPackage, [2254](#)
- Digikam::ItemFilterModelWorker, [2255](#)
- Digikam::ItemFilterSettings, [2257](#)
  - matches, [2258](#)
  - watchFlags, [2258](#)
- Digikam::ItemFiltersHistoryItemDelegate, [2259](#)
- Digikam::ItemFiltersHistoryModel, [2260](#)
- Digikam::ItemFiltersHistoryTreeItem, [2261](#)
- Digikam::ItemFullScreenOverlay, [2262](#)
  - checkIndex, [2265](#)
  - createButton, [2265](#)
  - setActive, [2265](#)
  - updateButton, [2265](#)
  - widgetEnterEvent, [2265](#)
  - widgetLeaveEvent, [2265](#)
- Digikam::ItemFullScreenOverlayButton, [2266](#)
  - icon, [2267](#)
  - sizeHint, [2267](#)
  - updateToolTip, [2268](#)
- Digikam::ItemGPS, [2269](#)
  - loadImageData, [2271](#)
  - saveChanges, [2271](#)
- Digikam::ItemGPSModelHelper, [2272](#)
  - bestRepresentativeIndexFromList, [2273](#)
  - itemCoordinates, [2273](#)
  - model, [2274](#)
  - pixmapFromRepresentativeIndex, [2274](#)
  - selectionModel, [2274](#)
- Digikam::ItemHistoryGraph, [2274](#)
  - addHistory, [2275](#)
  - addRelations, [2276](#)
  - addScannedHistory, [2276](#)
  - allImages, [2276](#)
  - categorize, [2276](#)
  - clear, [2276](#)
  - dropUnresolvedEntries, [2276](#)
  - fromInfo, [2276](#)
  - hasEdges, [2277](#)
  - hasUnresolvedEntries, [2277](#)
  - HistoryLoadingFlag, [2275](#)
  - leafImages, [2277](#)
  - LoadLeavesHistory, [2275](#)
  - LoadRelationCloud, [2275](#)
  - LoadSubjectHistory, [2275](#)
  - prepareForDisplay, [2277](#)
  - reduceEdges, [2277](#)
  - relationCloud, [2277](#)
  - rootImages, [2277](#)
  - sortForInfo, [2278](#)
- Digikam::ItemHistoryGraphData, [2279](#)
- Digikam::ItemHistoryGraphModel, [2283](#)
  - imageModel, [2285](#)
  - imageModelIndex, [2285](#)
  - indexForInfo, [2285](#)
  - setHistory, [2285](#)
- Digikam::ItemIconView, [2286](#)
  - allNeedGroupResolving, [2290](#)
  - allUrls, [2290](#)
  - selectedUrls, [2290](#)
  - slotFitToWindow, [2290](#)
  - slotImageQualitySorter, [2291](#)
  - slotRemoveTag, [2291](#)
- Digikam::ItemIconView::Private, [2291](#)
- Digikam::ItemInfo, [2292](#)
  - ~ItemInfo, [2295](#)
  - addTagPaths, [2296](#)
  - addToGroup, [2296](#)
  - albumId, [2296](#)
  - albumRootId, [2296](#)
  - aspectRatio, [2296](#)
  - category, [2296](#)

- clearGroup, [2296](#)
- colorLabel, [2297](#)
- comment, [2297](#)
- copyItem, [2297](#)
- currentReferencelImage, [2297](#)
- dateTime, [2297](#)
- dimensions, [2298](#)
- faceCount, [2298](#)
- filePath, [2298](#)
- fileSize, [2298](#)
- fileUrl, [2298](#)
- format, [2298](#)
- fromLocalFile, [2299](#)
- fromLocationAlbumAndName, [2299](#)
- fromUniqueHash, [2299](#)
- getDatabaseFieldsRaw, [2299](#)
- getSuggestedNames, [2299](#)
- groupedImages, [2299](#)
- groupImage, [2300](#)
- hasDerivedImages, [2300](#)
- hasGroupedImages, [2300](#)
- hash, [2300](#)
- historyImageId, [2300](#)
- id, [2300](#)
- imageComments, [2300](#)
- imageCommonContainer, [2301](#)
- imageCopyright, [2301](#)
- imageExtendedProperties, [2301](#)
- imageHistory, [2301](#)
- imagePosition, [2301](#)
- imageTagPair, [2301](#)
- isGrouped, [2301](#)
- isLocationAvailable, [2302](#)
- isNull, [2302](#)
- isRemoved, [2302](#)
- isVisible, [2302](#)
- ItemInfo, [2295](#)
- longitudeNumber, [2302](#)
- manualOrder, [2302](#)
- markDerivedFrom, [2302](#)
- metadataTemplate, [2303](#)
- modDateTime, [2303](#)
- name, [2303](#)
- orientation, [2303](#)
- pickLabel, [2303](#)
- rating, [2303](#)
- relationCloud, [2303](#)
- relativePath, [2304](#)
- removeAllTags, [2304](#)
- removeFromGroup, [2304](#)
- removeMetadataTemplate, [2304](#)
- removeTag, [2304](#)
- setColorLabel, [2304](#)
- setDateTime, [2305](#)
- setManualOrder, [2305](#)
- setMetadataTemplate, [2305](#)
- setModDateTime, [2305](#)
- setName, [2305](#)
- setOrientation, [2306](#)
- setPickLabel, [2306](#)
- setRating, [2306](#)
- setTag, [2306](#)
- setVisible, [2306](#)
- tagIds, [2307](#)
- thumbnailIdentifier, [2307](#)
- title, [2307](#)
- unconfirmedFaceCount, [2307](#)
- uniqueHash, [2307](#)
- uuid, [2307](#)
- Digikam::ItemInfoAlbumsJob, [2308](#)
- Digikam::ItemInfoCache, [2309](#)
  - albumRelativePath, [2309](#)
  - cacheByName, [2309](#)
  - dropInfo, [2310](#)
  - getImageGroupedCount, [2310](#)
  - infoForId, [2310](#)
  - infoForPath, [2310](#)
  - invalidate, [2310](#)
- Digikam::ItemInfoData, [2311](#)
- Digikam::ItemInfoJob, [2313](#)
- Digikam::ItemInfoList, [2314](#)
  - singleGroupMainItem, [2314](#)
- Digikam::ItemInfoReadLocker, [2315](#)
- Digikam::ItemInfoSet, [2315](#)
- Digikam::ItemInfoStatic, [2315](#)
- Digikam::ItemInfoTaskSplitter, [2317](#)
- Digikam::ItemInfoWriteLocker, [2319](#)
- Digikam::ItemListDragDropHandler, [2320](#)
- Digikam::ItemLISTER, [2320](#)
  - list, [2321](#)
  - listAreaRange, [2321](#)
  - listDateRange, [2321](#)
  - listHaarSearch, [2322](#)
  - listImageTagPropertySearch, [2322](#)
  - listPALbum, [2322](#)
  - listSearch, [2322](#)
  - listTag, [2323](#)
  - setListOnlyAvailable, [2323](#)
  - setRecursive, [2323](#)
- Digikam::ItemLISTER::Private, [2323](#)
- Digikam::ItemLISTERJobGrowingPartsSendingReceiver, [2324](#)
  - receive, [2325](#)
- Digikam::ItemLISTERJobPartsSendingReceiver, [2326](#)
  - receive, [2327](#)
- Digikam::ItemLISTERJobReceiver, [2328](#)
  - error, [2329](#)
- Digikam::ItemLISTERReceiver, [2330](#)
- Digikam::ItemLISTERRecord, [2331](#)
- Digikam::ItemLISTERValueListReceiver, [2332](#)
  - error, [2333](#)
  - receive, [2333](#)
- Digikam::ItemListModel, [2334](#)
  - imageInfosRemoved, [2338](#)
- Digikam::ItemMarkerTiler, [2339](#)
  - bestRepresentativeIndexFromList, [2340](#)

- getGlobalGroupState, [2340](#)
- getTile, [2341](#)
- getTileGroupState, [2341](#)
- getTileMarkerCount, [2341](#)
- getTileRepresentativeMarker, [2341](#)
- getTileSelectedCount, [2341](#)
- indicesEqual, [2341](#)
- onIndicesClicked, [2341](#)
- onIndicesMoved, [2342](#)
- pixmapFromRepresentativeIndex, [2342](#)
- prepareTiles, [2342](#)
- regenerateTiles, [2342](#)
- removeMarkerIndexFromGrid, [2342](#)
- setActive, [2343](#)
- tileNew, [2343](#)
- tilerFlags, [2343](#)
- Digikam::ItemMetadataAdjustmentHint, [2343](#)
  - AboutToEditMetadata, [2344](#)
  - AdjustmentStatus, [2343](#)
  - MetadataEditingAborted, [2344](#)
  - MetadataEditingFinished, [2344](#)
- Digikam::ItemModel, [2345](#)
  - addItemInfo, [2349](#)
  - addItemInfoSynchronously, [2349](#)
  - allRefreshingFinished, [2349](#)
  - clearItemInfos, [2349](#)
  - CreationDateRole, [2348](#)
  - ensureHasGroupedImages, [2349](#)
  - ensureHasItemInfo, [2349](#)
  - ExtraDataDuplicateCount, [2348](#)
  - ExtraDataRole, [2348](#)
  - FilterModelRoles, [2348](#)
  - imageChange, [2349](#)
  - imageInfo, [2350](#)
  - imageInfosAboutToBeAdded, [2350](#)
  - imageInfosAboutToBeRemoved, [2350](#)
  - imageInfosAdded, [2350](#)
  - imageInfosCleared, [2351](#)
  - imageInfosRemoved, [2351](#)
  - imageTagChange, [2351](#)
  - indexForItemInfo, [2351](#)
  - indexForPath, [2351](#)
  - isRefreshing, [2351](#)
  - ItemModelPointerRole, [2348](#)
  - ItemModelRoles, [2348](#)
  - LLeftPanelRole, [2348](#)
  - LRightPanelRole, [2348](#)
  - preprocess, [2352](#)
  - readyForIncrementalRefresh, [2352](#)
  - removeIndex, [2352](#)
  - requestIncrementalRefresh, [2352](#)
  - retrieveItemInfo, [2352](#)
  - setItemInfos, [2352](#)
  - setKeepsFilePathCache, [2352](#)
  - setPreprocessor, [2353](#)
  - setSendRemovalSignals, [2353](#)
  - setWatchFlags, [2353](#)
  - startIncrementalRefresh, [2353](#)
  - startRefresh, [2353](#)
  - SubclassRoles, [2348](#)
  - ThumbnailRole, [2348](#)
- Digikam::ItemPosition, [2354](#)
  - altitude, [2355](#)
  - altitudeFormatted, [2355](#)
  - apply, [2355](#)
  - isEmpty, [2355](#)
  - ItemPosition, [2354](#)
  - latitude, [2355](#)
  - latitudeFormatted, [2355](#)
  - latitudeNumber, [2355](#)
  - latitudeUserPresentableNumbers, [2356](#)
  - remove, [2356](#)
  - removeAltitude, [2356](#)
  - setAltitude, [2356](#)
  - setLatitude, [2356](#)
- Digikam::ItemPreviewCanvas, [2358](#)
- Digikam::ItemPreviewView, [2361](#)
  - acceptsMouseClicked, [2363](#)
- Digikam::ItemPropertiesColorsTab, [2364](#)
- Digikam::ItemPropertiesGPSTab, [2365](#)
- Digikam::ItemPropertiesHistoryTab, [2366](#)
- Digikam::ItemPropertiesMetadataTab, [2367](#)
- Digikam::ItemPropertiesSideBar, [2368](#)
  - doLoadState, [2371](#)
  - doSaveState, [2371](#)
- Digikam::ItemPropertiesSideBarDB, [2372](#)
  - doLoadState, [2376](#)
  - doSaveState, [2376](#)
  - itemChanged, [2376](#)
- Digikam::ItemPropertiesTab, [2377](#)
  - aspectRatioToString, [2380](#)
  - humanReadableBytesCount, [2380](#)
  - permissionsString, [2380](#)
  - shortenedMakeInfo, [2380](#)
  - shortenedTagPaths, [2380](#)
- Digikam::ItemPropertiesTab::Private, [2381](#)
- Digikam::ItemPropertiesVersionsTab, [2382](#)
- Digikam::ItemQueryBuilder, [2383](#)
  - setImageTagPropertiesJoined, [2383](#)
- Digikam::ItemQueryPostHook, [2383](#)
  - ItemQueryPostHook, [2384](#)
- Digikam::ItemQueryPostHooks, [2384](#)
  - addHook, [2384](#)
  - checkPosition, [2384](#)
- Digikam::ItemRatingOverlay, [2385](#)
  - createWidget, [2387](#)
  - hide, [2387](#)
  - setActive, [2387](#)
  - slotEntered, [2388](#)
  - visualChange, [2388](#)
  - widgetEnterEvent, [2388](#)
  - widgetLeaveEvent, [2388](#)
- Digikam::ItemRotateOverlay, [2389](#)
  - checkIndex, [2392](#)
  - createButton, [2392](#)
  - setActive, [2392](#)

- updateButton, [2392](#)
- widgetEnterEvent, [2392](#)
- widgetLeaveEvent, [2392](#)
- Digikam::ItemRotateOverlayButton, [2393](#)
  - icon, [2395](#)
  - sizeHint, [2395](#)
  - updateToolTip, [2395](#)
- Digikam::ItemScanInfo, [2395](#)
- Digikam::ItemScanner, [2395](#)
  - cleanScan, [2398](#)
  - commit, [2398](#)
  - copiedFrom, [2398](#)
  - creationDateFromFilesystem, [2398](#)
  - fileModified, [2398](#)
  - fillCommonContainer, [2399](#)
  - fillVideoMetadataContainer, [2399](#)
  - formatToString, [2399](#)
  - hasHistoryToResolve, [2399](#)
  - id, [2399](#)
  - iptcCorePropertyName, [2399](#)
  - itemScanInfo, [2400](#)
  - ItemScanner, [2397](#), [2398](#)
  - loadFromDisk, [2400](#)
  - newFile, [2400](#)
  - newFileFullScan, [2400](#)
  - rescan, [2400](#)
  - resolvedImageHistory, [2400](#)
  - resolveHistoryImageId, [2401](#)
  - resolveImageHistory, [2401](#)
  - sameReferredImage, [2401](#)
  - setCategory, [2401](#)
  - sortByProximity, [2401](#)
  - tagItemHistoryGraph, [2401](#)
- Digikam::ItemScanner::Private, [2402](#)
- Digikam::ItemScannerCommit, [2402](#)
- Digikam::ItemSelectionOverlay, [2403](#)
  - createButton, [2406](#)
  - setActive, [2406](#)
  - updateButton, [2406](#)
- Digikam::ItemSelectionOverlayButton, [2407](#)
  - icon, [2408](#)
  - sizeHint, [2408](#)
  - updateToolTip, [2409](#)
- Digikam::ItemSelectionPropertiesTab, [2410](#)
- Digikam::ItemShortInfo, [2412](#)
- Digikam::ItemSortCollator, [2412](#)
  - instance, [2413](#)
- Digikam::ItemSortSettings, [2413](#)
  - CategorizationMode, [2414](#)
  - compare, [2415](#)
  - compareByOrder, [2415](#)
  - compareCategories, [2415](#)
  - compareValue, [2415](#)
  - DefaultOrder, [2414](#)
  - lessThan, [2415](#), [2416](#)
  - lessThanByOrder, [2416](#)
  - naturalCompare, [2416](#)
  - NoCategories, [2414](#)
  - OneCategory, [2414](#)
  - SortByAspectRatio, [2415](#)
  - SortByFaces, [2415](#)
  - SortByImageSize, [2415](#)
  - SortOrder, [2414](#)
  - SortRole, [2414](#)
  - watchFlags, [2416](#)
- Digikam::ItemTagPair, [2417](#)
  - addProperty, [2418](#)
  - assignTag, [2418](#)
  - availablePairs, [2418](#)
  - isAssigned, [2418](#)
  - ItemTagPair, [2417](#)
  - unAssignTag, [2418](#)
- Digikam::ItemThumbnailBar, [2419](#)
  - hasHiddenGroupedImages, [2425](#)
  - setModelsFiltered, [2425](#)
  - setScrollBarPolicy, [2425](#)
  - slotSetupChanged, [2425](#)
- Digikam::ItemThumbnailDelegate, [2426](#)
  - acceptsActivation, [2430](#)
  - maximumSize, [2430](#)
  - setDefaultViewOptions, [2430](#)
  - updateContentWidth, [2430](#)
  - updateRects, [2431](#)
- Digikam::ItemThumbnailDelegatePrivate, [2432](#)
- Digikam::ItemThumbnailModel, [2435](#)
  - data, [2439](#)
  - imageInfosCleared, [2439](#)
  - ItemThumbnailModel, [2439](#)
  - preloadThumbnails, [2439](#)
  - setData, [2439](#)
  - setEmitDataChanged, [2440](#)
  - setPreloadThumbnails, [2440](#)
  - setThumbnailLoadThread, [2440](#)
- Digikam::ItemVersionsModel, [2441](#)
- Digikam::ItemViewCategorized, [2442](#)
  - clicked, [2445](#)
  - filterModel, [2445](#)
  - indexForCategoryAt, [2445](#)
  - keyPressed, [2445](#)
  - mapIndexForDragDrop, [2446](#)
  - nextIndexHint, [2446](#)
  - pixmapForDrag, [2446](#)
  - rowsRemoved, [2446](#)
  - scrollToRelaxed, [2446](#)
  - selectionChanged, [2447](#)
  - selectionCleared, [2447](#)
  - setInitialSelectedItem, [2447](#)
  - setScrollCurrentToCenter, [2447](#)
  - setScrollStepGranularity, [2447](#)
  - setSpacing, [2447](#)
  - setUsePointingHandCursor, [2447](#)
  - showContextMenuOnIndex, [2448](#)
  - showToolTip, [2448](#)
  - toFirstIndex, [2448](#)
  - viewportClicked, [2448](#)
- Digikam::ItemViewDelegate, [2449](#)

- acceptsActivation, [2452](#)
- acceptsToolTip, [2452](#)
- asDelegate, [2452](#)
- drawThumbnail, [2452](#)
- gridSize, [2453](#)
- imageInformationRect, [2453](#)
- mouseMoved, [2453](#)
- pixmapRect, [2453](#)
- ratingPixmap, [2453](#)
- ratingRect, [2454](#)
- setDefaultViewOptions, [2454](#)
- setRatingEdited, [2454](#)
- setSpacing, [2454](#)
- setThumbnailSize, [2454](#)
- Digikam::ItemViewDelegatePrivate, [2455](#)
  - clearRects, [2456](#)
- Digikam::ItemViewHoverButton, [2456](#)
  - icon, [2457](#)
  - setup, [2457](#)
  - sizeHint, [2457](#)
  - updateToolTip, [2458](#)
- Digikam::ItemViewImportDelegate, [2459](#)
  - acceptsActivation, [2462](#)
  - acceptsToolTip, [2462](#)
  - asDelegate, [2462](#)
  - gridSize, [2462](#)
  - imageInformationRect, [2463](#)
  - invalidatePaintingCache, [2463](#)
  - mouseMoved, [2463](#)
  - pixmapRect, [2463](#)
  - prepareRatingPixmaps, [2463](#)
  - ratingRect, [2463](#)
  - setDefaultViewOptions, [2464](#)
  - setRatingEdited, [2464](#)
  - setSpacing, [2464](#)
  - setThumbnailSize, [2464](#)
- Digikam::ItemViewImportDelegatePrivate, [2465](#)
  - clearRects, [2466](#)
- Digikam::ItemViewToolTip, [2467](#)
  - repositionRect, [2468](#)
  - show, [2468](#)
  - tipContents, [2468](#)
- Digikam::ItemViewUtilities, [2469](#)
- Digikam::ItemVisibilityController, [2471](#)
  - addItem, [2473](#)
  - clear, [2473](#)
  - createAnimation, [2473](#)
  - ExcludeFadingOut, [2473](#)
  - hasVisibleItems, [2473](#)
  - hiddenAndRemoved, [2474](#)
  - hideAndRemoveItem, [2474](#)
  - IncludeFadingOut, [2473](#)
  - IncludeFadingOutMode, [2472](#)
  - items, [2474](#)
  - propertiesAssigned, [2474](#)
  - setEasingCurve, [2474](#)
  - setItemThatShallBeShown, [2474](#)
  - setShallBeShown, [2475](#)
  - show, [2475](#)
  - showItem, [2475](#)
  - State, [2473](#)
  - visibleItems, [2475](#)
- Digikam::ItemVisibilityControllerPropertyObject, [2476](#)
  - ItemVisibilityControllerPropertyObject, [2477](#)
- Digikam::JPEGUtils::digikam\_source\_mgr, [2477](#)
- Digikam::JPEGUtils::JpegRotator, [2477](#)
  - ~JpegRotator, [2478](#)
  - autoExifTransform, [2478](#)
  - exifTransform, [2478](#)
  - JpegRotator, [2478](#)
  - setCurrentOrientation, [2479](#)
  - setDestinationFile, [2479](#)
  - setDocumentName, [2479](#)
- Digikam::KDNNodeBase, [2480](#)
  - createNode, [2481](#)
  - getClosestNeighbors, [2481](#)
  - getIdentity, [2481](#)
  - getPosition, [2481](#)
  - insert, [2481](#)
  - setNodeId, [2482](#)
- Digikam::KDNNodeBase::NodeCompareResult, [2482](#)
- Digikam::KDNNodeOpenFace, [2483](#)
  - createNode, [2484](#)
  - nodeCompare, [2484](#)
- Digikam::KDNNodeSFace, [2485](#)
  - createNode, [2486](#)
  - nodeCompare, [2486](#)
- Digikam::KDTreeBase, [2487](#)
  - add, [2488](#)
  - createNode, [2488](#)
  - getClosestNeighbors, [2488](#)
  - KDTreeBase, [2487](#)
- Digikam::KDTreeOpenFace, [2489](#)
- Digikam::KDTreeSFace, [2490](#)
- Digikam::KeywordSearchReader, [2491](#)
- Digikam::KeywordSearchWriter, [2493](#)
- Digikam::LabelsSideBarWidget, [2495](#)
  - applySettings, [2496](#)
  - changeAlbumFromHistory, [2496](#)
  - doLoadState, [2497](#)
  - doSaveState, [2497](#)
  - getCaption, [2497](#)
  - getIcon, [2497](#)
  - setActive, [2497](#)
- Digikam::LabelsTreeView, [2498](#)
  - colorRectPixmap, [2499](#)
  - doLoadState, [2500](#)
  - doSaveState, [2500](#)
  - goldenStarPixmap, [2500](#)
  - isCheckedable, [2500](#)
  - isLoadingState, [2500](#)
  - restoreSelectionFromHistory, [2500](#)
  - selectedLabels, [2501](#)
- Digikam::LanguagesList, [2501](#)
- Digikam::LcmsLock, [2502](#)
  - LcmsLock, [2502](#)

- Digikam::LensDistortionFilter, [2503](#)
  - [filterAction](#), [2506](#)
  - [filterIdentifier](#), [2506](#)
  - [readParameters](#), [2506](#)
- Digikam::LensDistortionPixelAccess, [2506](#)
- Digikam::LensFunCameraSelector, [2507](#)
  - [setPassiveMetadataUsage](#), [2508](#)
- Digikam::LensFunContainer, [2508](#)
- Digikam::LensFunFilter, [2509](#)
  - [filterAction](#), [2512](#)
  - [filterIdentifier](#), [2512](#)
  - [readParameters](#), [2512](#)
- Digikam::LensFunIface, [2512](#)
  - [lensDescription](#), [2513](#)
  - [makeDescription](#), [2513](#)
  - [modelDescription](#), [2513](#)
- Digikam::LensFunSettings, [2514](#)
- Digikam::LessThanByProximityToSubject, [2515](#)
- Digikam::LevelsContainer, [2515](#)
- Digikam::LevelsFilter, [2516](#)
  - [filterAction](#), [2519](#)
  - [filterIdentifier](#), [2519](#)
  - [readParameters](#), [2519](#)
- Digikam::LibsInfoDlg, [2520](#)
  - [LibsInfoDlg](#), [2521](#)
- Digikam::LightTablePreview, [2522](#)
- Digikam::LightTableThumbBar, [2525](#)
- Digikam::LightTableView, [2532](#)
- Digikam::LightTableWindow, [2534](#)
  - [infolface](#), [2536](#)
  - [loadItemInfos](#), [2536](#)
  - [setLeftRightItems](#), [2537](#)
  - [slotApplicationSettingsChanged](#), [2537](#)
- Digikam::LightTableWindow::Private, [2537](#)
- Digikam::ListItem, [2539](#)
  - [containsItem](#), [2540](#)
- Digikam::ListViewComboBox, [2541](#)
  - [installView](#), [2543](#)
  - [ListViewComboBox](#), [2542](#)
  - [sendViewportEventToView](#), [2543](#)
  - [view](#), [2543](#)
- Digikam::LoadingCache, [2544](#)
  - [addLoadingProcess](#), [2545](#)
  - [fileChanged](#), [2545](#)
  - [getCacheSize](#), [2545](#)
  - [isCacheable](#), [2545](#)
  - [notifyFileChanged](#), [2546](#)
  - [notifyNewLoadingProcess](#), [2546](#)
  - [putImage](#), [2546](#)
  - [putThumbnail](#), [2546](#)
  - [removeImage](#), [2546](#)
  - [removeImages](#), [2546](#)
  - [removeLoadingProcess](#), [2547](#)
  - [removeThumbnail](#), [2547](#)
  - [removeThumbnails](#), [2547](#)
  - [retrieveImage](#), [2547](#)
  - [retrieveLoadingProcess](#), [2547](#)
  - [retrieveThumbnail](#), [2547](#)
  - [setCacheSize](#), [2547](#)
  - [setFileWatch](#), [2548](#)
  - [setThumbnailCacheSize](#), [2548](#)
- Digikam::LoadingCache::CacheLock, [2548](#)
- Digikam::LoadingCacheFileWatch, [2549](#)
  - [notifyFileChanged](#), [2550](#)
- Digikam::LoadingCacheInterface, [2550](#)
  - [cleanCache](#), [2550](#)
  - [cleanThumbnailCache](#), [2550](#)
  - [cleanUp](#), [2550](#)
  - [connectToSignalFileChanged](#), [2550](#)
  - [fileChanged](#), [2551](#)
  - [putImage](#), [2551](#)
  - [setCacheOptions](#), [2551](#)
- Digikam::LoadingDescription, [2551](#)
  - [ApplyTransform](#), [2552](#)
  - [cacheKey](#), [2554](#)
  - [ColorManagementSettings](#), [2552](#)
  - [ConvertForDisplay](#), [2552](#)
  - [ConvertForOutput](#), [2552](#)
  - [equalsIgnoreReducedVersion](#), [2554](#)
  - [equalsOrBetterThan](#), [2554](#)
  - [isPreviewImage](#), [2554](#)
  - [isReducedVersion](#), [2554](#)
  - [isThumbnail](#), [2554](#)
  - [LoadingDescription](#), [2553](#)
  - [lookupCacheKeys](#), [2554](#)
  - [needCheckRawDecoding](#), [2554](#)
  - [operator==](#), [2555](#)
  - [possibleCacheKeys](#), [2555](#)
  - [RawDecodingCustomSettings](#), [2553](#)
  - [RawDecodingDefaultSettings](#), [2553](#)
  - [RawDecodingGlobalSettings](#), [2553](#)
  - [RawDecodingHint](#), [2552](#)
  - [RawDecodingTimeOptimized](#), [2553](#)
  - [thumbnailIdentifier](#), [2555](#)
- Digikam::LoadingDescription::PostProcessingParameters, [2555](#)
- Digikam::LoadingDescription::PreviewParameters, [2556](#)
- Digikam::LoadingProcess, [2557](#)
- Digikam::LoadingProcessListener, [2558](#)
- Digikam::LoadingTask, [2559](#)
  - [continueQuery](#), [2560](#)
  - [execute](#), [2560](#)
  - [progressInfo](#), [2561](#)
  - [type](#), [2561](#)
- Digikam::LoadSaveFileInfoProvider, [2561](#)
  - [dimensionsHint](#), [2562](#)
  - [orientationHint](#), [2562](#)
- Digikam::LoadSaveNotifier, [2563](#)
  - [thumbnailLoaded](#), [2564](#)
- Digikam::LoadSaveTask, [2565](#)
- Digikam::LoadSaveThread, [2567](#)
  - [~LoadSaveThread](#), [2570](#)
  - [AccessMode](#), [2569](#)
  - [AccessModeRead](#), [2569](#)
  - [AccessModeReadWrite](#), [2569](#)
  - [exifOrientation](#), [2570](#)

- imageLoaded, [2570](#)
- imageSaved, [2570](#)
- imageStartedLoading, [2570](#)
- imageStartedSaving, [2571](#)
- load, [2571](#)
- loadingProgress, [2571](#)
- moreCompleteLoadingAvailable, [2571](#)
- NotificationPolicy, [2570](#)
- NotificationPolicyDirect, [2570](#)
- NotificationPolicyTimeLimited, [2570](#)
- run, [2571](#)
- save, [2571](#)
- savingProgress, [2572](#)
- signalImageLoaded, [2572](#)
- signalImageStartedLoading, [2572](#)
- signalLoadingProgress, [2572](#)
- signalMoreCompleteLoadingAvailable, [2572](#)
- thumbnailLoaded, [2573](#)
- Digikam::LocalContrastContainer, [2573](#)
- Digikam::LocalContrastFilter, [2574](#)
  - filterAction, [2577](#)
  - filterIdentifier, [2577](#)
  - readParameters, [2577](#)
- Digikam::LocalContrastSettings, [2577](#)
- Digikam::LocalizeConfig, [2578](#)
- Digikam::LocalizeContainer, [2579](#)
  - ignoredWords, [2579](#)
- Digikam::LocalizeSelector, [2580](#)
- Digikam::LocalizeSelectorList, [2581](#)
- Digikam::LocalizeSettings, [2582](#)
  - instance, [2583](#)
  - setSettings, [2583](#)
  - settings, [2583](#)
- Digikam::LookupAltitude, [2584](#)
- Digikam::LookupAltitude::Request, [2585](#)
- Digikam::LookupAltitudeGeonames, [2586](#)
  - backendHumanName, [2587](#)
  - backendName, [2587](#)
  - cancel, [2587](#)
  - errorMessage, [2587](#)
  - getRequest, [2588](#)
  - getRequests, [2588](#)
  - getStatus, [2588](#)
  - startLookup, [2588](#)
- Digikam::LookupFactory, [2588](#)
- Digikam::MaintenanceData, [2589](#)
- Digikam::MaintenanceDlg, [2589](#)
- Digikam::MaintenanceDlg::Private, [2590](#)
- Digikam::MaintenanceMgr, [2591](#)
- Digikam::MaintenanceSettings, [2592](#)
- Digikam::MaintenanceThread, [2594](#)
  - signalAddItemsToProcess, [2596](#)
  - signalAdvance, [2596](#)
  - signalCanceled, [2596](#)
  - signalCompleted, [2596](#)
  - signalData, [2596](#)
  - signalFinished, [2596](#)
  - signalRemovePending, [2597](#)
  - signalStarted, [2597](#)
- Digikam::MaintenanceTool, [2597](#)
  - setNotificationEnabled, [2599](#)
  - setUseMultiCoreCPU, [2599](#)
  - signalCanceled, [2599](#)
  - signalComplete, [2600](#)
- Digikam::MakerNoteWidget, [2601](#)
  - getMetadataTitle, [2603](#)
  - getTagDescription, [2603](#)
  - getTagTitle, [2603](#)
  - loadFromURL, [2603](#)
- Digikam::ManagedLoadSaveThread, [2604](#)
  - load, [2609](#)
  - LoadingMode, [2607](#)
  - LoadingModeNormal, [2607](#)
  - LoadingModeShared, [2607](#)
  - LoadingPolicy, [2607](#)
  - LoadingPolicyAppend, [2608](#)
  - LoadingPolicyFirstRemovePrevious, [2608](#)
  - LoadingPolicyPreload, [2608](#)
  - LoadingPolicyPrepend, [2608](#)
  - LoadingPolicySimpleAppend, [2608](#)
  - LoadingPolicySimplePrepend, [2608](#)
  - LoadingTaskFilter, [2608](#)
  - LoadingTaskFilterAll, [2608](#)
  - LoadingTaskFilterPreloading, [2608](#)
  - ManagedLoadSaveThread, [2609](#)
  - save, [2609](#)
  - setLoadingPolicy, [2609](#)
  - stopLoading, [2609](#)
  - stopSaving, [2609](#)
  - TerminationPolicy, [2608](#)
  - TerminationPolicyTerminateAll, [2608](#)
  - TerminationPolicyTerminateLoading, [2608](#)
  - TerminationPolicyTerminatePreloading, [2608](#)
  - TerminationPolicyWait, [2608](#)
- Digikam::MapBackend, [2610](#)
  - mapWidget, [2612](#)
  - mouseModeChanged, [2612](#)
  - setActive, [2612](#)
- Digikam::MapDragData, [2612](#)
- Digikam::MapDragDropHandler, [2613](#)
  - accepts, [2614](#)
  - createMimeData, [2614](#)
  - dropEvent, [2614](#)
- Digikam::MapViewModelHelper, [2615](#)
  - bestRepresentativeIndexFromList, [2616](#)
  - itemCoordinates, [2617](#)
  - model, [2617](#)
  - onIndicesClicked, [2617](#)
  - pixmapFromRepresentativeIndex, [2618](#)
  - selectionModel, [2618](#)
- Digikam::MapWidget, [2618](#)
  - ~MapWidget, [2622](#)
  - addUngroupedModel, [2622](#)
  - adjustBoundariesToGroupedMarkers, [2622](#)
  - applyCacheToBackend, [2623](#)
  - convertZoomToBackendZoom, [2623](#)

- dragEnterEvent, [2623](#)
- getColorInfos, [2623](#)
- getDecoratedPixmapForCluster, [2624](#)
- removeUngroupedModel, [2624](#)
- setBackend, [2624](#)
- setGroupedModel, [2624](#)
- setSortKey, [2624](#)
- setThumbnailSize, [2624](#)
- slotClustersClicked, [2624](#)
- slotClustersMoved, [2624](#)
- slotItemDisplaySettingsChanged, [2624](#)
- slotMouseModeChanged, [2625](#)
- slotNewSelectionFromMap, [2625](#)
- slotUpdateActionsEnabled, [2625](#)
- updateClusters, [2625](#)
- Digikam::MapWidget::Private, [2625](#)
- Digikam::MapWidgetView, [2626](#)
  - currentCamItemInfo, [2629](#)
  - currentItemInfo, [2629](#)
  - doLoadState, [2629](#)
  - doSaveState, [2629](#)
  - getActiveState, [2629](#)
  - MapWidgetView, [2628](#)
  - setActive, [2629](#)
- Digikam::Mat, [2630](#)
  - cols, [2630](#)
  - data, [2630](#)
  - rows, [2630](#)
- Digikam::Matrix, [137](#)
- Digikam::MdKeyListViewItem, [2631](#)
- Digikam::MediaPlayerView, [2632](#)
- Digikam::MetadataHub, [2633](#)
  - cleanupTags, [2635](#)
  - FullWrite, [2634](#)
  - FullWriteIfChanged, [2634](#)
  - load, [2635](#)
  - loadTags, [2635](#)
  - MetadataAvailable, [2634](#)
  - MetadataHub, [2634](#)
  - MetadataInvalid, [2634](#)
  - PartialWrite, [2634](#)
  - Status, [2634](#)
  - willWriteMetadata, [2635](#)
  - write, [2636](#), [2637](#)
  - WriteMode, [2634](#)
  - writeTags, [2637](#), [2638](#)
  - writeToBaloo, [2638](#)
  - writeToMetadata, [2638](#)
- Digikam::MetadataHubMngr, [2639](#)
- Digikam::MetadataKeys, [2640](#)
  - getDbValue, [2641](#)
- Digikam::MetadataListView, [2642](#)
- Digikam::MetadataListViewItem, [2643](#)
- Digikam::MetadataOption, [2644](#)
  - parseOperation, [2645](#)
- Digikam::MetadataOptionDialog, [2646](#)
- Digikam::MetadataPage, [2647](#)
- Digikam::MetadataPanel, [2648](#)
- Digikam::MetadataRemover, [2650](#)
  - MetadataRemover, [2653](#)
  - setUseMultiCoreCPU, [2653](#)
- Digikam::MetadataRemoveTask, [2654](#)
- Digikam::MetadataSelector, [2655](#)
- Digikam::MetadataSelectorItem, [2656](#)
- Digikam::MetadataSelectorView, [2657](#)
- Digikam::MetadataStatusBar, [2658](#)
- Digikam::MetadataSynchronizer, [2659](#)
  - MetadataSynchronizer, [2662](#)
  - setUseMultiCoreCPU, [2662](#)
- Digikam::MetadataSyncTask, [2663](#)
- Digikam::MetadataWidget, [2665](#)
  - setup, [2666](#)
- Digikam::MetaEngine, [2667](#)
  - ~MetaEngine, [2674](#)
  - addToXmpTagStringBag, [2674](#)
  - AltLangMap, [2672](#)
  - applyChanges, [2674](#)
  - Backend, [2673](#)
  - backendName, [2675](#)
  - canWriteComment, [2675](#)
  - canWriteExif, [2675](#)
  - canWriteIptc, [2675](#)
  - canWriteXmp, [2675](#)
  - clearComments, [2675](#)
  - clearExif, [2675](#)
  - clearIptc, [2676](#)
  - clearXmp, [2676](#)
  - convertDegreeAngleToDouble, [2676](#)
  - convertFromGPSCoordinateString, [2676](#)
  - convertToGPSCoordinateString, [2676](#), [2677](#)
  - convertToRational, [2677](#)
  - convertToRationalSmallDenominator, [2677](#)
  - convertToUserPresentableNumbers, [2677](#), [2678](#)
  - createExifUserStringFromValue, [2678](#)
  - detectLanguageAlt, [2678](#)
  - ExifToolBackend, [2673](#)
  - Exiv2Backend, [2673](#)
  - Exiv2Version, [2678](#)
  - exportChanges, [2678](#)
  - FFmpegBackend, [2673](#)
  - getComments, [2679](#)
  - getCommentsDecoded, [2679](#)
  - getDigitizationDateTime, [2679](#)
  - getExifComment, [2679](#)
  - getExifEncoded, [2679](#)
  - getExifTagComment, [2679](#)
  - getExifTagData, [2680](#)
  - getExifTagDescription, [2680](#)
  - getExifTagLong, [2680](#)
  - getExifTagRational, [2680](#)
  - getExifTagsDataList, [2680](#)
  - getExifTagString, [2681](#)
  - getExifTagTitle, [2681](#)
  - getExifTagVariant, [2681](#)
  - getExifThumbnail, [2681](#)
  - getFilePath, [2682](#)



getGPSAltitude, 2682  
getGPSInfo, 2682  
getGPSLatitudeNumber, 2682  
getGPSLatitudeString, 2682  
getIptc, 2682  
getIptcKeywords, 2683  
getIptcSubCategories, 2683  
getIptcSubjects, 2683  
getIptcTagData, 2683  
getIptcTagDescription, 2683  
getIptcTagsDataList, 2683  
getIptcTagsList, 2684  
getIptcTagsStringList, 2684  
getIptcTagString, 2684  
getIptcTagTitle, 2684  
getItemColorWorkSpace, 2684  
getItemDateTime, 2685  
getItemDimensions, 2685  
getItemIccProfile, 2685  
getItemOrientation, 2685  
getItemPreview, 2685  
getMakernoteTagsList, 2685  
getMimeType, 2685  
getPixelSize, 2686  
getStdExifTagsList, 2686  
getXmp, 2686  
getXmpKeywords, 2686  
getXmpSubCategories, 2686  
getXmpSubjects, 2686  
getXmpTagDescription, 2686  
getXmpTagsDataList, 2687  
getXmpTagsList, 2687  
getXmpTagString, 2687  
getXmpTagStringBag, 2687  
getXmpTagStringLangAlt, 2687  
getXmpTagStringListLangAlt, 2688  
getXmpTagStringSeq, 2688  
getXmpTagTitle, 2688  
getXmpTagVariant, 2688  
hasComments, 2688  
hasExif, 2689  
hasIptc, 2689  
hasSidecar, 2689  
hasXmp, 2689  
ImageColorWorkSpace, 2673  
ImageMagickBackend, 2673  
ImageOrientation, 2673  
initializeExiv2, 2689  
initializeGPSInfo, 2689  
isEmpty, 2689  
LibHeifBackend, 2673  
LibRawBackend, 2673  
load, 2690  
loadFromData, 2690  
loadFromDataAndMerge, 2690  
loadFromSidecarAndMerge, 2690  
MetaDataMap, 2672  
MetadataWritingMode, 2673  
metadataWritingMode, 2690  
MetaEngine, 2674  
NoBackend, 2673  
readWithExifTool, 2690  
registerXmpNameSpace, 2691  
removeExifTag, 2691  
removeExifThumbnail, 2691  
removeFromXmpTagStringBag, 2691  
removeGPSInfo, 2691  
removeIptcTag, 2691  
removeXmpKeywords, 2692  
removeXmpSubCategories, 2692  
removeXmpSubjects, 2692  
removeXmpTag, 2692  
rotateExifQImage, 2692  
save, 2692  
setComments, 2693  
setExif, 2693  
setExifComment, 2693  
setExifTagData, 2693  
setExifTagLong, 2693  
setExifTagRational, 2693  
setExifTagString, 2694  
setExifTagURational, 2694  
setExifTagUShort, 2694  
setExifTagVariant, 2694  
setExifThumbnail, 2694  
setFilePath, 2695  
setGPSInfo, 2695  
setImageDateTime, 2695  
setIptc, 2695  
setIptcKeywords, 2696  
setIptcSubCategories, 2696  
setIptcSubjects, 2696  
setIptcTagData, 2696  
setIptcTagsStringList, 2696  
setIptcTagString, 2697  
setItemColorWorkSpace, 2697  
setItemDimensions, 2697  
setItemIccProfile, 2697  
setItemOrientation, 2697  
setItemPreview, 2697  
setItemProgramId, 2698  
setMetadataWritingMode, 2698  
setProgramId, 2698  
setReadWithExifTool, 2698  
setTiffThumbnail, 2698  
setUpdateFileTimeStamp, 2699  
setUseCompatibleFileName, 2699  
setUseXMPSidecar4Reading, 2699  
setWriteDngFiles, 2699  
setWriteRawFiles, 2699  
setWriteWithExifTool, 2699  
setXmp, 2699  
setXmpKeywords, 2700  
setXmpSubCategories, 2700  
setXmpSubjects, 2700  
setXmpTagString, 2700

- setXmpTagStringBag, [2700](#)
- setXmpTagStringLangAlt, [2701](#)
- setXmpTagStringListLangAlt, [2701](#)
- setXmpTagStringSeq, [2701](#)
- sidecarFilePathForFile, [2701](#)
- sidecarPath, [2701](#)
- sidecarUrl, [2701](#), [2702](#)
- supportBmff, [2702](#)
- supportJpegXL, [2702](#)
- supportMetadataWriting, [2702](#)
- supportXmp, [2702](#)
- TagsMap, [2672](#)
- unregisterXmpNameSpace, [2702](#)
- updateFileTimeStamp, [2702](#)
- useCompatibleFileName, [2703](#)
- useXMPSidecar4Reading, [2703](#)
- VideoMergeBackend, [2673](#)
- WRITE\_TO\_FILE\_ONLY, [2673](#)
- WRITE\_TO\_SIDE CAR\_AND\_FILE, [2673](#)
- WRITE\_TO\_SIDE CAR\_ONLY, [2673](#)
- WRITE\_TO\_SIDE CAR\_ONLY\_FOR\_READ\_ONLY\_FILES, [filterIdentifier](#), [2723](#)  
[2673](#)
- writeDngFiles, [2703](#)
- writeRawFiles, [2703](#)
- writeWithExifTool, [2703](#)
- XmpTagType, [2674](#)
- Digikam::MetaEngine::Private, [2704](#)
  - convertCommentValue, [2705](#)
  - decodeGPSCoordinate, [2705](#)
  - detectEncodingAndDecode, [2705](#)
  - extractIptcTagString, [2705](#)
  - isUtf8, [2705](#)
  - printExiv2ExceptionError, [2705](#)
  - printExiv2MessageHandler, [2705](#)
- Digikam::MetaEngineData, [2706](#)
- Digikam::MetaEngineData::Private, [2706](#)
- Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >, [2707](#)
  - exclusiveMerge, [2707](#)
  - mergeAll, [2707](#)
  - mergeFields, [2707](#)
- Digikam::MetaEnginePreviews, [2708](#)
  - data, [2709](#)
  - dataSize, [2709](#)
  - image, [2709](#)
  - MetaEnginePreviews, [2708](#)
- Digikam::MetaEngineRotation, [2709](#)
  - exifOrientation, [2711](#)
  - FlipHorizontal, [2710](#)
  - FlipVertical, [2710](#)
  - isNoTransform, [2711](#)
  - MetaEngineRotation, [2710](#)
  - NoTransformation, [2710](#)
  - operator\*=: [2711](#)
  - Rotate180, [2710](#)
  - Rotate270, [2710](#)
  - Rotate90, [2710](#)
  - toTransform, [2711](#)
  - TransformationAction, [2710](#)
  - transformations, [2712](#)
- Digikam::MetaEngineSettings, [2712](#)
  - exifRotate, [2713](#)
  - instance, [2713](#)
  - setSettings, [2713](#)
  - settings, [2713](#)
- Digikam::MetaEngineSettingsContainer, [2713](#)
  - RotationBehaviorFlag, [2715](#)
- Digikam::MigrateFromDigikam4Page, [2716](#)
  - checkForMigration, [2717](#)
  - isMigrationChecked, [2717](#)
- Digikam::MimeFilter, [2718](#)
  - HEIFFiles, [2718](#)
  - RasterGraphics, [2718](#)
  - RAWFiles, [2718](#)
  - TypeMimeFilter, [2718](#)
- Digikam::MixerContainer, [2719](#)
- Digikam::MixerFilter, [2720](#)
  - filterAction, [2723](#)
  - readParameters, [2723](#)
- Digikam::MixerSettings, [2724](#)
- Digikam::MLClassifierFoundation, [2725](#)
- Digikam::MLClassifierFoundation::VotingGroups, [2726](#)
- Digikam::MLClassifierFoundation::VotingGroups::VoteTally, [2726](#)
- Digikam::MLPipelineFoundation, [2727](#)
  - cancel, [2729](#)
  - Classifier, [2729](#)
  - Extractor, [2729](#)
  - Finder, [2729](#)
  - Loader, [2729](#)
  - MLPipelineStage, [2729](#)
  - None, [2729](#)
  - Trainer, [2729](#)
  - Writer, [2729](#)
- Digikam::MLPipelineFoundation::\_MLPipelinePerformanceProfile, [2730](#)
- Digikam::MLPipelinePackageFoundation, [2731](#)
- Digikam::MLPipelinePackageNotify, [2732](#)
- Digikam::ModelCompleter, [2733](#)
  - setItemModel, [2733](#)
- Digikam::ModelIndexBasedComboBox, [2735](#)
  - ModelIndexBasedComboBox, [2736](#)
- Digikam::ModelMenu, [2737](#)
  - prePopulated, [2738](#)
- Digikam::Modifier, [2739](#)
  - parseOperation, [2740](#)
- Digikam::MonthWidget, [2741](#)
- Digikam::MysqlAdminBinary, [2742](#)
- Digikam::MysqlInitBinary, [2745](#)
- Digikam::MysqlServerBinary, [2748](#)
- Digikam::MysqlUpgradeBinary, [2751](#)
- Digikam::NamespaceEditDlg, [2753](#)
- Digikam::NamespaceEntry, [2754](#)
  - convertRatio, [2755](#)
  - namespaceName, [2755](#)

- Digikam::NamespaceListView, [2756](#)
  - startDrag, [2757](#)
- Digikam::NetworkManager, [2757](#)
  - getNetworkManager, [2758](#)
  - instance, [2758](#)
- Digikam::NewItemFinder, [2759](#)
  - CompleteCollectionScan, [2762](#)
  - FinderMode, [2761](#)
  - ScanDeferredFiles, [2762](#)
  - ScheduleCollectionScan, [2762](#)
- Digikam::NewlyAppearedFile, [2762](#)
- Digikam::NoDuplicatesImportFilterModel, [2763](#)
- Digikam::NoDuplicatesItemFilterModel, [2766](#)
- Digikam::NoiseDetector, [2768](#)
  - detect, [2769](#)
- Digikam::NonDeterministicRandomData, [2769](#)
  - NonDeterministicRandomData, [2770](#)
- Digikam::NormalizeFilter, [2771](#)
  - filterAction, [2774](#)
  - filterIdentifier, [2774](#)
  - readParameters, [2774](#)
- Digikam::NormalSearchTreeView, [2775](#)
  - ~NormalSearchTreeView, [2780](#)
  - addCustomContextMenuActions, [2780](#)
  - copySearch, [2780](#)
  - editSearch, [2780](#)
  - handleCustomContextMenuAction, [2780](#)
  - newSearch, [2781](#)
  - NormalSearchTreeView, [2779](#)
- Digikam::NRContainer, [2781](#)
  - thresholds, [2781](#)
- Digikam::NREstimate, [2782](#)
  - NREstimate, [2785](#)
  - setLogFilesPath, [2785](#)
  - settings, [2785](#)
  - startAnalyse, [2785](#)
- Digikam::NRFilter, [2786](#)
  - filterAction, [2789](#)
  - filterIdentifier, [2789](#)
  - readParameters, [2789](#)
- Digikam::NRSettings, [2790](#)
- Digikam::OilPaintFilter, [2791](#)
  - filterAction, [2794](#)
  - filterIdentifier, [2794](#)
  - readParameters, [2794](#)
- Digikam::OnlineVersionChecker, [2794](#)
  - bundleProperties, [2795](#)
  - lastCheckDate, [2795](#)
- Digikam::OnlineVersionDlg, [2796](#)
- Digikam::OnlineVersionDwnl, [2797](#)
- Digikam::OpenCVDNNFaceDetector, [2797](#)
  - detectFaces, [2798](#)
  - recommendedImageSizeForDetection, [2798](#)
- Digikam::OpenCVDNNFaceRecognizer, [2799](#)
  - Classifier, [2799](#)
  - clearTraining, [2800](#)
  - DB, [2800](#)
  - OpenCV\_KNN, [2800](#)
  - prepareForRecognition, [2800](#)
  - recognize, [2800](#)
  - remove, [2800](#)
  - setNbNeighbors, [2801](#)
  - setThreshold, [2801](#)
  - SVM, [2800](#)
  - train, [2801](#)
  - Tree, [2800](#)
- Digikam::OpenCVDNNFaceRecognizer::Private, [2801](#)
- Digikam::OpenCVDNNFaceRecognizer::Private::ParallelRecognizer, [2802](#)
- Digikam::OpenCVDNNFaceRecognizer::Private::ParallelTrainer, [2803](#)
- Digikam::OpenfacePreprocessor, [2803](#)
  - loadModels, [2803](#)
- Digikam::OpenFilePage, [2804](#)
- Digikam::Option, [2805](#)
  - parseOperation, [2806](#)
- Digikam::OverlayWidget, [2808](#)
- Digikam::PackageLoadingDescriptionList, [2809](#)
- Digikam::PageItem, [2810](#)
- Digikam::PALbum, [2811](#)
  - databaseUrl, [2813](#)
- Digikam::PALbumPath, [2813](#)
- Digikam::PanIconFrame, [2814](#)
  - close, [2815](#)
  - exec, [2815](#)
  - keyPressEvent, [2815](#)
  - popup, [2815](#)
  - resizeEvent, [2816](#)
  - setMainWidget, [2816](#)
- Digikam::PanIconWidget, [2817](#)
  - regionSelectionMoved, [2818](#)
  - signalSelectionMoved, [2818](#)
- Digikam::ParallelAdapter< A >, [2819](#)
  - asQObject, [2821](#)
  - mocMetaObject, [2821](#)
  - ParallelAdapter, [2821](#)
  - staticMetacallPointer, [2821](#)
  - WorkerObjectQtMetacall, [2821](#)
- Digikam::ParallelPipes, [2822](#)
- Digikam::ParallelWorkers, [2824](#)
  - asQObject, [2825](#)
  - mocMetaObject, [2825](#)
  - optimalWorkerCount, [2826](#)
  - optimalWorkerCountReached, [2826](#)
  - ParallelWorkers, [2825](#)
  - replacementQtMetacall, [2826](#)
  - schedule, [2826](#)
  - WorkerObjectQtMetacall, [2826](#)
- Digikam::Parser, [2827](#)
  - parseStringsValid, [2828](#)
- Digikam::ParseResults, [2828](#)
- Digikam::ParseSettings, [2829](#)
- Digikam::PeopleSideBarWidget, [2830](#)
  - applySettings, [2832](#)
  - changeAlbumFromHistory, [2832](#)
  - doLoadState, [2832](#)

- doSaveState, [2832](#)
- getCaption, [2832](#)
- getIcon, [2832](#)
- setActive, [2833](#)
- Digikam::PersistentWidgetDelegateOverlay, [2834](#)
  - hide, [2836](#)
  - PersistentWidgetDelegateOverlay, [2836](#)
  - setActive, [2836](#)
  - setFocusOnWidget, [2837](#)
  - setPersistent, [2837](#)
  - showOnIndex, [2837](#)
  - slotEntered, [2837](#)
  - slotLayoutChanged, [2837](#)
  - slotReset, [2837](#)
  - slotRowsRemoved, [2838](#)
  - slotViewportEntered, [2838](#)
  - viewportLeaveEvent, [2838](#)
- Digikam::PhotoInfoContainer, [2838](#)
- Digikam::PickLabelFilter, [2840](#)
- Digikam::PickLabelMenuAction, [2842](#)
- Digikam::PickLabelSelector, [2843](#)
- Digikam::PickLabelWidget, [2844](#)
  - pickLabels, [2845](#)
  - setButtonsExclusive, [2845](#)
  - setDescriptionBoxVisible, [2846](#)
  - setPickLabels, [2846](#)
- Digikam::PlaceholderWidget, [2846](#)
- Digikam::PointTransformAffine, [2847](#)
- Digikam::PositionKeys, [2847](#)
  - getDbValue, [2848](#)
  - PositionKeys, [2848](#)
- Digikam::PreviewList, [2849](#)
- Digikam::PreviewListItem, [2850](#)
- Digikam::PreviewLoadingTask, [2851](#)
  - execute, [2853](#)
- Digikam::PreviewLoadThread, [2854](#)
  - load, [2858](#)
  - loadFast, [2858](#)
  - loadFastButLarge, [2859](#)
  - loadFastSynchronously, [2859](#)
  - loadHighQuality, [2859](#)
  - PreviewLoadThread, [2858](#)
  - setDisplayingWidget, [2859](#)
- Digikam::PreviewPage, [2860](#)
- Digikam::PreviewSettings, [2861](#)
  - FastButLargePreview, [2861](#)
  - FastPreview, [2861](#)
  - HighQualityPreview, [2861](#)
  - Quality, [2861](#)
- Digikam::PreviewThreadWrapper, [2862](#)
- Digikam::PreviewToolBar, [2863](#)
  - NoPreviewMode, [2864](#)
  - PreviewBothImagesHorz, [2864](#)
  - PreviewBothImagesHorzCont, [2864](#)
  - PreviewBothImagesVert, [2864](#)
  - PreviewBothImagesVertCont, [2864](#)
  - PreviewMode, [2864](#)
  - PreviewOriginalImage, [2864](#)
  - PreviewTargetImage, [2864](#)
  - PreviewToggleOnMouseOver, [2864](#)
- Digikam::PrivateProgressItemCreator, [2864](#)
  - addProgressItem, [2865](#)
  - createProgressItem, [2865](#)
- Digikam::ProcessLauncher, [2866](#)
  - elapsedTime, [2867](#)
  - exitCode, [2867](#)
  - output, [2867](#)
  - setConsoleTraces, [2867](#)
  - startProcess, [2867](#)
  - success, [2867](#)
- Digikam::ProgressEntry, [2868](#)
- Digikam::ProgressItem, [2868](#)
  - advance, [2870](#)
  - canBeCanceled, [2870](#)
  - hasThumbnail, [2870](#)
  - id, [2870](#)
  - label, [2870](#)
  - parent, [2870](#)
  - progress, [2871](#)
  - progressItemAdded, [2871](#)
  - progressItemCanceled, [2871](#)
  - progressItemCompleted, [2871](#)
  - progressItemLabel, [2871](#)
  - progressItemProgress, [2872](#)
  - progressItemStatus, [2872](#)
  - progressItemThumbnail, [2872](#)
  - progressItemUsesBusyIndicator, [2872](#)
  - setLabel, [2873](#)
  - setProgress, [2873](#)
  - setShowAtStart, [2873](#)
  - setStatus, [2873](#)
  - setThumbnail, [2874](#)
  - setUsesBusyIndicator, [2874](#)
  - showAtStart, [2874](#)
  - status, [2874](#)
  - usesBusyIndicator, [2874](#)
- Digikam::ProgressManager, [2875](#)
  - addProgressItem, [2877](#)
  - createProgressItem, [2877](#), [2878](#)
  - findItembyId, [2879](#)
  - getUniqueID, [2879](#)
  - instance, [2879](#)
  - isEmpty, [2879](#)
  - progressItemAdded, [2880](#)
  - progressItemCanceled, [2880](#)
  - progressItemCompleted, [2880](#)
  - progressItemLabel, [2880](#)
  - progressItemProgress, [2880](#)
  - progressItemStatus, [2881](#)
  - progressItemThumbnail, [2881](#)
  - progressItemUsesBusyIndicator, [2881](#)
  - singleItem, [2881](#)
  - slotStandardCancelHandler, [2881](#)
- Digikam::ProgressView, [2882](#)
- Digikam::ProxyClickLineEdit, [2885](#)
  - ProxyClickLineEdit, [2886](#)

- Digikam::ProxyLineEdit, 2887
  - mouseMoveEvent, 2888
  - mousePressEvent, 2888
  - ProxyLineEdit, 2888
  - setWidget, 2888
- Digikam::QListImageListProvider, 2889
  - atEnd, 2890
  - image, 2890
  - images, 2890
  - proceed, 2890
  - setImages, 2890
  - setUnpairedImages, 2891
  - size, 2891
- Digikam::QMapForAdaptors< Key, Value >, 2891
- Digikam::QueueListView, 2892
  - All, 2893
  - ItemListType, 2893
  - Pending, 2893
  - Selected, 2893
- Digikam::QueueListViewItem, 2894
- Digikam::QueueMgrWindow, 2895
  - infolface, 2897
  - queuesMap, 2897
- Digikam::QueueMgrWindow::Private, 2898
- Digikam::QueuePool, 2899
  - applySettings, 2900
- Digikam::QueuePoolBar, 2901
- Digikam::QueueSettings, 2901
- Digikam::QueueSettingsView, 2902
- Digikam::QueueToolTip, 2903
- Digikam::RadioButtonHBox, 2904
- Digikam::RainDropFilter, 2905
  - filterAction, 2908
  - filterIdentifier, 2908
  - readParameters, 2908
- Digikam::RandomNumberGenerator, 2908
  - currentSeed, 2909
  - nonDeterministicSeed, 2909
  - number, 2909
  - RandomNumberGenerator, 2909
  - reseed, 2910
  - seed, 2910
  - seedByTime, 2910
  - seedNonDeterministic, 2910
  - timeSeed, 2910
  - yesOrNo, 2910
- Digikam::RangeDialog, 2911
- Digikam::RangeModifier, 2912
  - parseOperation, 2914
- Digikam::RatingBox, 2915
- Digikam::RatingComboBox, 2917
  - Null, 2918
  - RatingValue, 2918
- Digikam::RatingComboBoxDelegate, 2919
- Digikam::RatingComboBoxModel, 2920
- Digikam::RatingComboBoxWidget, 2921
  - RatingComboBoxWidget, 2923
- Digikam::RatingFilter, 2924
- Digikam::RatingFilterWidget, 2926
- Digikam::RatingMenuAction, 2928
- Digikam::RatingStarDrawer, 2929
- Digikam::RatingWidget, 2930
  - starPolygon, 2931
- Digikam::RawCameraDlg, 2932
- Digikam::RawPage, 2933
- Digikam::RawProcessingFilter, 2935
  - filterAction, 2939
  - filterIdentifier, 2939
  - filterImage, 2939
  - RawProcessingFilter, 2938
  - readParameters, 2939
  - setObserver, 2939
  - setOutputProfile, 2940
  - setSettings, 2940
- Digikam::RecognitionBenchmark, 2941
  - result, 2943
- Digikam::RecognitionBenchmark::Statistics, 2943
- Digikam::RecognitionPreprocessor, 2944
  - preprocess, 2944
- Digikam::RecognitionTrainingProvider, 2945
  - images, 2946
  - newImages, 2946
- Digikam::RecognitionTrainingUpdateQueue, 2946
- Digikam::RecognitionWorker, 2947
  - aboutToDeactivate, 2949
  - process, 2949
- Digikam::RedEye::RegressionTree, 2949
  - operator(), 2950
- Digikam::RedEye::ShapePredictor, 2950
- Digikam::RedEye::SplitFeature, 2951
- Digikam::RedEyeCorrectionContainer, 2951
- Digikam::RedEyeCorrectionFilter, 2952
  - filterAction, 2955
  - filterIdentifier, 2955
  - RedEyeCorrectionFilter, 2955
- Digikam::RedEyeCorrectionSettings, 2956
- Digikam::RefocusFilter, 2957
  - filterAction, 2960
  - filterIdentifier, 2960
  - readParameters, 2960
- Digikam::RefocusMatrix, 2960
- Digikam::RegionFrameItem, 2961
  - setHudWidget, 2963
  - setViewportRect, 2963
- Digikam::RemoveBookmarksCommand, 2964
- Digikam::RemoveDoublesModifier, 2966
  - parseOperation, 2967
- Digikam::RemoveFilterAction, 2968
- Digikam::RenameCustomizer, 2969
- Digikam::RenameFileJob, 2970
- Digikam::ReplaceDialog, 2972
- Digikam::ReplaceModifier, 2973
  - parseOperation, 2974
- Digikam::RestoreDTrashItemsJob, 2975
- Digikam::RGBBackend, 2977
  - backendName, 2978

- callRGBBackend, 2978
- getErrorMessage, 2978
- RGBBackend, 2977
- Digikam::RGInfo, 2978
  - ~RGInfo, 2978
  - coordinates, 2979
  - id, 2979
  - rgData, 2979
  - RGInfo, 2978
- Digikam::RGTagModel, 2979
  - addDataInTree, 2983
  - addExternalTags, 2983
  - addNewData, 2983
  - addNewTag, 2983
  - addSpacerTag, 2984
  - branchFromIndex, 2984
  - climbTreeAndGetSpacers, 2984
  - deleteAllSpacersOrNewTags, 2984
  - deleteTag, 2985
  - findAndDeleteSpacersOrNewTags, 2985
  - fromSourceIndex, 2985
  - getSpacerAddress, 2985
  - getSpacers, 2987
  - getTagType, 2987
  - readNewTags, 2987
  - readTag, 2987
  - RGTagModel, 2982
  - toSourceIndex, 2988
- Digikam::RGWidget, 2988
  - ~RGWidget, 2989
  - eventFilter, 2990
  - readSettingsFromGroup, 2990
  - RGWidget, 2989
  - saveSettingsToGroup, 2990
  - setUIEnabled, 2990
  - signalProgressChanged, 2990
  - signalProgressSetup, 2991
  - signalSetUIEnabled, 2991
  - signalUndoCommand, 2991
- Digikam::RubberItem, 2992
- Digikam::Rule, 2995
  - addToken, 2996
  - escapeToken, 2996
  - isValid, 2996
  - parseOperation, 2997
  - regExp, 2997
  - registerButton, 2997
  - registerMenu, 2999
  - reset, 2999
  - setUseTokenMenu, 2999
  - tokens, 3000
  - useTokenMenu, 3000
- Digikam::RuleDialog, 3000
- Digikam::RuleType, 3000
- Digikam::RuleTypeForConversion, 3000
- Digikam::SafeTemporaryFile, 3001
- Digikam::SAlbum, 3002
  - databaseUrl, 3004
  - getTemporaryHaarTitle, 3004
  - getTemporaryTitle, 3005
  - isTemporarySearch, 3005
- Digikam::SaveProperties, 3005
- Digikam::SavingContext, 3005
- Digikam::SavingTask, 3006
  - continueQuery, 3007
  - execute, 3007
  - progressInfo, 3007
  - type, 3008
- Digikam::ScanController, 3009
  - abortInitialization, 3011
  - beginFileMetadataWrite, 3011
  - cancelAllAndSuspendCollectionScan, 3011
  - cancelCompleteScan, 3011
  - completeCollectionScan, 3011
  - completeCollectionScanInBackground, 3012
  - databaseInitialization, 3012
  - finishFileMetadataWrite, 3012
  - getNewIdsList, 3012
  - hintAtModificationOfItems, 3012
  - hintAtMoveOrCopyOfAlbum, 3012
  - hintAtMoveOrCopyOfItems, 3013
  - restart, 3013
  - restartCollectionScan, 3013
  - resumeCollectionScan, 3013
  - scannedInfo, 3013
  - scheduleCollectionScan, 3013
  - scheduleCollectionScanExternal, 3014
  - scheduleCollectionScanRelaxed, 3014
  - shutDown, 3014
  - suspendCollectionScan, 3014
  - updateUniqueHash, 3014
- Digikam::ScanController::FileMetadataWrite, 3015
- Digikam::ScanController::Private, 3015
- Digikam::ScanControllerCreator, 3016
- Digikam::ScanControllerLoadingCacheFileWatch, 3017
- Digikam::ScanStateFilter, 3018
  - run, 3020
- Digikam::SchemeManager, 3020
  - ~SchemeManager, 3024
  - ActiveBackground, 3022
  - ActiveText, 3024
  - adjustBackground, 3025
  - adjustForeground, 3025
  - AlternateBackground, 3022
  - background, 3025
  - BackgroundRole, 3022
  - Button, 3022
  - ColorSet, 3022
  - Complementary, 3023
  - contrast, 3026
  - contrastF, 3026
  - createApplicationPalette, 3026
  - DarkShade, 3024
  - decoration, 3026
  - DecorationRole, 3023
  - FocusColor, 3023

- foreground, [3027](#)
- ForegroundRole, [3023](#)
- HoverColor, [3023](#)
- InactiveText, [3024](#)
- LightShade, [3024](#)
- LinkBackground, [3022](#)
- LinkText, [3024](#)
- MidlightShade, [3024](#)
- MidShade, [3024](#)
- NegativeBackground, [3022](#)
- NegativeText, [3024](#)
- NeutralBackground, [3022](#)
- NeutralText, [3024](#)
- NormalBackground, [3022](#)
- NormalText, [3024](#)
- operator=, [3027](#)
- PositiveBackground, [3022](#)
- PositiveText, [3024](#)
- SchemeManager, [3024](#)
- Selection, [3023](#)
- shade, [3027](#), [3028](#)
- ShadeRole, [3024](#)
- ShadowShade, [3024](#)
- Tooltip, [3023](#)
- View, [3022](#)
- VisitedBackground, [3022](#)
- VisitedText, [3024](#)
- Window, [3022](#)
- Digikam::ScriptingSettings, [3028](#)
- Digikam::SearchChangeset, [3029](#)
- Digikam::SearchesDBJobInfo, [3030](#)
- Digikam::SearchesDBJobsThread, [3032](#)
  - searchesListing, [3034](#)
- Digikam::SearchesJob, [3035](#)
- Digikam::SearchField, [3036](#)
  - createField, [3038](#)
  - isVisible, [3038](#)
  - setVisible, [3038](#)
  - write, [3038](#)
- Digikam::SearchFieldAlbum, [3039](#)
  - read, [3041](#)
  - reset, [3041](#)
  - setupValueWidgets, [3041](#)
  - setValueWidgetsVisible, [3041](#)
  - valueWidgetRects, [3041](#)
  - write, [3042](#)
- Digikam::SearchFieldCheckBox, [3043](#)
  - read, [3045](#)
  - reset, [3045](#)
  - setupValueWidgets, [3045](#)
  - setValueWidgetsVisible, [3045](#)
  - valueWidgetRects, [3045](#)
  - write, [3046](#)
- Digikam::SearchFieldChoice, [3047](#)
  - read, [3049](#)
  - reset, [3049](#)
  - setupValueWidgets, [3049](#)
  - setValueWidgetsVisible, [3049](#)
- valueWidgetRects, [3050](#)
- write, [3050](#)
- Digikam::SearchFieldColorDepth, [3051](#)
  - read, [3053](#)
  - setupValueWidgets, [3053](#)
- Digikam::SearchFieldComboBox, [3054](#)
  - reset, [3056](#)
  - setupValueWidgets, [3056](#)
  - setValueWidgetsVisible, [3056](#)
  - valueWidgetRects, [3056](#)
  - write, [3056](#)
- Digikam::SearchFieldGroup, [3057](#)
- Digikam::SearchFieldGroupLabel, [3058](#)
- Digikam::SearchFieldKeyword, [3060](#)
  - read, [3062](#)
  - write, [3062](#)
- Digikam::SearchFieldLabels, [3063](#)
  - read, [3065](#)
  - reset, [3065](#)
  - setupValueWidgets, [3065](#)
  - setValueWidgetsVisible, [3065](#)
  - valueWidgetRects, [3065](#)
  - write, [3066](#)
- Digikam::SearchFieldMonthDay, [3067](#)
  - read, [3069](#)
  - reset, [3069](#)
  - setupValueWidgets, [3069](#)
  - setValueWidgetsVisible, [3069](#)
  - valueWidgetRects, [3069](#)
  - write, [3070](#)
- Digikam::SearchFieldPageOrientation, [3071](#)
  - read, [3073](#)
  - setupValueWidgets, [3073](#)
- Digikam::SearchFieldRangeDate, [3074](#)
  - read, [3076](#)
  - reset, [3076](#)
  - setupValueWidgets, [3076](#)
  - setValueWidgetsVisible, [3076](#)
  - valueWidgetRects, [3077](#)
  - write, [3077](#)
- Digikam::SearchFieldRangeDouble, [3078](#)
  - read, [3080](#)
  - reset, [3080](#)
  - setupValueWidgets, [3080](#)
  - setValueWidgetsVisible, [3080](#)
  - valueWidgetRects, [3081](#)
  - write, [3081](#)
- Digikam::SearchFieldRangeInt, [3082](#)
  - read, [3084](#)
  - reset, [3084](#)
  - setupValueWidgets, [3084](#)
  - setValueWidgetsVisible, [3084](#)
  - valueWidgetRects, [3085](#)
  - write, [3085](#)
- Digikam::SearchFieldRangeTime, [3086](#)
  - read, [3088](#)
  - reset, [3088](#)
  - setupValueWidgets, [3088](#)

- setValueWidgetsVisible, 3088
  - valueWidgetRects, 3088
  - write, 3089
- Digikam::SearchFieldRating, 3090
  - read, 3092
  - reset, 3092
  - setValueWidgets, 3092
  - setValueWidgetsVisible, 3092
  - valueWidgetRects, 3092
  - write, 3093
- Digikam::SearchFieldText, 3094
  - read, 3096
  - reset, 3096
  - setValueWidgets, 3096
  - setValueWidgetsVisible, 3096
  - valueWidgetRects, 3096
  - write, 3097
- Digikam::SearchFilterModel, 3098
  - isFiltering, 3101
  - matches, 3101
  - setFilterSearchType, 3101
  - setListTemporarySearches, 3101
- Digikam::SearchGroup, 3102
  - addGroupToLayout, 3104
  - createSearchGroup, 3104
- Digikam::SearchGroupLabel, 3105
- Digikam::SearchInfo, 3106
  - operator<, 3106
- Digikam::SearchModel, 3107
  - albumData, 3112
  - albumForId, 3112
  - SearchModel, 3112
  - setPixmapForNormalSearches, 3113
  - setReplaceNames, 3113
- Digikam::SearchModificationHelper, 3114
  - ~SearchModificationHelper, 3115
  - createFuzzySearchFromDropped, 3115
  - createFuzzySearchFromImage, 3116
  - createFuzzySearchFromSketch, 3116
  - SearchModificationHelper, 3115
  - slotCreateFuzzySearchFromDropped, 3116
  - slotCreateFuzzySearchFromImage, 3117
  - slotCreateFuzzySearchFromSketch, 3117
  - slotCreateTimeLineSearch, 3118
  - slotSearchDelete, 3118
  - slotSearchRename, 3118
- Digikam::SearchSideBarWidget, 3119
  - applySettings, 3120
  - changeAlbumFromHistory, 3120
  - doLoadState, 3121
  - doSaveState, 3121
  - getCaption, 3121
  - getIcon, 3121
  - setActive, 3121
- Digikam::SearchTabHeader, 3122
- Digikam::SearchTextBar, 3123
  - doLoadState, 3125
  - doSaveState, 3125
  - getCurrentHighlightState, 3125
  - HAS\_RESULT, 3125
  - HighlightState, 3125
  - NEUTRAL, 3125
  - NO\_RESULT, 3125
  - setCaseSensitive, 3125
  - setHighlightOnResult, 3126
- Digikam::SearchTextBarDb, 3127
  - setFilterModel, 3129
  - setModel, 3129
- Digikam::SearchTextFilterSettings, 3130
- Digikam::SearchTextSettings, 3131
- Digikam::SearchTreeView, 3132
- Digikam::SearchView, 3137
  - addGroupToLayout, 3139
  - bottomBarPixmap, 3139
  - createSearchGroup, 3139
  - groupLabelPixmap, 3139
- Digikam::SearchViewBottomBar, 3140
- Digikam::SearchViewThemedPartsCache, 3141
- Digikam::SearchWindow, 3142
  - readSearch, 3143
  - reset, 3143
  - search, 3143
  - searchEdited, 3143
  - SearchWindow, 3143
- Digikam::SearchXmlCachingReader, 3144
  - SearchXmlCachingReader, 3146
- Digikam::SearchXmlReader, 3147
  - defaultFieldOperator, 3148
  - fieldOperator, 3148
  - groupCaption, 3148
  - groupOperator, 3149
  - isFieldElement, 3149
  - isGroupElement, 3149
  - readNext, 3149
  - readToEndOfElement, 3149
  - readToFirstField, 3149
  - readToStartOfElement, 3149
  - value, 3150
- Digikam::SearchXmlWriter, 3151
  - finish, 3152
  - finishField, 3152
  - finishGroup, 3153
  - keywordSearch, 3153
  - SearchXmlWriter, 3152
  - setDefaultFieldOperator, 3153
  - setFieldOperator, 3153
  - setGroupCaption, 3153
  - setGroupOperator, 3153
  - writeField, 3153
  - writeGroup, 3154
  - writeValue, 3154
  - xml, 3154
- Digikam::SequenceNumberDialog, 3155
- Digikam::SequenceNumberOption, 3156
  - parseOperation, 3157
- Digikam::Setup, 3159



- execDialog, [3161](#)
- execSinglePage, [3161](#)
- Digikam::SetupAlbumView, [3162](#)
- Digikam::SetupCamera, [3163](#)
- Digikam::SetupCategory, [3164](#)
- Digikam::SetupCollectionDelegate, [3165](#)
  - createItemWidgets, [3167](#)
  - updateItemWidgets, [3167](#)
- Digikam::SetupCollectionModel, [3169](#)
  - AppendDecorationRole, [3171](#)
  - CategoryButtonDisplayRole, [3171](#)
  - DeleteDecorationRole, [3171](#)
  - IsAppendRole, [3171](#)
  - IsCategoryRole, [3171](#)
  - IsDeleteRole, [3171](#)
  - IsUpdateRole, [3171](#)
  - SetupCollectionDataRole, [3171](#)
  - SetupCollectionModel, [3172](#)
  - slotAppendPressed, [3172](#)
  - slotCategoryButtonPressed, [3172](#)
  - UpdateDecorationRole, [3171](#)
- Digikam::SetupCollectionModel::Item, [3172](#)
- Digikam::SetupCollections, [3173](#)
- Digikam::SetupCollectionTreeView, [3174](#)
- Digikam::SetupDatabase, [3175](#)
- Digikam::SetupEditor, [3176](#)
- Digikam::SetupEditorIface, [3177](#)
- Digikam::SetupGeolocation, [3178](#)
- Digikam::SetupICC, [3179](#)
  - SetupICC, [3179](#)
- Digikam::SetupImageQualitySorter, [3180](#)
- Digikam::SetupIOFiles, [3181](#)
- Digikam::SetupLightTable, [3181](#)
- Digikam::SetupMetadata, [3182](#)
- Digikam::SetupMetadata::Private, [3183](#)
- Digikam::SetupMime, [3184](#)
- Digikam::SetupMisc, [3185](#)
- Digikam::SetupMisc::Private, [3186](#)
- Digikam::SetupPlugins, [3187](#)
- Digikam::SetupRaw, [3188](#)
- Digikam::SetupTemplate, [3189](#)
- Digikam::SetupToolTip, [3190](#)
- Digikam::SetupVersioning, [3191](#)
- Digikam::SharedLoadingTask, [3192](#)
  - accessMode, [3194](#)
  - addListener, [3194](#)
  - cacheKey, [3194](#)
  - completed, [3194](#)
  - execute, [3194](#)
  - loadSaveNotifier, [3194](#)
  - notifyNewLoadingProcess, [3195](#)
  - progressInfo, [3195](#)
  - querySendNotifyEvent, [3195](#)
  - removeListener, [3195](#)
  - setResult, [3195](#)
- Digikam::SharedLoadSaveThread, [3196](#)
- Digikam::SharedQueue< T >, [3199](#)
- Digikam::SharpContainer, [3199](#)
- Digikam::SharpenFilter, [3201](#)
  - filterAction, [3204](#)
  - filterIdentifier, [3204](#)
  - readParameters, [3204](#)
  - SharpenFilter, [3204](#)
- Digikam::SharpSettings, [3205](#)
- Digikam::ShearFilter, [3206](#)
  - filterAction, [3209](#)
  - filterIdentifier, [3209](#)
  - readParameters, [3209](#)
- Digikam::ShowHideVersionsOverlay, [3210](#)
  - checkIndex, [3213](#)
  - createButton, [3213](#)
  - setActive, [3213](#)
  - updateButton, [3213](#)
- Digikam::Sidebar, [3214](#)
  - activeNextTab, [3217](#)
  - activePreviousTab, [3217](#)
  - appendTab, [3217](#)
  - backup, [3217](#)
  - deleteTab, [3218](#)
  - doLoadState, [3218](#)
  - doSaveState, [3218](#)
  - expand, [3218](#)
  - getActiveTab, [3218](#)
  - isExpanded, [3218](#)
  - restore, [3218](#), [3219](#)
  - setActiveTab, [3219](#)
  - shrink, [3219](#)
  - Sidebar, [3216](#)
  - signalChangedTab, [3219](#)
  - signalViewChanged, [3219](#)
- Digikam::Sidebar::Private, [3220](#)
  - isMinimized, [3220](#)
- Digikam::SidebarSplitter, [3221](#)
  - restoreState, [3222](#)
  - saveState, [3222](#), [3223](#)
  - setSize, [3223](#)
  - SidebarSplitter, [3222](#)
  - size, [3223](#)
- Digikam::SidebarSplitter::Private, [3223](#)
- Digikam::SidebarState, [3223](#)
- Digikam::SidebarWidget, [3224](#)
  - ~SidebarWidget, [3225](#)
  - applySettings, [3225](#)
  - changeAlbumFromHistory, [3225](#)
  - getCaption, [3226](#)
  - getIcon, [3226](#)
  - requestActiveTab, [3226](#)
  - setActive, [3226](#)
  - SidebarWidget, [3225](#)
  - signalNotificationError, [3227](#)
- Digikam::SidecarFinder, [3227](#)
- Digikam::SimilarityDb, [3227](#)
  - clearImageSimilarity, [3228](#)
  - copySimilarityAttributes, [3228](#)
  - getDirtyOrMissingFingerprints, [3228](#)
  - getDirtyOrMissingFingerprintURLs, [3229](#)

- getImageSimilarity, [3229](#)
- getImageSimilarityAlgorithms, [3229](#)
- getLegacySetting, [3229](#)
- getSetting, [3230](#)
- hasDirtyOrMissingFingerprint, [3230](#)
- hasFingerprint, [3230](#)
- hasFingerprints, [3231](#)
- integrityCheck, [3231](#)
- registeredImageIds, [3232](#)
- removeImageFingerprint, [3232](#)
- removeImageSimilarity, [3232](#)
- setSetting, [3233](#)
- vacuum, [3233](#)
- Digikam::SimilarityDbAccess, [3233](#)
  - checkReadyForUse, [3234](#)
  - cleanUpDatabase, [3234](#)
  - initDbEngineErrorHandler, [3234](#)
  - isInitialized, [3234](#)
  - parameters, [3235](#)
  - setLastError, [3235](#)
  - setParameters, [3235](#)
  - SimilarityDbAccess, [3234](#)
- Digikam::SimilarityDbBackend, [3236](#)
  - initSchema, [3239](#)
- Digikam::SimilarityDbSchemaUpdater, [3239](#)
- Digikam::SimpleCollectionScannerObserver, [3239](#)
  - continueQuery, [3240](#)
- Digikam::SimpleTreeModel, [3240](#)
- Digikam::SimpleTreeModel::Item, [3241](#)
- Digikam::SinglePhotoPreviewLayout, [3242](#)
  - addItem, [3243](#)
  - maxZoomFactor, [3243](#)
  - setGraphicsView, [3244](#)
  - setScaleFitToWindow, [3244](#)
- Digikam::SketchWidget, [3245](#)
  - setSketchImageFromXML, [3246](#)
  - sketchImageToXML, [3246](#)
- Digikam::SlideVideo, [3247](#)
- Digikam::SoftProofDialog, [3248](#)
- Digikam::SolidHardwareDlg, [3249](#)
- Digikam::SolidVolumeInfo, [3250](#)
- Digikam::SparseModelIndexVector, [3250](#)
- Digikam::SpellCheckConfig, [3251](#)
- Digikam::SqueezedComboBox, [3252](#)
  - ~SqueezedComboBox, [3253](#)
  - addSqueezedItem, [3253](#)
  - contains, [3254](#)
  - findOriginalText, [3254](#)
  - insertSqueezedItem, [3254](#)
  - insertSqueezedList, [3254](#)
  - item, [3255](#)
  - itemHighlighted, [3255](#)
  - setCurrent, [3255](#)
  - sizeHint, [3256](#)
  - SqueezedComboBox, [3253](#)
- Digikam::StackedView, [3256](#)
  - isInSingleFileMode, [3257](#)
- Digikam::StartScanPage, [3258](#)
- Digikam::State, [3259](#)
- Digikam::StateSavingObject, [3259](#)
  - ~StateSavingObject, [3261](#)
  - DIRECT\_CHILDREN, [3261](#)
  - doLoadState, [3261](#)
  - doSaveState, [3261](#)
  - entryName, [3262](#)
  - getConfigGroup, [3262](#)
  - getStateSavingDepth, [3262](#)
  - INSTANCE, [3261](#)
  - loadState, [3262](#)
  - RECURSIVE, [3261](#)
  - saveState, [3263](#)
  - setConfigGroup, [3263](#)
  - setEntryPrefix, [3263](#)
  - setStateSavingDepth, [3263](#)
  - StateSavingDepth, [3260](#)
  - StateSavingObject, [3261](#)
- Digikam::StatusBarProgressWidget, [3264](#)
- Digikam::StatusProgressBar, [3265](#)
- Digikam::StayPoppedUpComboBox, [3267](#)
  - installView, [3268](#)
  - sendViewportEventToView, [3268](#)
  - StayPoppedUpComboBox, [3268](#)
- Digikam::StretchFilter, [3270](#)
  - filterAction, [3273](#)
  - filterIdentifier, [3273](#)
  - readParameters, [3273](#)
- Digikam::StyleSheetDebugger, [3273](#)
  - StyleSheetDebugger, [3274](#)
- Digikam::SubjectData, [3274](#)
- Digikam::SubjectEdit, [3275](#)
- Digikam::SubjectWidget, [3277](#)
- Digikam::SubQueryBuilder, [3278](#)
- Digikam::SyncJob, [3279](#)
  - getTagThumbnail, [3279](#)
- Digikam::SystemSettings, [3279](#)
  - HttpProxy, [3280](#)
  - ProxyType, [3280](#)
  - Socks5Proxy, [3280](#)
- Digikam::SystemSettingsWidget, [3281](#)
- Digikam::TableView, [3282](#)
  - doLoadState, [3284](#)
  - doSaveState, [3284](#)
  - invertSelection, [3284](#)
  - selectAll, [3284](#)
  - slotAwayFromSelection, [3285](#)
  - slotDeleteSelected, [3285](#)
  - slotSetCurrentWhenAvailable, [3285](#)
- Digikam::TableViewColumn, [3285](#)
  - columnAffectedByChangeset, [3286](#)
  - compare, [3286](#)
  - data, [3287](#)
  - getColumnFlags, [3287](#)
  - paint, [3287](#)
  - sizeHint, [3287](#)
  - updateThumbnailSize, [3287](#)
- Digikam::TableViewColumnConfiguration, [3288](#)

- Digikam::TableViewColumnConfigurationWidget, [3288](#)
- Digikam::TableViewColumnDescription, [3289](#)
- Digikam::TableViewColumnFactory, [3289](#)
- Digikam::TableViewColumnProfile, [3290](#)
  - loadSettings, [3290](#)
- Digikam::TableViewColumns::ColumnAudioVideoProperties, [3291](#)
  - compare, [3293](#)
  - data, [3293](#)
  - getColumnFlags, [3293](#)
  - getTitle, [3293](#)
  - setConfiguration, [3293](#)
- Digikam::TableViewColumns::ColumnDigikamProperties, [3295](#)
  - columnAffectedByChangeset, [3297](#)
  - compare, [3297](#)
  - data, [3297](#)
  - getColumnFlags, [3297](#)
  - getDescription, [3297](#)
  - getTitle, [3298](#)
- Digikam::TableViewColumns::ColumnFileConfigurationWidget, [3298](#)
  - getNewConfiguration, [3299](#)
- Digikam::TableViewColumns::ColumnFileProperties, [3300](#)
  - compare, [3302](#)
  - data, [3302](#)
  - getColumnFlags, [3302](#)
  - getConfigurationWidget, [3302](#)
  - getTitle, [3302](#)
  - setConfiguration, [3303](#)
- Digikam::TableViewColumns::ColumnGeoConfigurationWidget, [3303](#)
  - getNewConfiguration, [3304](#)
- Digikam::TableViewColumns::ColumnGeoProperties, [3305](#)
  - compare, [3307](#)
  - data, [3307](#)
  - getColumnFlags, [3307](#)
  - getConfigurationWidget, [3307](#)
  - getTitle, [3307](#)
  - setConfiguration, [3308](#)
- Digikam::TableViewColumns::ColumnItemProperties, [3309](#)
  - compare, [3311](#)
  - data, [3311](#)
  - getColumnFlags, [3311](#)
  - getTitle, [3311](#)
- Digikam::TableViewColumns::ColumnPhotoConfigurationWidget, [3312](#)
  - getNewConfiguration, [3313](#)
- Digikam::TableViewColumns::ColumnPhotoProperties, [3314](#)
  - compare, [3316](#)
  - data, [3316](#)
  - getColumnFlags, [3316](#)
  - getConfigurationWidget, [3316](#)
  - getTitle, [3316](#)
- setConfiguration, [3317](#)
- Digikam::TableViewColumns::ColumnThumbnail, [3318](#)
  - data, [3320](#)
  - getColumnFlags, [3320](#)
  - getTitle, [3320](#)
  - paint, [3320](#)
  - sizeHint, [3320](#)
  - updateThumbnailSize, [3320](#)
- Digikam::TableViewConfigurationDialog, [3321](#)
- Digikam::TableViewItemDelegate, [3322](#)
  - sizeHint, [3322](#)
- Digikam::TableViewModel, [3323](#)
  - addColumnAt, [3325](#)
  - flags, [3325](#)
  - indexFromImageId, [3325](#)
  - infoFromItem, [3325](#)
  - loadColumnProfile, [3325](#)
  - parent, [3325](#)
  - sort, [3325](#)
- Digikam::TableViewModel::Item, [3326](#)
- Digikam::TableViewSelectionModeSyncer, [3326](#)
  - TableViewSelectionModeSyncer, [3327](#)
- Digikam::TableViewShared, [3327](#)
- Digikam::TableViewTreeView, [3328](#)
  - dragDropHandler, [3329](#)
  - hasHiddenGroupedImages, [3329](#)
  - mapIndexForDragDrop, [3329](#)
  - pixmapForDrag, [3330](#)
- Digikam::TagChangeset, [3330](#)
  - Operation, [3330](#)
  - PropertiesChanged, [3330](#)
- Digikam::TagCheckView, [3331](#)
  - addCustomContextMenuActions, [3336](#)
  - checkedTagsChanged, [3336](#)
  - doLoadState, [3337](#)
  - doSaveState, [3337](#)
  - setCheckNewTags, [3337](#)
- Digikam::TagCompleter, [3338](#)
  - setContextParentTag, [3339](#)
  - setSupportingTagModel, [3339](#)
  - TagCompleter, [3338](#)
  - update, [3339](#)
- Digikam::TagData, [3339](#)
- Digikam::TagDragDropHandler, [3340](#)
  - accepts, [3341](#)
  - createMimeData, [3341](#)
  - dropEvent, [3341](#)
  - mimeTypes, [3341](#)
- Digikam::TagEditDlg, [3342](#)
  - createTAlbum, [3343](#)
- Digikam::TagFilterView, [3344](#)
  - ~TagFilterView, [3350](#)
  - addCustomContextMenuActions, [3350](#)
  - handleCustomContextMenuAction, [3350](#)
  - TagFilterView, [3349](#)
- Digikam::TagFolderView, [3351](#)
  - ~TagFolderView, [3355](#)
  - addCustomContextMenuActions, [3356](#)

- contextMenuEvent, [3356](#)
- contextMenuTitle, [3356](#)
- handleCustomContextMenuAction, [3356](#)
- setContextMenuItems, [3357](#)
- setShowDeleteFaceTagsAction, [3357](#)
- setShowFindDuplicateAction, [3357](#)
- TagFolderView, [3355](#)
- tagPropsEdit, [3358](#)
- Digikam::TaggingAction, [3358](#)
  - TaggingAction, [3359](#)
  - Type, [3358](#)
- Digikam::TaggingActionFactory, [3359](#)
  - MatchContainingFragment, [3360](#)
  - MatchStartingWithFragment, [3360](#)
  - NameMatchMode, [3360](#)
  - setConstraintInterface, [3360](#)
- Digikam::TaggingActionFactory::ConstraintInterface, [3361](#)
- Digikam::TagInfo, [3361](#)
- Digikam::TagList, [3362](#)
  - restoreSettings, [3363](#)
- Digikam::TagMngrListModel, [3363](#)
  - addItem, [3364](#)
  - data, [3364](#)
  - dropMimeData, [3364](#)
  - supportedDropActions, [3365](#)
- Digikam::TagMngrListView, [3365](#)
  - startDrag, [3366](#)
- Digikam::TagMngrTreeView, [3367](#)
  - contextMenuEvent, [3372](#)
  - setContextMenuItems, [3372](#)
- Digikam::TagModel, [3373](#)
  - albumData, [3378](#)
  - albumForId, [3378](#)
  - decorationRoleData, [3378](#)
  - fontRoleData, [3378](#)
  - TagModel, [3378](#)
- Digikam::TagModificationHelper, [3379](#)
  - ~TagModificationHelper, [3381](#)
  - bindMultipleTags, [3381](#)
  - bindTag, [3381](#)
  - boundMultipleTags, [3381](#)
  - boundTag, [3381](#)
  - slotFaceTagDelete, [3381](#), [3382](#)
  - slotMultipleFaceTagDel, [3382](#)
  - slotMultipleTagDel, [3382](#)
  - slotMultipleTagsToFaceTags, [3383](#)
  - slotTagDelete, [3383](#)
  - slotTagEdit, [3383](#), [3384](#)
  - slotTagNew, [3384](#)
  - slotTagToFaceTag, [3385](#)
  - TagModificationHelper, [3380](#)
- Digikam::TagProperties, [3385](#)
  - addProperty, [3386](#)
  - getOrCreate, [3386](#)
  - hasProperty, [3386](#)
  - TagProperties, [3386](#)
  - value, [3387](#)
- Digikam::TagPropertiesFilterModel, [3388](#)
  - isFiltering, [3391](#)
  - matches, [3391](#)
- Digikam::TagProperty, [3391](#)
- Digikam::TagPropertyName, [3392](#)
- Digikam::TagPropWidget, [3392](#)
- Digikam::TagRegion, [3393](#)
  - absoluteToRelative, [3394](#)
  - adjustToOrientation, [3394](#)
  - intersects, [3394](#)
  - mapToOriginalSize, [3395](#)
  - relativeToAbsolute, [3395](#)
  - reverseToOrientation, [3395](#)
  - TagRegion, [3394](#)
  - toRect, [3395](#)
  - toVariant, [3396](#)
  - toXml, [3396](#)
- Digikam::TagsActionMngr, [3396](#)
  - actionCollections, [3397](#)
  - registerLabelsActions, [3397](#)
  - registerTagsActionCollections, [3397](#)
  - updateTagShortcut, [3397](#)
- Digikam::TagsCache, [3398](#)
  - canBeWrittenToMetadata, [3400](#)
  - colorLabelForTag, [3400](#)
  - colorLabelFromTags, [3400](#)
  - colorLabelTags, [3400](#)
  - containsPublicTags, [3401](#)
  - createTag, [3401](#)
  - getOrCreateInternalTag, [3401](#)
  - getOrCreateTag, [3401](#)
  - getOrCreateTagWithProperty, [3401](#)
  - hasProperty, [3401](#)
  - hasTag, [3402](#)
  - IncludeLeadingSlash, [3400](#)
  - isInternalTag, [3402](#)
  - LeadingSlashPolicy, [3400](#)
  - NoLeadingSlash, [3400](#)
  - parentTag, [3402](#)
  - parentTags, [3402](#)
  - pickLabelForTag, [3402](#)
  - pickLabelFromTags, [3402](#)
  - pickLabelTags, [3403](#)
  - properties, [3403](#)
  - propertyValue, [3403](#)
  - publicTags, [3403](#)
  - shortenedTagPaths, [3403](#)
  - tagAdded, [3403](#)
  - tagForColorLabel, [3404](#)
  - tagForName, [3404](#)
  - tagForPath, [3404](#)
  - tagForPickLabel, [3404](#)
  - tagName, [3404](#)
  - tagPath, [3404](#)
  - tagsContaining, [3405](#)
  - tagsForName, [3405](#)
  - tagsWithProperty, [3405](#)
  - tagsWithPropertyCached, [3405](#)

- Digikam::TagsDBJobInfo, [3406](#)
- Digikam::TagsDBJobsThread, [3407](#)
  - tagsListing, [3408](#)
- Digikam::TagsEdit, [3409](#)
- Digikam::TagShortInfo, [3410](#)
- Digikam::TagsJob, [3410](#)
- Digikam::TagsLineEditOverlay, [3412](#)
  - createWidget, [3414](#)
  - hide, [3414](#)
  - setActive, [3415](#)
  - slotEntered, [3415](#)
  - visualChange, [3415](#)
- Digikam::TagsManager, [3416](#)
  - doLoadState, [3417](#)
  - doSaveState, [3417](#)
- Digikam::TagsManagerFilterModel, [3419](#)
  - matches, [3422](#)
- Digikam::TagsPopupMenu, [3422](#)
  - DISPLAY, [3423](#)
  - Mode, [3423](#)
- Digikam::TagTreeView, [3424](#)
  - filteredModel, [3428](#)
- Digikam::TagTreeViewSelectComboBox, [3429](#)
- Digikam::TagViewSideBarWidget, [3432](#)
  - applySettings, [3434](#)
  - changeAlbumFromHistory, [3434](#)
  - doLoadState, [3434](#)
  - doSaveState, [3434](#)
  - getCaption, [3434](#)
  - getIcon, [3434](#)
  - setActive, [3435](#)
- Digikam::TAlbum, [3436](#)
  - databaseUrl, [3438](#)
  - tagPath, [3438](#)
- Digikam::Template, [3439](#)
  - isEmpty, [3440](#)
  - isNull, [3440](#)
  - m\_authors, [3440](#)
  - m\_authorsPosition, [3440](#)
  - m\_contactInfo, [3440](#)
  - m\_copyright, [3440](#)
  - m\_credit, [3441](#)
  - m\_instructions, [3441](#)
  - m\_locationInfo, [3441](#)
  - m\_rightUsageTerms, [3441](#)
  - m\_source, [3441](#)
  - m\_subjects, [3441](#)
  - m\_templateTitle, [3441](#)
  - merge, [3440](#)
  - operator==, [3440](#)
- Digikam::TemplateList, [3442](#)
- Digikam::TemplateListItem, [3443](#)
- Digikam::TemplateManager, [3444](#)
- Digikam::TemplatePanel, [3445](#)
- Digikam::TemplateSelector, [3446](#)
- Digikam::TemplateViewer, [3448](#)
- Digikam::TextFilter, [3450](#)
- Digikam::TextureContainer, [3451](#)
- Digikam::TextureFilter, [3452](#)
  - filterAction, [3455](#)
  - filterIdentifier, [3455](#)
  - readParameters, [3455](#)
- Digikam::TextureSettings, [3455](#)
- Digikam::ThemeManager, [3456](#)
- Digikam::ThemeManager::Private, [3457](#)
- Digikam::ThreadManager, [3458](#)
- Digikam::ThumbBarDock, [3459](#)
  - getToggleAction, [3460](#)
  - reInitialize, [3460](#)
  - shouldBeVisible, [3460](#)
- Digikam::ThumbnailAligningDelegate, [3461](#)
- Digikam::ThumbnailCreator, [3461](#)
  - deleteThumbnailsFromDisk, [3462](#)
  - errorString, [3462](#)
  - fileThumbnailInfo, [3462](#)
  - identifierForDetail, [3463](#)
  - load, [3463](#)
  - loadDetail, [3463](#)
  - pregenerate, [3463](#)
  - setExifRotate, [3463](#)
  - setLoadingProperties, [3463](#)
  - setOnlyLargeThumbnails, [3464](#)
  - setRemoveAlphaChannel, [3464](#)
  - setThumbnailInfoProvider, [3464](#)
  - setThumbnailSize, [3464](#)
  - store, [3464](#)
  - storedSize, [3464](#)
  - ThumbnailCreator, [3462](#)
  - thumbnailSize, [3465](#)
- Digikam::ThumbnailCreator::Private, [3465](#)
- Digikam::ThumbnailIdentifier, [3466](#)
  - filePath, [3466](#)
  - id, [3466](#)
- Digikam::ThumbnailImage, [3467](#)
- Digikam::ThumbnailImageCatcher, [3467](#)
  - cancel, [3468](#)
  - enqueue, [3468](#)
  - setActive, [3469](#)
  - ThumbnailImageCatcher, [3468](#)
- Digikam::ThumbnailImageCatcher::Private, [3469](#)
- Digikam::ThumbnailImageCatcher::Private::CatcherResult, [3469](#)
- Digikam::ThumbnailInfo, [3470](#)
  - customIdentifier, [3471](#)
  - fileName, [3471](#)
  - isAccessible, [3471](#)
  - contentType, [3471](#)
  - modificationDate, [3471](#)
  - orientationHint, [3471](#)
  - uniqueHash, [3472](#)
- Digikam::ThumbnailInfoProvider, [3472](#)
- Digikam::ThumbnailLoadingTask, [3473](#)
  - execute, [3475](#)
  - postProcess, [3475](#)
- Digikam::ThumbnailLoadThread, [3476](#)
  - defaultThread, [3480](#)

- deleteThumbnail, 3480
- find, 3481
- findBuffered, 3482
- findGroup, 3482
- initializeNoThumbnailStorage, 3482
- initializeThumbnailDatabase, 3482
- lastDescriptions, 3482
- load, 3482
- maximumThumbnailSize, 3483
- pixmapToThumbnailSize, 3483
- pregenerateGroup, 3483
- preload, 3483
- setDisplayingWidget, 3483
- setHighlightPixmap, 3483
- setPixmapRequested, 3484
- setSendSurrogatePixmap, 3484
- setThumbnailSize, 3484
- storeDetailThumbnail, 3484
- thumbnailLoaded, 3484
- thumbnailToPixmapSize, 3485
- Digikam::ThumbnailLoadThread::Private, 3485
- Digikam::ThumbnailLoadThreadStaticPriv, 3485
- Digikam::ThumbnailResult, 3486
- Digikam::ThumbnailSize, 3486
  - Size, 3486
  - Small, 3487
- Digikam::ThumbsDb, 3487
  - findAll, 3487
  - findByFilePath, 3487
  - integrityCheck, 3488
  - removeByFilePath, 3488
  - removeByUniqueHash, 3488
  - vacuum, 3488
- Digikam::ThumbsDbAccess, 3488
  - setLastError, 3489
  - ThumbsDbAccess, 3489
- Digikam::ThumbsDbBackend, 3490
  - initSchema, 3493
- Digikam::ThumbsDbInfo, 3493
- Digikam::ThumbsDbInfoProvider, 3493
  - thumbnailInfo, 3494
- Digikam::ThumbsDbSchemaUpdater, 3494
- Digikam::ThumbsGenerator, 3495
  - setUseMultiCoreCPU, 3498
  - ThumbsGenerator, 3497
- Digikam::ThumbsTask, 3498
- Digikam::TileGrouper, 3500
  - updateClusters, 3500
- Digikam::TileIndex, 3500
- Digikam::TimeAdjustContainer, 3501
  - atLeastOneUpdateToProcess, 3502
  - enableExifTool, 3502
- Digikam::TimeAdjustSettings, 3503
  - detAdjustmentByClockPhotoUrl, 3503
- Digikam::TimelineSideBarWidget, 3504
  - applySettings, 3505
  - changeAlbumFromHistory, 3505
  - doLoadState, 3506
  - doSaveState, 3506
  - getCaption, 3506
  - getIcon, 3506
  - setActive, 3506
- Digikam::TimeLineWidget, 3507
  - FuzzySelection, 3509
  - LinScale, 3508
  - LogScale, 3508
  - ScaleMode, 3508
  - Selected, 3509
  - selectedDateRange, 3509
  - SelectionMode, 3509
  - Unselected, 3509
- Digikam::TimeZoneComboBox, 3509
- Digikam::Token, 3510
  - action, 3511
  - description, 3511
  - id, 3511
  - signalTokenTriggered, 3511
- Digikam::TonalityContainer, 3512
- Digikam::TonalityFilter, 3513
  - filterAction, 3516
  - filterIdentifier, 3516
  - readParameters, 3516
- Digikam::ToolListViewGroup, 3516
- Digikam::ToolListViewItem, 3517
- Digikam::ToolSettingsView, 3518
- Digikam::ToolsListView, 3519
- Digikam::ToolsView, 3520
- Digikam::TooltipCreator, 3521
- Digikam::TooltipDialog, 3521
- Digikam::TooltipsPage, 3522
- Digikam::TrackCorrelator, 3523
  - Digikam::TrackCorrelator::Correlation, 3524
  - Digikam::TrackCorrelator::CorrelationOptions, 3524
- Digikam::TrackCorrelatorThread, 3525
- Digikam::TrackListModel, 3526
  - headerData, 3527
  - index, 3527
- Digikam::TrackManager, 3527
  - clear, 3528
  - Digikam::TrackManager::Track, 3528
  - Digikam::TrackManager::TrackPoint, 3529
- Digikam::TrackReader, 3529
  - Digikam::TrackReader::TrackReadResult, 3530
- Digikam::TrainerWorker, 3531
  - aboutToDeactivate, 3533
  - process, 3533
- Digikam::TrainingDataProvider, 3534
  - images, 3534
  - newImages, 3534
- Digikam::TransactionItem, 3536
  - setStatus, 3537
- Digikam::TransactionItemView, 3538
- Digikam::TransitionMngr, 3539
  - Digikam::TransitionMngr::Private, 3539
- Digikam::TransitionPreview, 3541
- Digikam::TrashView, 3542

- getThumbnailSize, [3543](#)
- lastSelectedItemUrl, [3543](#)
- model, [3543](#)
- setThumbnailSize, [3543](#)
- statusBarText, [3543](#)
- Digikam::TreeBranch, [3544](#)
- Digikam::TreeProxyModel, [3544](#)
- Digikam::TreeViewComboBox, [3545](#)
  - installView, [3547](#)
  - sendViewportEventToView, [3547](#)
  - TreeViewComboBox, [3546](#)
  - view, [3547](#)
- Digikam::TreeViewLineEditComboBox, [3548](#)
  - installLineEdit, [3550](#)
  - installView, [3550](#)
  - setLineEditText, [3550](#)
  - TreeViewLineEditComboBox, [3550](#)
- Digikam::TrimmedModifier, [3551](#)
  - parseOperation, [3552](#)
- Digikam::TwoProgressItemsContainer, [3553](#)
- Digikam::UMSCamera, [3555](#)
  - cameraAbout, [3557](#)
  - cameraDriverType, [3557](#)
  - cameraManual, [3557](#)
  - cameraMDSID, [3557](#)
  - cameraSummary, [3558](#)
  - cancel, [3558](#)
  - capture, [3558](#)
  - deleteItem, [3558](#)
  - doConnect, [3558](#)
  - downloadItem, [3558](#)
  - getFolders, [3558](#)
  - getFreeSpace, [3559](#)
  - getItemInfo, [3559](#)
  - getItemInfoList, [3559](#)
  - getMetadata, [3559](#)
  - getPreview, [3559](#)
  - getThumbnail, [3559](#)
  - setLockItem, [3560](#)
  - uploadItem, [3560](#)
- Digikam::UndoAction, [3561](#)
- Digikam::UndoActionIrreversible, [3562](#)
- Digikam::UndoActionReversible, [3563](#)
- Digikam::UndoCache, [3564](#)
  - clear, [3564](#)
  - clearFrom, [3564](#)
  - getData, [3564](#)
  - putData, [3564](#)
- Digikam::UndoManager, [3565](#)
- Digikam::UndoMetadataContainer, [3565](#)
  - fromImage, [3565](#)
  - toImage, [3565](#)
- Digikam::UndoState, [3566](#)
- Digikam::UniqueModifier, [3567](#)
  - parseOperation, [3568](#)
  - reset, [3569](#)
- Digikam::UnsharpMaskFilter, [3570](#)
  - filterAction, [3573](#)
  - filterIdentifier, [3573](#)
  - readParameters, [3573](#)
- Digikam::VersionFileInfo, [3573](#)
- Digikam::VersionFileOperation, [3573](#)
  - allFilePaths, [3574](#)
  - MoveToIntermediate, [3574](#)
  - NewFile, [3574](#)
  - Replace, [3574](#)
  - SaveAndDelete, [3574](#)
  - StoreIntermediates, [3574](#)
  - Task, [3574](#)
  - VersionFileOperation, [3574](#)
- Digikam::VersioningPromptUserSaveDialog, [3575](#)
- Digikam::VersionItemFilterSettings, [3575](#)
  - matches, [3576](#)
  - setExceptionList, [3576](#)
- Digikam::VersionManager, [3577](#)
- Digikam::VersionManagerSettings, [3578](#)
- Digikam::VersionNamingScheme, [3579](#)
  - baseName, [3580](#)
  - directory, [3580](#)
  - incrementedCounter, [3580](#)
  - initialCounter, [3580](#)
  - intermediateFileName, [3581](#)
  - versionFileName, [3581](#)
  - VersionNamingScheme, [3580](#)
- Digikam::VersionsDelegate, [3582](#)
  - asDelegate, [3584](#)
- Digikam::VersionsTreeView, [3585](#)
  - dragDropHandler, [3586](#)
  - mapIndexForDragDrop, [3586](#)
  - pixmapForDrag, [3587](#)
- Digikam::VersionsWidget, [3587](#)
- Digikam::VideoFrame, [3588](#)
- Digikam::VideoInfoContainer, [3588](#)
- Digikam::VideoMetadataContainer, [3589](#)
- Digikam::VideoStripFilter, [3589](#)
- Digikam::VideoThumbDecoder, [3589](#)
- Digikam::VideoThumbDecoder::Private, [3590](#)
- Digikam::VideoThumbnailer, [3590](#)
- Digikam::VideoThumbWriter, [3591](#)
- Digikam::VidPlayerDlg, [3591](#)
- Digikam::VidSlideSettings, [3591](#)
  - AVI, [3595](#)
  - BLUERAY, [3596](#)
  - conflictRule, [3597](#)
  - CVD1, [3595](#)
  - CVD2, [3595](#)
  - DVD1, [3596](#)
  - DVD2, [3596](#)
  - DVGA, [3596](#)
  - EDTV1, [3596](#)
  - EDTV2, [3596](#)
  - EGA, [3596](#)
  - FLASH, [3594](#)
  - HDPLUS, [3596](#)
  - HDTV, [3596](#)
  - HSXGA, [3596](#)

HUXGA, [3596](#)  
 HVGA, [3595](#)  
 HXGA, [3596](#)  
 MJPEG, [3594](#)  
 MKV, [3595](#)  
 MP4, [3595](#)  
 MPEG2, [3594](#)  
 MPEG4, [3594](#)  
 MPG, [3595](#)  
 NTSC, [3595](#)  
 outputDir, [3597](#)  
 PAL, [3595](#)  
 QSXGA, [3596](#)  
 QSXGAPLUS, [3596](#)  
 QUXGA, [3596](#)  
 QVGA, [3595](#)  
 QXGA, [3596](#)  
 readSettings, [3597](#)  
 SDTV1, [3595](#)  
 SDTV2, [3596](#)  
 SDTV3, [3596](#)  
 Selection, [3594](#)  
 SVCD1, [3595](#)  
 SVCD2, [3596](#)  
 SVGA, [3596](#)  
 SXGA, [3596](#)  
 SXGAPLUS, [3596](#)  
 THEORA, [3594](#)  
 TXGA, [3596](#)  
 UHD4K, [3596](#)  
 UHD5K, [3596](#)  
 UHD6K, [3596](#)  
 UHD8K, [3596](#)  
 UW10K, [3596](#)  
 UW16K, [3597](#)  
 UWFHD, [3596](#)  
 UXGA, [3596](#)  
 VBR04, [3594](#)  
 VBR05, [3594](#)  
 VBR10, [3594](#)  
 VBR12, [3594](#)  
 VBR15, [3594](#)  
 VBR20, [3594](#)  
 VBR25, [3594](#)  
 VBR30, [3594](#)  
 VBR40, [3594](#)  
 VBR45, [3594](#)  
 VBR50, [3594](#)  
 VBR60, [3594](#)  
 VBR80, [3594](#)  
 VCD1, [3595](#)  
 VCD2, [3595](#)  
 VGA, [3596](#)  
 VidBitRate, [3594](#)  
 VidCodec, [3594](#)  
 videoTypeNames, [3597](#)  
 VidFormat, [3594](#)  
 VidPlayer, [3595](#)  
 VidStd, [3595](#)  
 VidType, [3595](#)  
 WEBMVP8, [3594](#)  
 WHSXGA, [3596](#)  
 WHUXGA, [3596](#)  
 WHXGA, [3596](#)  
 WMV7, [3594](#)  
 WMV8, [3594](#)  
 WMV9, [3594](#)  
 WQHD, [3596](#)  
 WQSXGA, [3596](#)  
 WQUXGA, [3596](#)  
 WQXGA, [3596](#)  
 WQXGAPLUS, [3596](#)  
 WSXGA, [3596](#)  
 WSXGAPLUS, [3596](#)  
 WUXGA, [3596](#)  
 WVGA, [3596](#)  
 WXGA1, [3596](#)  
 WXGA2, [3596](#)  
 X264, [3594](#)  
 XVGA, [3596](#)  
 Digikam::VidSlideTask, [3598](#)  
 Digikam::VidSlideThread, [3600](#)  
     prepareFrames, [3601](#)  
 Digikam::VisibilityController, [3602](#)  
     addObject, [3603](#)  
     addWidget, [3603](#)  
     isVisible, [3603](#)  
     setContainerWidget, [3603](#)  
 Digikam::VisibilityObject, [3604](#)  
 Digikam::WBContainer, [3604](#)  
     black, [3605](#)  
 Digikam::WBFilter, [3606](#)  
     autoWBAdjustementFromColor, [3609](#)  
     filterAction, [3609](#)  
     filterIdentifier, [3609](#)  
     filterImage, [3609](#)  
     readParameters, [3610](#)  
 Digikam::WBSettings, [3610](#)  
 Digikam::WebBrowserDlg, [3611](#)  
 Digikam::WebWidget, [3612](#)  
 Digikam::WelcomePage, [3613](#)  
 Digikam::WelcomePageView, [3614](#)  
 Digikam::WelcomePageViewPage, [3615](#)  
 Digikam::WorkerObject, [3616](#)  
     aboutToDeactivate, [3618](#)  
     aboutToQuitLoop, [3618](#)  
     connectAndSchedule, [3618](#)  
     deactivate, [3618](#)  
     DeactivatingMode, [3617](#)  
     FlushSignals, [3617](#)  
     KeepSignals, [3617](#)  
     PhaseOut, [3617](#)  
     schedule, [3618](#)  
     setPriority, [3618](#)  
     shutDown, [3619](#)  
     WorkerObject, [3617](#)



- Digikam::Workflow, [3619](#)
- Digikam::WorkflowDlg, [3620](#)
- Digikam::WorkflowItem, [3621](#)
- Digikam::WorkflowList, [3622](#)
- Digikam::WorkflowManager, [3623](#)
  - load, [3624](#)
  - save, [3624](#)
- Digikam::WorkingWidget, [3624](#)
- Digikam::WSAlbum, [3625](#)
  - setBaseAlbum, [3625](#)
- Digikam::WSComboBoxIntermediate, [3626](#)
  - setIntermediate, [3626](#)
  - WSComboBoxIntermediate, [3626](#)
- Digikam::WSLoginDialog, [3627](#)
- Digikam::WSNewAlbumDialog, [3628](#)
- Digikam::WSSelectUserDlg, [3629](#)
- Digikam::WSSettings, [3630](#)
- Digikam::WSSettingsWidget, [3632](#)
- Digikam::WSToolDialog, [3634](#)
- Digikam::WSToolUtils, [3635](#)
  - randomString, [3635](#)
- Digikam::XbelReader, [3636](#)
- Digikam::XbelWriter, [3636](#)
- Digikam::XmpWidget, [3637](#)
  - getMetadataTitle, [3639](#)
  - getTagDescription, [3639](#)
  - getTagTitle, [3639](#)
  - loadFromURL, [3639](#)
- dimensions
  - Digikam::ItemInfo, [2298](#)
- dimensionsHint
  - Digikam::DatabaseLoadSaveFileInfoProvider, [780](#)
  - Digikam::LoadSaveFileInfoProvider, [2562](#)
- DImg
  - Digikam::DImg, [1076](#), [1077](#)
- DImgBuiltinFilter
  - Digikam::DImgBuiltinFilter, [1090](#), [1091](#)
- DImgChildItem
  - Digikam::DImgChildItem, [1095](#)
- DImgLoaderPrms
  - Digikam, [124](#)
- DImgThreadedAnalyser
  - Digikam::DImgThreadedAnalyser, [1114](#)
- DImgThreadedFilter
  - Digikam::DImgThreadedFilter, [1116](#), [1117](#)
- DIRECT\_CHILDREN
  - Digikam::StateSavingObject, [3261](#)
- DirectMatch
  - Digikam::AlbumFilterModel, [269](#)
- directory
  - Digikam::DefaultVersionNamingScheme, [1000](#)
  - Digikam::VersionNamingScheme, [3580](#)
- dirtyMainToolBar
  - Digikam::DXmlGuiWindow::Private, [1449](#)
- disableHistogramGuide
  - Digikam::HistogramPainter, [1887](#)
- disableIfEmpty
  - Digikam::IccProfilesMenuAction, [1930](#)
- discoverFormat
  - Digikam::FileSaveOptionsBox, [1687](#)
- DISPLAY
  - Digikam::TagsPopupMenu, [3423](#)
- Display
  - Digikam::IccProfile, [1923](#)
- displayName
  - Digikam::BasicDImgFilterGenerator< T >, [462](#)
  - Digikam::DImgBuiltinFilter, [1091](#)
  - Digikam::DImgFilterGenerator, [1098](#)
  - Digikam::DImgFilterManager, [1100](#)
- DisplayFlag
  - Digikam::DFontProperties, [1027](#)
- displayProfiles
  - Digikam::IccSettings, [1938](#)
- displaySoftProofingTransform
  - Digikam::IccManager, [1917](#)
- DItemsListIsLessThanHandler
  - Digikam, [124](#)
- DMultiTabBarTab
  - Digikam::DMultiTabBarTab, [1209](#)
- DNG\_SDK\_INTERNAL\_ERROR
  - Digikam::DNGWriter, [1212](#)
- DNGVersion
  - Digikam::DRawInfo, [1391](#)
- DNNDetectorSSD
  - Digikam, [125](#)
- DNNDetectorYOLOv3
  - Digikam, [125](#)
- DNNDetectorYuNet
  - Digikam, [125](#)
- DNotificationPopup
  - Digikam::DNotificationPopup, [1249](#)
- DNotificationWidget
  - Digikam::DNotificationWidget, [1260](#)
- DNotificationWrapper
  - Digikam, [128](#)
- doConnect
  - Digikam::GPCamera, [1787](#)
  - Digikam::UMSCamera, [3558](#)
- DocumentedHistory
  - Digikam::FilterAction, [1706](#)
- doLoadState
  - Digikam::AbstractAlbumTreeView, [152](#)
  - Digikam::AbstractCheckableAlbumTreeView, [175](#)
  - Digikam::AlbumFolderViewSideBarWidget, [276](#)
  - Digikam::DateFolderView, [811](#)
  - Digikam::DateFolderViewSideBarWidget, [814](#)
  - Digikam::FaceScanWidget, [1627](#)
  - Digikam::FilterSideBarWidget, [1717](#)
  - Digikam::FuzzySearchSideBarWidget, [1755](#)
  - Digikam::FuzzySearchView, [1759](#)
  - Digikam::GPSSearchSideBarWidget, [1830](#)
  - Digikam::GPSSearchView, [1834](#)
  - Digikam::ImportItemPropertiesSideBarImport, [2075](#)
  - Digikam::ItemPropertiesSideBar, [2371](#)
  - Digikam::ItemPropertiesSideBarDB, [2376](#)

- Digikam::LabelsSideBarWidget, [2497](#)
- Digikam::LabelsTreeView, [2500](#)
- Digikam::MapWidgetView, [2629](#)
- Digikam::PeopleSideBarWidget, [2832](#)
- Digikam::SearchSideBarWidget, [3121](#)
- Digikam::SearchTextBar, [3125](#)
- Digikam::Sidebar, [3218](#)
- Digikam::StateSavingObject, [3261](#)
- Digikam::TableView, [3284](#)
- Digikam::TagCheckView, [3337](#)
- Digikam::TagsManager, [3417](#)
- Digikam::TagViewSideBarWidget, [3434](#)
- Digikam::TimelineSideBarWidget, [3506](#)
- ShowFoto::ShowfotoFolderViewSideBar, [3688](#)
- ShowFoto::ShowfotoStackViewSideBar, [3743](#)
- DOnlineTranslator
  - Digikam::DOnlineTranslator, [1272](#)
- DOnlineTts
  - Digikam::DOnlineTts, [1287](#)
- DontStretchPixels
  - Digikam::DRawDecoderSettings, [1378](#)
- doSaveState
  - Digikam::AbstractAlbumTreeView, [152](#)
  - Digikam::AbstractCheckableAlbumTreeView, [175](#)
  - Digikam::AlbumFolderViewSideBarWidget, [276](#)
  - Digikam::DateFolderView, [811](#)
  - Digikam::DateFolderViewSideBarWidget, [814](#)
  - Digikam::FaceScanWidget, [1627](#)
  - Digikam::FilterSideBarWidget, [1717](#)
  - Digikam::FuzzySearchSideBarWidget, [1755](#)
  - Digikam::FuzzySearchView, [1759](#)
  - Digikam::GPSSearchSideBarWidget, [1830](#)
  - Digikam::GPSSearchView, [1834](#)
  - Digikam::ImportItemPropertiesSideBarImport, [2076](#)
  - Digikam::ItemPropertiesSideBar, [2371](#)
  - Digikam::ItemPropertiesSideBarDB, [2376](#)
  - Digikam::LabelsSideBarWidget, [2497](#)
  - Digikam::LabelsTreeView, [2500](#)
  - Digikam::MapWidgetView, [2629](#)
  - Digikam::PeopleSideBarWidget, [2832](#)
  - Digikam::SearchSideBarWidget, [3121](#)
  - Digikam::SearchTextBar, [3125](#)
  - Digikam::Sidebar, [3218](#)
  - Digikam::StateSavingObject, [3261](#)
  - Digikam::TableView, [3284](#)
  - Digikam::TagCheckView, [3337](#)
  - Digikam::TagsManager, [3417](#)
  - Digikam::TagViewSideBarWidget, [3434](#)
  - Digikam::TimelineSideBarWidget, [3506](#)
  - ShowFoto::ShowfotoFolderViewSideBar, [3688](#)
  - ShowFoto::ShowfotoStackViewSideBar, [3743](#)
- downloaded
  - Digikam::CamItemInfo, [557](#)
- DownloadedNo
  - Digikam::CamItemInfo, [557](#)
- DownloadedYes
  - Digikam::CamItemInfo, [557](#)
- DownloadFailed
  - Digikam::CamItemInfo, [557](#)
- downloadItem
  - Digikam::GPCamera, [1788](#)
  - Digikam::UMSCamera, [3558](#)
- DownloadStarted
  - Digikam::CamItemInfo, [557](#)
- DownloadStatus
  - Digikam::CamItemInfo, [556](#)
- DownloadUnknown
  - Digikam::CamItemInfo, [557](#)
- DPlainTextEdit
  - Digikam::DPlainTextEdit, [1295](#)
- DPlugin
  - Digikam::DPlugin, [1299](#)
- DPluginBqm
  - Digikam::DPluginBqm, [1308](#)
- DPluginConfView
  - Digikam::DPluginConfView, [1311](#)
- DPluginDImg
  - Digikam::DPluginDImg, [1324](#)
- DPluginEditor
  - Digikam::DPluginEditor, [1329](#)
- DPluginGeneric
  - Digikam::DPluginGeneric, [1333](#)
- DPluginRawImport
  - Digikam::DPluginRawImport, [1342](#)
- DPointSelect
  - Digikam::DPointSelect, [1345](#)
- DPopupFrame
  - Digikam::DPopupFrame, [1350](#)
- dragDropFlags
  - Digikam::DragDropModelImplementation, [1362](#)
- dragDropFlagsV2
  - Digikam::DragDropModelImplementation, [1362](#)
- dragDropHandler
  - Digikam::AbstractAlbumModel, [144](#)
  - Digikam::DragDropViewImplementation, [1365](#)
  - Digikam::ImportCategorizedView, [2016](#)
  - Digikam::ItemCategorizedView, [2187](#)
  - Digikam::TableViewTreeView, [3329](#)
  - Digikam::VersionsTreeView, [3586](#)
  - ShowFoto::ShowfotoCategorizedView, [3654](#)
- DragDropModelImplementation
  - Digikam::DragDropModelImplementation, [1362](#)
- dragEnterEvent
  - Digikam::DragDropViewImplementation, [1365](#)
  - Digikam::MapWidget, [2623](#)
- drawArrow
  - Digikam::DSelector, [1401](#)
- drawCategory
  - Digikam::DCategoryDrawer, [893](#)
  - Digikam::ImportCategoryDrawer, [2021](#)
  - Digikam::ItemCategoryDrawer, [2191](#)
- drawContents
  - Digikam::DColorValueSelector, [905](#)
  - Digikam::DHueSaturationSelector, [1040](#)
  - Digikam::DPointSelect, [1346](#)

- Digikam::DSelector, [1401](#)
- DRawDecoder
  - Digikam::DRawDecoder, [1368](#)
- DRawDecoderSettings
  - Digikam::DRawDecoderSettings, [1377](#)
- DRawDecoderWidget
  - Digikam::DRawDecoderWidget, [1384](#)
- DRawDecoding
  - Digikam::DRawDecoding, [1385](#)
- drawDraggedItems
  - Digikam::DCategorizedView::Private, [888](#)
- DRawInfo
  - Digikam::DRawInfo, [1389](#)
- drawMarker
  - Digikam::DPointSelect, [1346](#)
- drawNewCategory
  - Digikam::DCategorizedView::Private, [888](#)
- drawPalette
  - Digikam::DColorValueSelector, [905](#)
  - Digikam::DHueSaturationSelector, [1041](#)
- drawThumbnail
  - Digikam::ItemViewDelegate, [2452](#)
- dropEvent
  - Digikam::AbstractItemDragDropHandler, [188](#)
  - Digikam::AlbumDragDropHandler, [265](#)
  - Digikam::AlbumModelDragDropHandler, [314](#)
  - Digikam::ImportDragDropHandler, [2047](#)
  - Digikam::ItemDragDropHandler, [2227](#)
  - Digikam::MapDragDropHandler, [2614](#)
  - Digikam::TagDragDropHandler, [3341](#)
  - ShowFoto::ShowfotoDragDropHandler, [3671](#)
- dropInfo
  - Digikam::ItemInfoCache, [2310](#)
- dropMimeData
  - Digikam::TagMngrListModel, [3364](#)
- dropUnresolvedEntries
  - Digikam::ItemHistoryGraph, [2276](#)
- DTextEdit
  - Digikam::DTextEdit, [1412](#)
- DuplicatesFinder
  - Digikam::DuplicatesFinder, [1431](#)
- DVD1
  - Digikam::VidSlideSettings, [3596](#)
- DVD2
  - Digikam::VidSlideSettings, [3596](#)
- DVGA
  - Digikam::VidSlideSettings, [3596](#)
- DWItemDelegate
  - Digikam::DWItemDelegate, [1435](#)
- DWItemDelegatePool
  - Digikam::DWItemDelegatePool, [1438](#)
- DynamicThread
  - Digikam::DynamicThread, [1453](#)
- eccentricity
  - Digikam::Ellipsoid, [1494](#)
- edgeDifference
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1841](#)
- EdgesToLeaf
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1841](#)
- EditableSearchTreeView
  - Digikam::EditableSearchTreeView, [1463](#)
- editGroup
  - Digikam::FileActionMngrDatabaseWorker, [1673](#)
- editKeyboardShortcuts
  - Digikam::DXmlGuiWindow, [1447](#)
- Editor
  - Digikam::DPluginAction, [1304](#)
- EditorColors
  - Digikam::DPluginAction, [1304](#)
- EditorDecorate
  - Digikam::DPluginAction, [1304](#)
- EditorEnhance
  - Digikam::DPluginAction, [1304](#)
- EditorFile
  - Digikam::DPluginAction, [1304](#)
- EditorFilters
  - Digikam::DPluginAction, [1304](#)
- EditorTransform
  - Digikam::DPluginAction, [1304](#)
- editRegion
  - Digikam::FacePipeline, [1567](#)
- editSearch
  - Digikam::NormalSearchTreeView, [2780](#)
- editTag
  - Digikam::FacePipeline, [1567](#)
- EDTV1
  - Digikam::VidSlideSettings, [3596](#)
- EDTV2
  - Digikam::VidSlideSettings, [3596](#)
- effectiveInputProfile
  - Digikam::lccTransform, [1943](#)
- EffectType
  - Digikam::EffectMngr, [1490](#)
- EGA
  - Digikam::VidSlideSettings, [3596](#)
- elapsedTime
  - Digikam::ProcessLauncher, [2867](#)
- elementsInfo
  - Digikam::DCategorizedView::Private, [890](#)
- Ellipsoid
  - Digikam::Ellipsoid, [1492](#)
- ellipsoid
  - Digikam::GeodeticCalculator, [1765](#)
- embeddedProfile
  - Digikam::lccTransform, [1943](#)
- Emotion
  - Digikam::DOnlineTts, [1285](#)
- emotion
  - Digikam::DOnlineTts, [1287](#)
- emotionCode
  - Digikam::DOnlineTts, [1287](#)
- emptyDTrashItems
  - Digikam::IOJobsThread, [2161](#)
- enableBlackPoint

- Digikam::DRawDecoderSettings, [1378](#)
- enableColumn
  - Digikam::DFontProperties, [1029](#)
- enabled
  - Digikam::DConfigDlgWdgItem, [956](#)
- enableExifTool
  - Digikam::TimeAdjustContainer, [3502](#)
- enableFractionMagic
  - Digikam::CustomStepsIntSpinBox, [763](#)
- enableHistogramGuideByColor
  - Digikam::HistogramPainter, [1887](#)
- enableKineticScrollFor
  - ShowFoto::ShowfotoKineticScroller, [3712](#)
- enableWhitePoint
  - Digikam::DRawDecoderSettings, [1379](#)
- encodeFrames
  - Digikam::FFmpegLauncher, [1662](#)
- endDate
  - Digikam::CoreDbUrl, [740](#)
- engineName
  - Digikam::DOnlineTranslator, [1273](#)
- EnhanceTool
  - Digikam::BatchTool, [465](#)
- enqueue
  - Digikam::FacePipelineBase, [1575](#)
  - Digikam::ThumbnailImageCatcher, [3468](#)
- ensureHasGroupedImages
  - Digikam::ItemModel, [2349](#)
- ensureHasItemInfo
  - Digikam::ItemModel, [2349](#)
- ensureIsPerson
  - Digikam::FaceTags, [1636](#)
- entries
  - Digikam::DImageHistory, [1068](#)
- entryName
  - Digikam::StateSavingObject, [3262](#)
- equalsIgnoreReducedVersion
  - Digikam::LoadingDescription, [2554](#)
- equalsOrBetterThan
  - Digikam::LoadingDescription, [2554](#)
- error
  - Digikam::DBJobsThread, [863](#)
  - Digikam::DOnlineTranslator, [1273](#)
  - Digikam::DOnlineTts, [1288](#)
  - Digikam::ItemListerJobReceiver, [2329](#)
  - Digikam::ItemListerValueListReceiver, [2333](#)
- errorDescription
  - Digikam::BatchTool, [467](#)
- ErrorMessage
  - Digikam::DConfigDlgTitle, [931](#)
- errorMessage
  - Digikam::LookupAltitudeGeonames, [2587](#)
- errorsList
  - Digikam::DBJobsThread, [863](#)
  - Digikam::IOJobsThread, [2161](#)
- errorString
  - Digikam::DOnlineTranslator, [1273](#)
  - Digikam::DOnlineTts, [1288](#)
- Digikam::ThumbnailCreator, [3462](#)
- escapeToken
  - Digikam::Rule, [2996](#)
- event
  - Digikam::DDateTable, [981](#)
- eventFilter
  - Digikam::BackendMarble, [450](#)
  - Digikam::RGWidget, [2990](#)
- excludeChildrenCount
  - Digikam::AbstractCountingAlbumModel, [181](#)
- ExcludeFadingOut
  - Digikam::ItemVisibilityController, [2473](#)
- ExcludeFolder
  - Digikam::Haarface, [1876](#)
- exclusiveMerge
  - Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >, [2707](#)
- exec
  - Digikam::BdEngineBackend, [484](#)
  - Digikam::ContextMenuHelper, [669](#)
  - Digikam::DPopupFrame, [1350](#)
  - Digikam::ImportContextMenuHelper, [2027](#)
  - Digikam::PanIconFrame, [2815](#)
- execDBAction
  - Digikam::BdEngineBackend, [484](#)
- execDBActionQuery
  - Digikam::BdEngineBackend, [484](#)
- execDialog
  - Digikam::Setup, [3161](#)
- execDirectSql
  - Digikam::BdEngineBackend, [484](#)
- execDirectSqlWithResult
  - Digikam::BdEngineBackend, [484](#)
- execQuery
  - Digikam::BdEngineBackend, [485](#)
- execSinglePage
  - Digikam::Setup, [3161](#)
  - ShowFoto::ShowfotoSetup, [3724](#)
- execSql
  - Digikam::BdEngineBackend, [485](#)
- execUpsertDBAction
  - Digikam::BdEngineBackend, [486](#)
- execute
  - Digikam::LoadingTask, [2560](#)
  - Digikam::PreviewLoadingTask, [2853](#)
  - Digikam::SavingTask, [3007](#)
  - Digikam::SharedLoadingTask, [3194](#)
  - Digikam::ThumbnailLoadingTask, [3475](#)
- exifOrientation
  - Digikam::DImg, [1081](#)
  - Digikam::LoadSaveThread, [2570](#)
  - Digikam::MetaEngineRotation, [2711](#)
- exifRotate
  - Digikam::MetaEngineSettings, [2713](#)
- exifRotated
  - Digikam::Canvas, [566](#)
- exifToolAvailable
  - Digikam::ExifToolParser, [1522](#)

- Digikam::ExifToolProcess, [1528](#)
- ExifToolBackend
  - Digikam::MetaEngine, [2673](#)
- ExifToolData
  - Digikam::ExifToolParser, [1520](#)
- exifToolError
  - Digikam::ExifToolProcess, [1529](#)
- exifToolErrorString
  - Digikam::ExifToolProcess, [1529](#)
- exifToolsBusy
  - Digikam::ExifToolProcess, [1529](#)
- ExifToolProcess
  - Digikam::ExifToolProcess, [1528](#)
- exifTransform
  - Digikam::JPEGUtils::JpegRotator, [2478](#)
- exitCode
  - Digikam::ProcessLauncher, [2867](#)
- Exiv2Backend
  - Digikam::MetaEngine, [2673](#)
- Exiv2Version
  - Digikam::MetaEngine, [2678](#)
- expand
  - Digikam::Sidebar, [3218](#)
- expandEverything
  - Digikam::AbstractAlbumTreeView, [153](#)
- expandMatches
  - Digikam::AbstractAlbumTreeView, [153](#)
- ExplicitBranch
  - Digikam::FilterAction, [1706](#)
- expoCorrection
  - Digikam::DRawDecoderSettings, [1379](#)
- expoCorrectionHighlight
  - Digikam::DRawDecoderSettings, [1379](#)
- expoCorrectionShift
  - Digikam::DRawDecoderSettings, [1379](#)
- exportChanges
  - Digikam::MetaEngine, [2678](#)
- exportWidget
  - Digikam::DPluginDImg, [1324](#)
  - Digikam::DPluginLoader, [1337](#)
- exposureIndex
  - Digikam::DRawInfo, [1391](#)
- exposureIndicatorMode
  - Digikam::ExposureSettingsContainer, [1539](#)
- exposureProgram
  - Digikam::DRawInfo, [1391](#)
- exposureTime
  - Digikam::DRawInfo, [1391](#)
- extraAboutData
  - Digikam::DPlugin, [1300](#)
  - Digikam::DPluginDImg, [1325](#)
- extraAboutDataRowTitles
  - Digikam::DPlugin, [1300](#)
  - Digikam::DPluginDImg, [1325](#)
- extraAboutDataTitle
  - Digikam::DPlugin, [1300](#)
  - Digikam::DPluginDImg, [1325](#)
- extractIptcTagString
  - Digikam::MetaEngine::Private, [2705](#)
- extractJsonForItem
  - Digikam::DTrash, [1421](#)
- Extractor
  - Digikam::MLPipelineFoundation, [2729](#)
- extractor
  - Digikam::FacePipelineDetect, [1579](#)
  - Digikam::FacePipelineDetectRecognize, [1583](#)
  - Digikam::FacePipelineEdit, [1587](#)
  - Digikam::FacePipelineRecognize, [1601](#)
  - Digikam::FacePipelineReset, [1605](#)
  - Digikam::FacePipelineRetrain, [1609](#)
- extractRAWData
  - Digikam::DRawDecoder, [1370](#)
- extraData
  - Digikam::Album, [256](#)
- ExtraDataDuplicateCount
  - Digikam::ImportItemModel, [2068](#)
  - Digikam::ItemModel, [2348](#)
  - ShowFoto::ShowfotoItemModel, [3698](#)
- ExtraDataRole
  - Digikam::ImportItemModel, [2068](#)
  - Digikam::ItemModel, [2348](#)
  - ShowFoto::ShowfotoItemModel, [3698](#)
- ExtraRoles
  - Digikam::CategorizedItemModel, [577](#)
- FACE
  - Digikam::Album, [255](#)
- faceCount
  - Digikam::ItemInfo, [2298](#)
- faceCountForPersonInImage
  - Digikam::FaceTagsEditor, [1641](#)
- FaceDbAccess
  - Digikam::FaceDbAccess, [1547](#)
- FaceDbAccessUnlock
  - Digikam::FaceDbAccessUnlock, [1548](#)
- FaceDbOperationGroup
  - Digikam::FaceDbOperationGroup, [1551](#)
- FaceDetectionModel
  - Digikam::FaceScanSettings, [1624](#)
- FaceDetectionSize
  - Digikam::FaceScanSettings, [1624](#)
- FaceDetector
  - Digikam::FaceDetector, [1553](#)
- faceenum2size
  - Digikam, [135](#)
- FaceGroup
  - Digikam::FaceGroup, [1557](#)
- faceNameForTag
  - Digikam::FaceTags, [1636](#)
- faceParameters
  - Digikam::DbEngineParameters, [844](#)
- FaceRecognitionModel
  - Digikam::FaceScanSettings, [1624](#)
- faceRectToDisplayRect
  - Digikam::FaceUtils, [1650](#)
- FaceType
  - Digikam::DConfigDlg, [911](#)

- Digikam::DConfigDlgView, [938](#)
- faceType
  - Digikam::DConfigDlgView, [939](#)
- FastButLargePreview
  - Digikam::PreviewSettings, [2861](#)
- FastPreview
  - Digikam::PreviewSettings, [2861](#)
- FavoriteFolder
  - ShowFoto::ShowfotoStackViewFavoriteItem, [3734](#)
- FavoriteItem
  - ShowFoto::ShowfotoStackViewFavoriteItem, [3734](#)
- FavoriteRoot
  - ShowFoto::ShowfotoStackViewFavoriteItem, [3734](#)
- FavoriteType
  - ShowFoto::ShowfotoStackViewFavoriteItem, [3734](#)
- FFMpegBackend
  - Digikam::MetaEngine, [2673](#)
- fieldOperator
  - Digikam::SearchXmlReader, [3148](#)
- FILE\_NOT\_SUPPORTED
  - Digikam::DNGWriter, [1212](#)
- fileChanged
  - Digikam::LoadingCache, [2545](#)
  - Digikam::LoadingCacheInterface, [2551](#)
- FileDate
  - ShowFoto::ShowfotoFolderViewList, [3685](#)
  - ShowFoto::ShowfotoStackViewList, [3741](#)
- fileFormat
  - Digikam::DImg, [1081](#)
- fileModified
  - Digikam::ItemScanner, [2398](#)
- fileName
  - Digikam::ThumbnailInfo, [3471](#)
- fileOriginData
  - Digikam::DImg, [1081](#)
- filePath
  - Digikam::lccProfile, [1924](#)
  - Digikam::ItemInfo, [2298](#)
  - Digikam::ThumbnailIdentifier, [3466](#)
- FileSaveOptionsBox
  - Digikam::FileSaveOptionsBox, [1687](#)
- FileScanMode
  - Digikam::CollectionScanner, [624](#)
- fileSize
  - Digikam::ItemInfo, [2298](#)
- fileThumbnailInfo
  - Digikam::ThumbnailCreator, [3462](#)
- fileUrl
  - Digikam::CoreDbUrl, [740](#)
  - Digikam::ItemInfo, [2298](#)
- fill
  - Digikam::DImg, [1082](#)
- fillCommonContainer
  - Digikam::ItemScanner, [2399](#)
- fillFromOtherCurves
  - Digikam::ImageCurves, [1956](#)
- fillImageData
  - Digikam::Haar::ImageData, [1873](#)
- fillTemplate
  - Digikam::ItemCopyright, [2205](#)
- fillVideoMetadataContainer
  - Digikam::ItemScanner, [2399](#)
- fillWeeksCombo
  - Digikam::DDatePicker::Private, [976](#)
- FilmGrainFilter
  - Digikam::FilmGrainFilter, [1701](#)
- filter
  - Digikam::DPluginConfView, [1311](#)
  - Digikam::EditorToolThreaded, [1481](#)
  - ShowFoto::ShowfotoStackViewFavoriteList, [3737](#)
- filterAction
  - Digikam::AntiVignettingFilter, [381](#)
  - Digikam::AutoExpoFilter, [422](#)
  - Digikam::AutoLevelsFilter, [426](#)
  - Digikam::BCGFilter, [478](#)
  - Digikam::BlurFilter, [508](#)
  - Digikam::BlurFXFilter, [512](#)
  - Digikam::BorderFilter, [525](#)
  - Digikam::BWSepiaFilter, [536](#)
  - Digikam::CBFilter, [581](#)
  - Digikam::CharcoalFilter, [587](#)
  - Digikam::ColorFXFilter, [640](#)
  - Digikam::ContentAwareFilter, [658](#)
  - Digikam::CurvesFilter, [754](#)
  - Digikam::DImgBuiltinFilter, [1091](#)
  - Digikam::DImgThreadedFilter, [1117](#)
  - Digikam::DistortionFXFilter, [1150](#)
  - Digikam::EmbossFilter, [1501](#)
  - Digikam::EqualizeFilter, [1509](#)
  - Digikam::FilmFilter, [1697](#)
  - Digikam::FilmGrainFilter, [1701](#)
  - Digikam::FilterActionFilter, [1711](#)
  - Digikam::FreeRotationFilter, [1746](#)
  - Digikam::GreycstorationFilter, [1862](#)
  - Digikam::HotPixelFixer, [1901](#)
  - Digikam::HSLFilter, [1911](#)
  - Digikam::lccTransformFilter, [1948](#)
  - Digikam::InfraredFilter, [2146](#)
  - Digikam::InvertFilter, [2153](#)
  - Digikam::LensDistortionFilter, [2506](#)
  - Digikam::LensFunFilter, [2512](#)
  - Digikam::LevelsFilter, [2519](#)
  - Digikam::LocalContrastFilter, [2577](#)
  - Digikam::MixerFilter, [2723](#)
  - Digikam::NormalizeFilter, [2774](#)
  - Digikam::NRFilter, [2789](#)
  - Digikam::OilPaintFilter, [2794](#)
  - Digikam::RainDropFilter, [2908](#)
  - Digikam::RawProcessingFilter, [2939](#)
  - Digikam::RedEyeCorrectionFilter, [2955](#)
  - Digikam::RefocusFilter, [2960](#)
  - Digikam::SharpenFilter, [3204](#)
  - Digikam::ShearFilter, [3209](#)
  - Digikam::StretchFilter, [3273](#)
  - Digikam::TextureFilter, [3455](#)
  - Digikam::TonalityFilter, [3516](#)

- Digikam::UnsharpMaskFilter, [3573](#)
- Digikam::WBFilter, [3609](#)
- FilterActionFilter
  - Digikam::FilterActionFilter, [1711](#)
- filterAlbum
  - Digikam::AbstractAlbumModel, [144](#)
- FilterBehavior
  - Digikam::AlbumFilterModel, [269](#)
- filteredModel
  - Digikam::TagTreeView, [3428](#)
- filterIcon
  - Digikam::DImgFilterManager, [1100](#)
- filterIdentifier
  - Digikam::AntiVignettingFilter, [381](#)
  - Digikam::AutoExpoFilter, [422](#)
  - Digikam::AutoLevelsFilter, [426](#)
  - Digikam::BCGFilter, [478](#)
  - Digikam::BlurFilter, [508](#)
  - Digikam::BlurFXFilter, [512](#)
  - Digikam::BorderFilter, [525](#)
  - Digikam::BWSepiaFilter, [536](#)
  - Digikam::CBFilter, [581](#)
  - Digikam::CharcoalFilter, [587](#)
  - Digikam::ColorFXFilter, [640](#)
  - Digikam::ContentAwareFilter, [658](#)
  - Digikam::CurvesFilter, [754](#)
  - Digikam::DImgThreadedFilter, [1118](#)
  - Digikam::DistortionFXFilter, [1150](#)
  - Digikam::EmbossFilter, [1501](#)
  - Digikam::EqualizeFilter, [1509](#)
  - Digikam::FilmFilter, [1697](#)
  - Digikam::FilmGrainFilter, [1701](#)
  - Digikam::FilterActionFilter, [1712](#)
  - Digikam::FreeRotationFilter, [1746](#)
  - Digikam::GreycstorationFilter, [1862](#)
  - Digikam::HotPixelFixer, [1901](#)
  - Digikam::HSLFilter, [1911](#)
  - Digikam::lccTransformFilter, [1948](#)
  - Digikam::InfraredFilter, [2146](#)
  - Digikam::InvertFilter, [2153](#)
  - Digikam::LensDistortionFilter, [2506](#)
  - Digikam::LensFunFilter, [2512](#)
  - Digikam::LevelsFilter, [2519](#)
  - Digikam::LocalContrastFilter, [2577](#)
  - Digikam::MixerFilter, [2723](#)
  - Digikam::NormalizeFilter, [2774](#)
  - Digikam::NRFilter, [2789](#)
  - Digikam::OilPaintFilter, [2794](#)
  - Digikam::RainDropFilter, [2908](#)
  - Digikam::RawProcessingFilter, [2939](#)
  - Digikam::RedEyeCorrectionFilter, [2955](#)
  - Digikam::RefocusFilter, [2960](#)
  - Digikam::SharpenFilter, [3204](#)
  - Digikam::ShearFilter, [3209](#)
  - Digikam::StretchFilter, [3273](#)
  - Digikam::TextureFilter, [3455](#)
  - Digikam::TonalityFilter, [3516](#)
  - Digikam::UnsharpMaskFilter, [3573](#)
  - Digikam::WBFilter, [3609](#)
  - filterImage
    - Digikam::DImgThreadedFilter, [1118](#)
    - Digikam::FilterActionFilter, [1712](#)
    - Digikam::lccTransformFilter, [1948](#)
    - Digikam::RawProcessingFilter, [2939](#)
    - Digikam::WBFilter, [3609](#)
  - filterMatches
    - Digikam::ItemFilterModel, [2245](#)
  - filterMatchesForText
    - Digikam::ItemFilterModel, [2245](#)
  - FilterMode
    - Digikam::FacePipeline, [1566](#)
    - Digikam::FacePipelineBase, [1574](#)
  - filterModel
    - Digikam::AlbumSelectComboBox, [332](#)
    - Digikam::ImportCategorizedView, [2016](#)
    - Digikam::ItemCategorizedView, [2187](#)
    - Digikam::ItemViewCategorized, [2445](#)
    - ShowFoto::ShowfotoCategorizedView, [3654](#)
  - FilterModelRoles
    - Digikam::ItemModel, [2348](#)
  - filterPattern
    - Digikam::DRawInfo, [1391](#)
  - filterSettingsChanged
    - Digikam::ItemFilterModel, [2245](#)
  - FilterSideBarWidget
    - Digikam::FilterSideBarWidget, [1716](#)
  - FiltersTool
    - Digikam::BatchTool, [465](#)
  - find
    - Digikam::ThumbnailLoadThread, [3481](#)
  - findActionByName
    - Digikam::DPluginEditor, [1329](#)
    - Digikam::DPluginGeneric, [1333](#)
  - findAlbum
    - Digikam::AlbumManager, [292](#)
  - findAll
    - Digikam::ThumbsDb, [3487](#)
  - findAndDeleteSpacersOrNewTags
    - Digikam::RGTagModel, [2985](#)
  - findBuffered
    - Digikam::ThumbnailLoadThread, [3482](#)
  - findByAttributes
    - Digikam::FacialRecognitionWrapper::Private, [1656](#)
  - findByFilePath
    - Digikam::ThumbsDb, [3487](#)
  - findByNameAndCreationDate
    - Digikam::CoreDB, [688](#)
  - findDAlbum
    - Digikam::AlbumManager, [292](#)
  - findDuplicates
    - Digikam::HaarIface, [1877](#)
  - Finder
    - Digikam::MLPipelineFoundation, [2729](#)
  - finder
    - Digikam::FacePipelineDetect, [1579](#)
    - Digikam::FacePipelineDetectRecognize, [1583](#)

- Digikam::FacePipelineEdit, [1587](#)
- Digikam::FacePipelineRecognize, [1601](#)
- Digikam::FacePipelineReset, [1605](#)
- Digikam::FacePipelineRetrain, [1609](#)
- FinderMode
  - Digikam::NewItemFinder, [2761](#)
- findExecutable
  - Digikam::DFileOperations, [1021](#)
- findGroup
  - Digikam::ThumbnailLoadThread, [3482](#)
- findIdentity
  - Digikam::FacialRecognitionWrapper, [1654](#)
  - Digikam::IdentityProvider, [1952](#)
- findImageld
  - Digikam::CoreDB, [688](#)
- findInDownloadHistory
  - Digikam::CoreDB, [688](#)
- findItembyId
  - Digikam::ProgressManager, [2879](#)
- findOrCreateTAlbums
  - Digikam::AlbumManager, [293](#)
- findOriginalText
  - Digikam::SqueezedComboBox, [3254](#)
- findPALbum
  - Digikam::AlbumManager, [293](#)
- findSAlbum
  - Digikam::AlbumManager, [294](#)
- findSAlbumsBySearchType
  - Digikam::AlbumManager, [294](#)
- findTagsWithProperty
  - Digikam::AlbumManager, [296](#)
- findTAlbum
  - Digikam::AlbumManager, [296](#)
- findToolByName
  - Digikam::DPluginBqm, [1308](#)
- findVolumeForLocation
  - Digikam::CollectionManager::Private, [619](#)
- findVolumeForUrl
  - Digikam::CollectionManager::Private, [619](#)
- findWidgets
  - Digikam::DWItemDelegatePool, [1438](#)
- findZeroDegree
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1841](#)
- FingerPrintsGenerator
  - Digikam::FingerPrintsGenerator, [1726](#)
- finish
  - Digikam::SearchXmlWriter, [3152](#)
- finishCompleteScan
  - Digikam::CollectionScanner, [625](#)
- finished
  - Digikam::DImgThreadedFilter, [1118](#)
  - Digikam::HidingStateChanger, [1884](#)
- finishedScanningAlbumRoot
  - Digikam::CollectionScanner, [625](#)
- finishField
  - Digikam::SearchXmlWriter, [3152](#)
- finishFileMetadataWrite
  - Digikam::ScanController, [3012](#)
- finishGroup
  - Digikam::SearchXmlWriter, [3153](#)
- finishScanner
  - Digikam::CollectionScanner::Private, [629](#)
- firmware
  - Digikam::DRawInfo, [1392](#)
- firstChild
  - Digikam::Album, [257](#)
- fitToSelect
  - Digikam::Canvas, [566](#)
- fitToSize
  - Digikam::ImageZoomSettings, [2007](#)
- fitToSizeZoomFactor
  - Digikam::ImageZoomSettings, [2007](#)
- fixColorsHighlights
  - Digikam::DRawDecoderSettings, [1379](#)
- Flag
  - Digikam::AbstractAlbumTreeView, [150](#)
  - Digikam::FilterAction, [1706](#)
- flags
  - Digikam::TableViewModel, [3325](#)
- FLASH
  - Digikam::VidSlideSettings, [3594](#)
- flashUsed
  - Digikam::DRawInfo, [1392](#)
- FlipHorizontal
  - Digikam::MetaEngineRotation, [2710](#)
- FlipVertical
  - Digikam::MetaEngineRotation, [2710](#)
- FlushSignals
  - Digikam::WorkerObject, [3617](#)
- fo
  - Digikam::GeodeticCalculator, [1767](#)
- focalLength
  - Digikam::DRawInfo, [1392](#)
- FocusColor
  - Digikam::SchemeManager, [3023](#)
- focusedIndex
  - Digikam::DWItemDelegate, [1436](#)
- focusLineEdit
  - Digikam::AdvancedRenameWidget, [246](#)
- FocusPoint
  - Digikam::FocusPoint, [1730](#)
- FolderViewRole
  - ShowFoto::ShowfotoFolderViewList, [3684](#)
- font
  - Digikam::DFontProperties, [1030](#)
- FontColumn
  - Digikam::DFontProperties, [1027](#)
- FontDiff
  - Digikam::DFontProperties, [1028](#)
- fontDiffFlags
  - Digikam::DFontProperties, [1030](#)
- FontListCriteria
  - Digikam::DFontProperties, [1028](#)
- fontRoleData
  - Digikam::AbstractAlbumModel, [144](#)



- Digikam::TagModel, [3378](#)
- fontSelected
  - Digikam::DFontProperties, [1030](#)
- fontSize
  - Digikam::DDatePicker, [973](#)
- fontsize
  - Digikam::DDataTable::Private, [985](#)
- foreground
  - Digikam::SchemeManager, [3027](#)
- ForegroundRole
  - Digikam::SchemeManager, [3023](#)
- FORMAT
  - Digikam::DImg, [1075](#)
  - Digikam::FileSaveOptionsBox, [1686](#)
- format
  - Digikam::DImg, [1082](#)
  - Digikam::ItemInfo, [2298](#)
- formatToString
  - Digikam::ItemScanner, [2399](#)
- ForRecognition
  - Digikam::FacePipelineFaceTagsIface, [1593](#)
- fractionMagicValue
  - Digikam::CustomStepsIntSpinBox, [763](#)
- fromAlbumAndName
  - Digikam::CoreDbUrl, [740](#)
- fromDateForMonth
  - Digikam::CoreDbUrl, [740](#)
- fromDateForYear
  - Digikam::CoreDbUrl, [741](#)
- fromDateRange
  - Digikam::CoreDbUrl, [741](#)
- fromFileUrl
  - Digikam::CoreDbUrl, [741](#)
- fromImage
  - Digikam::UndoMetadataContainer, [3565](#)
- fromInfo
  - Digikam::ItemHistoryGraph, [2276](#)
- fromListing
  - Digikam::FaceTagsIface, [1646](#)
- fromLocalFile
  - Digikam::ItemInfo, [2299](#)
- fromLocationAlbumAndName
  - Digikam::ItemInfo, [2299](#)
- fromRFC3066
  - Digikam::DOnlineTranslator, [1273](#)
- fromSourceIndex
  - Digikam::RGTagModel, [2985](#)
- fromTagIds
  - Digikam::CoreDbUrl, [741](#)
- fromUniqueHash
  - Digikam::ItemInfo, [2299](#)
- fromVariant
  - Digikam::FaceTagsIface, [1646](#)
- FS\_ALBUMGUI
  - Digikam, [125](#)
- FS\_EDITOR
  - Digikam, [125](#)
- FS\_IMPORTUI
  - Digikam, [125](#)
- FS\_LIGHTTABLE
  - Digikam, [125](#)
- FS\_NONE
  - Digikam, [125](#)
- FS\_SIDEBARS
  - Digikam, [125](#)
- FS\_STATUSBAR
  - Digikam, [125](#)
- FS\_THUMBBAR
  - Digikam, [125](#)
- FS\_TOOLBARS
  - Digikam, [125](#)
- fsOptions
  - Digikam::DXmlGuiWindow::Private, [1449](#)
- fulfillsRestrictions
  - Digikam::HaarIface, [1877](#)
- FULL\_SIZE
  - Digikam::DNGWriter, [1213](#)
- FullFiltering
  - Digikam::AlbumFilterModel, [269](#)
- FullImage
  - Digikam::ImageIface, [1969](#)
- FullImageHistogram
  - Digikam, [126](#)
- fullScreenAction
  - Digikam::DXmlGuiWindow::Private, [1449](#)
- fullScreenBtn
  - Digikam::DXmlGuiWindow::Private, [1449](#)
- fullScreenHideSideBars
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- fullScreenHideStatusBar
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- fullScreenHideThumbBar
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- fullScreenHideToolBars
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- fullScreenIsActive
  - Digikam::DXmlGuiWindow, [1447](#)
- FullScreenOptions
  - Digikam, [125](#)
- fullScreenParent
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- fullSize
  - Digikam::DRawInfo, [1392](#)
- FullWrite
  - Digikam::DisjointMetadata, [1139](#)
  - Digikam::MetadataHub, [2634](#)
- FullWritelfChanged
  - Digikam::DisjointMetadata, [1139](#)
  - Digikam::MetadataHub, [2634](#)
- FuzzySelection
  - Digikam::TimeLineWidget, [3509](#)
- generatedName
  - Digikam::AlbumLabelsSearchHandler, [282](#)
- generateObjects
  - Digikam::DNNBaseDetectorModel, [1217](#), [1218](#)
- generateTagsList

- Digikam::AutoTagsAssign, [427](#)
- generateUrls
  - Digikam::DOnlineTts, [1288](#)
- Generic
  - Digikam::DPluginAction, [1304](#)
- GenericExport
  - Digikam::DPluginAction, [1304](#)
- GenericImport
  - Digikam::DPluginAction, [1304](#)
- GenericMetadata
  - Digikam::DPluginAction, [1304](#)
- GenericTool
  - Digikam::DPluginAction, [1304](#)
- GenericView
  - Digikam::DPluginAction, [1304](#)
- geoCoordinates
  - Digikam::BackendGoogleMaps, [443](#)
  - Digikam::BackendMarble, [450](#)
- GeodeticCalculator
  - Digikam::GeodeticCalculator, [1762](#)
- GeoGroupStateEnum
  - Digikam, [125](#)
- GeofaceHelperParseLatLonString
  - Digikam, [128](#)
- GeofaceMinMarkerGroupingRadius
  - Digikam, [136](#)
- GeoPainter\_drawPixmapAtCoordinates
  - Digikam::BackendMarble, [451](#)
- getActionValue
  - Digikam::DbEngineActionType, [835](#)
- getActiveState
  - Digikam::MapWidgetView, [2629](#)
- getActiveTab
  - Digikam::Sidebar, [3218](#)
- getAlbumAndSubalbumsForPath
  - Digikam::CoreDB, [689](#)
- getAlbumAverageDate
  - Digikam::CoreDB, [689](#)
- getAlbumForPath
  - Digikam::CoreDB, [689](#)
- getAlbumHighestDate
  - Digikam::CoreDB, [690](#)
- getAlbumLowestDate
  - Digikam::CoreDB, [690](#)
- getAlbumModificationDate
  - Digikam::CoreDB, [690](#)
- getAlbumModificationMap
  - Digikam::CoreDB, [691](#)
- getAlbumRelativePath
  - Digikam::CoreDB, [691](#)
- getAlbumRootId
  - Digikam::CoreDB, [691](#)
- getAlbumRoots
  - Digikam::CoreDB, [691](#)
- getAlbumShortInfos
  - Digikam::CoreDB, [692](#)
- getAlbumsOnAlbumRoot
  - Digikam::CoreDB, [692](#)
- getAlbumThumbnail
  - Digikam::AlbumThumbnailLoader, [357](#)
- getAlbumThumbnailDirectly
  - Digikam::AlbumThumbnailLoader, [357](#)
- getAllCreationDates
  - Digikam::CoreDB, [692](#)
- getAllItems
  - Digikam::CoreDB, [692](#)
- getAllItemsWithAlbum
  - Digikam::CoreDB, [692](#)
- getAudioCodecsProperties
  - Digikam::FFmpegConfigHelper, [1659](#)
- getAutodetectedPersonString
  - Digikam::FaceTagsIface, [1646](#)
- getBestAndWorstPossibleScore
  - Digikam::HaarIface, [1878](#)
- getCacheSize
  - Digikam::LoadingCache, [2545](#)
- getCameraSerialNumber
  - Digikam::DMetadata, [1184](#)
- getCaption
  - Digikam::AlbumFolderViewSideBarWidget, [276](#)
  - Digikam::DateFolderViewSideBarWidget, [814](#)
  - Digikam::FuzzySearchSideBarWidget, [1755](#)
  - Digikam::GPSSearchSideBarWidget, [1830](#)
  - Digikam::LabelsSideBarWidget, [2497](#)
  - Digikam::PeopleSideBarWidget, [2832](#)
  - Digikam::SearchSideBarWidget, [3121](#)
  - Digikam::SidebarWidget, [3226](#)
  - Digikam::TagViewSideBarWidget, [3434](#)
  - Digikam::TimelineSideBarWidget, [3506](#)
- getCenter
  - Digikam::BackendGoogleMaps, [443](#)
  - Digikam::BackendMarble, [451](#)
- getClosestNeighbors
  - Digikam::KDNodeBase, [2481](#)
  - Digikam::KDTreeBase, [2488](#)
- getColorInfos
  - Digikam::MapWidget, [2623](#)
- getColumnFlags
  - Digikam::TableViewColumn, [3287](#)
  - Digikam::TableViewColumns::ColumnAudioVideoProperties, [3293](#)
  - Digikam::TableViewColumns::ColumnDigikamProperties, [3297](#)
  - Digikam::TableViewColumns::ColumnFileProperties, [3302](#)
  - Digikam::TableViewColumns::ColumnGeoProperties, [3307](#)
  - Digikam::TableViewColumns::ColumnItemProperties, [3311](#)
  - Digikam::TableViewColumns::ColumnPhotoProperties, [3316](#)
  - Digikam::TableViewColumns::ColumnThumbnail, [3320](#)
- getComments
  - Digikam::MetaEngine, [2679](#)
- getCommentsDecoded

- Digikam::MetaEngine, [2679](#)
- getComponentValue
  - Digikam, [128](#)
- getComposer
  - Digikam::DColorComposer, [901](#)
- getConfigGroup
  - Digikam::StateSavingObject, [3262](#)
- getConfigurationWidget
  - Digikam::TableViewColumns::ColumnFileProperties, [3302](#)
  - Digikam::TableViewColumns::ColumnGeoProperties, [3307](#)
  - Digikam::TableViewColumns::ColumnPhotoProperties, [3316](#)
- getContainer
  - Digikam::ImageCurves, [1956](#)
- getCopyrightInformation
  - Digikam::DMetadata, [1184](#)
- getCoreDatabaseNameOrDir
  - Digikam::DbEngineParameters, [844](#)
- getCurrentHighlightState
  - Digikam::SearchTextBar, [3125](#)
- getCustomProperty
  - Digikam::DConfigDlgMgr, [921](#)
- getCustomPropertyChangedSignal
  - Digikam::DConfigDlgMgr, [921](#)
- getDAAlbumsCount
  - Digikam::AlbumManager, [296](#)
- getData
  - Digikam::UndoCache, [3564](#)
- getDatabaseEncoding
  - Digikam::CoreDB, [692](#)
- getDatabaseFieldsRaw
  - Digikam::ItemInfo, [2299](#)
- getDBAction
  - Digikam::BdEngineBackend, [486](#)
- getDbValue
  - Digikam::CommonKeys, [652](#)
  - Digikam::DbKeysCollection, [865](#)
  - Digikam::MetadataKeys, [2641](#)
  - Digikam::PositionKeys, [2848](#)
- getDecoratedPixmapForCluster
  - Digikam::MapWidget, [2624](#)
- getDescription
  - Digikam::TableViewColumns::ColumnDigikamProperties, [3297](#)
- getDigitizationDateTime
  - Digikam::MetaEngine, [2679](#)
- getDirtyOrMissingFaceImageUrls
  - Digikam::CoreDB, [693](#)
- getDirtyOrMissingFingerprints
  - Digikam::SimilarityDb, [3228](#)
- getDirtyOrMissingFingerprintURLs
  - Digikam::SimilarityDb, [3229](#)
- getDownloadInformation
  - Digikam::DNNModelManager, [1233](#)
- getErrorMessage
  - Digikam::BackendGeonamesRG, [436](#)
- Digikam::BackendGeonamesUSRG, [439](#)
- Digikam::BackendOsmRG, [458](#)
- Digikam::RGBBackend, [2978](#)
- getExifComment
  - Digikam::MetaEngine, [2679](#)
- getExifEncoded
  - Digikam::MetaEngine, [2679](#)
- getExifTagComment
  - Digikam::MetaEngine, [2679](#)
- getExifTagData
  - Digikam::MetaEngine, [2680](#)
- getExifTagDescription
  - Digikam::MetaEngine, [2680](#)
- getExifTagLong
  - Digikam::MetaEngine, [2680](#)
- getExifTagRational
  - Digikam::MetaEngine, [2680](#)
- getExifTagsDataList
  - Digikam::MetaEngine, [2680](#)
- getExifTagString
  - Digikam::MetaEngine, [2681](#)
- getExifTagTitle
  - Digikam::MetaEngine, [2681](#)
- getExifTagVariant
  - Digikam::MetaEngine, [2681](#)
- getExifThumbnail
  - Digikam::MetaEngine, [2681](#)
- getExifToolResult
  - Digikam::ExifToolProcess, [1529](#)
- getExtensionsProperties
  - Digikam::FFmpegConfigHelper, [1659](#)
- getFaceCount
  - Digikam::AlbumManager, [296](#)
- getFaceEmbedding
  - Digikam::DNNOpenFaceExtractor, [1240](#)
  - Digikam::DNNSFaceExtractor, [1244](#)
- getFaceThumbnailDirectly
  - Digikam::AlbumThumbnailLoader, [357](#)
- getFilePath
  - Digikam::MetaEngine, [2682](#)
- getFilterSettings
  - Digikam::CoreDB, [693](#)
- getFirstItemWithFaceTag
  - Digikam::CoreDB, [693](#)
- getFolders
  - Digikam::GPCamera, [1788](#)
  - Digikam::UMSCamera, [3558](#)
- getFontList
  - Digikam::DFontProperties, [1030](#)
- getFormatStatistics
  - Digikam::CoreDB, [693](#)
- getFreeSpace
  - Digikam::DKCamera, [1166](#)
  - Digikam::GPCamera, [1788](#)
  - Digikam::UMSCamera, [3559](#)
- getGlobalGroupState
  - Digikam::GPSMarkerTiler, [1821](#)
  - Digikam::ItemMarkerTiler, [2340](#)

- getGPSAltitude
  - Digikam::MetaEngine, [2682](#)
- getGPSInfo
  - Digikam::MetaEngine, [2682](#)
- getGPSLatitudeNumber
  - Digikam::MetaEngine, [2682](#)
- getGPSLatitudeString
  - Digikam::MetaEngine, [2682](#)
- getGraph
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1841](#)
- getGroupingOperateOnAll
  - Digikam::ApplicationSettings, [388](#)
- getHSL
  - Digikam::DColor, [898](#)
- getIccProfile
  - Digikam::DMetadata, [1184](#)
- getIcon
  - Digikam::AlbumFolderViewSideBarWidget, [276](#)
  - Digikam::DateFolderViewSideBarWidget, [814](#)
  - Digikam::FuzzySearchSideBarWidget, [1755](#)
  - Digikam::GPSSearchSideBarWidget, [1830](#)
  - Digikam::LabelsSideBarWidget, [2497](#)
  - Digikam::PeopleSideBarWidget, [2832](#)
  - Digikam::SearchSideBarWidget, [3121](#)
  - Digikam::SidebarWidget, [3226](#)
  - Digikam::TagViewSideBarWidget, [3434](#)
  - Digikam::TimelineSideBarWidget, [3506](#)
- getIconFromService
  - Digikam::DServiceMenu, [1402](#)
- getIconShowOverlays
  - Digikam::ApplicationSettings, [389](#)
- getIdentialFiles
  - Digikam::CoreDB, [693](#)
- getIdentity
  - Digikam::KDNodeBase, [2481](#)
- getImageGroupedCount
  - Digikam::ItemInfoCache, [2310](#)
- getImageld
  - Digikam::CoreDB, [693](#)
- getImagelds
  - Digikam::CoreDB, [694](#), [696](#)
- getImageMetadata
  - Digikam::CoreDB, [696](#)
- getImageProperty
  - Digikam::CoreDB, [696](#)
- getImagesFields
  - Digikam::CoreDB, [696](#)
- getImageSimilarity
  - Digikam::SimilarityDb, [3229](#)
- getImageSimilarityAlgorithms
  - Digikam::SimilarityDb, [3229](#)
- getImagesRelatedFrom
  - Digikam::CoreDB, [697](#)
- getImagesRelatingTo
  - Digikam::CoreDB, [697](#)
- getImagesWithImageTagProperty
  - Digikam::CoreDB, [697](#)
- getImagesWithProperty
  - Digikam::CoreDB, [697](#)
- getImageTagProperties
  - Digikam::CoreDB, [697](#)
- getImageUuid
  - Digikam::CoreDB, [697](#)
- getImgSelection
  - Digikam::EditorCore, [1467](#)
- getInputImageSize
  - Digikam::DNNBaseDetectorModel, [1218](#)
- getIptc
  - Digikam::MetaEngine, [2682](#)
- getIptcKeywords
  - Digikam::MetaEngine, [2683](#)
- getIptcSubCategories
  - Digikam::MetaEngine, [2683](#)
- getIptcSubjects
  - Digikam::MetaEngine, [2683](#)
- getIptcTagData
  - Digikam::MetaEngine, [2683](#)
- getIptcTagDescription
  - Digikam::MetaEngine, [2683](#)
- getIptcTagsDataList
  - Digikam::MetaEngine, [2683](#)
- getIptcTagsList
  - Digikam::MetaEngine, [2684](#)
- getIptcTagsStringList
  - Digikam::MetaEngine, [2684](#)
- getIptcTagString
  - Digikam::MetaEngine, [2684](#)
- getIptcTagTitle
  - Digikam::MetaEngine, [2684](#)
- getItemAlbum
  - Digikam::CoreDB, [698](#)
- getItemColorWorkSpace
  - Digikam::MetaEngine, [2684](#)
- getItemComments
  - Digikam::CoreDB, [698](#)
- getItemCommonTagIDs
  - Digikam::CoreDB, [698](#)
- getItemCopyright
  - Digikam::CoreDB, [698](#)
- getItemDateTime
  - Digikam::MetaEngine, [2685](#)
- getItemDimensions
  - Digikam::MetaEngine, [2685](#)
- getItemFacesMap
  - Digikam::DMetadata, [1184](#)
- getItemFromAlbum
  - Digikam::AlbumManager, [297](#)
  - Digikam::CoreDB, [699](#)
- getItemHistory
  - Digikam::CoreDB, [699](#)
- getItemIccProfile
  - Digikam::MetaEngine, [2685](#)
- getItemIDsAndURLsInAlbum
  - Digikam::CoreDB, [699](#)
- getItemIDsInAlbum

- Digikam::CoreDB, [699](#)
- getItemIDsInTag
  - Digikam::CoreDB, [700](#)
- getItemInfo
  - Digikam::GPCamera, [1788](#)
  - Digikam::UMSCamera, [3559](#)
- getItemInformation
  - Digikam::CoreDB, [700](#)
- getItemName
  - Digikam::CoreDB, [700](#)
- getItemNamesInAlbum
  - Digikam::CoreDB, [701](#)
- getItemOrientation
  - Digikam::MetaEngine, [2685](#)
- getItemPosition
  - Digikam::CoreDB, [701](#)
- getItemPreview
  - Digikam::MetaEngine, [2685](#)
- getItemScanInfo
  - Digikam::CoreDB, [701](#)
- getItemScanInfos
  - Digikam::CoreDB, [701](#)
- getItemForUuid
  - Digikam::CoreDB, [701](#)
- getItemShortInfo
  - Digikam::CoreDB, [702](#)
- getItemInfoList
  - Digikam::DKCamera, [1166](#)
  - Digikam::GPCamera, [1788](#)
  - Digikam::UMSCamera, [3559](#)
- getItemTagIDs
  - Digikam::CoreDB, [702](#)
- getItemURLsWithTag
  - Digikam::CoreDB, [702](#)
- getItemTagIDs
  - Digikam::CoreDB, [702](#)
- getItemTagName
  - Digikam::CoreDB, [703](#)
- getItemURLsInAlbum
  - Digikam::CoreDB, [703](#)
- getItemURLsInTag
  - Digikam::CoreDB, [703](#)
- getLegacySetting
  - Digikam::SimilarityDb, [3229](#)
- getLensDescription
  - Digikam::DMetadata, [1185](#)
- getListFromImageMetadata
  - Digikam::CoreDB, [704](#)
- getMakernoteTagsList
  - Digikam::MetaEngine, [2685](#)
- getMarkerModelLevel
  - Digikam::BackendGoogleMaps, [443](#)
  - Digikam::BackendMarble, [451](#)
- getMetadata
  - Digikam::DImg, [1082](#)
  - Digikam::GPCamera, [1788](#)
  - Digikam::UMSCamera, [3559](#)
- getMetadataField
  - Digikam::DMetadata, [1185](#)
- getMetadataTitle
  - Digikam::ExifWidget, [1537](#)
  - Digikam::ICCPProfileWidget, [1935](#)
  - Digikam::IptcWidget, [2167](#)
  - Digikam::MakerNoteWidget, [2603](#)
  - Digikam::XmpWidget, [3639](#)
- getMimeType
  - Digikam::MetaEngine, [2685](#)
- getModel
  - Digikam::DNNModelManager, [1233](#)
- getModelPath
  - Digikam::DNNModelBase, [1230](#)
- getMsecsInfo
  - Digikam::DMetadata, [1186](#)
- getNeedResetExifOrientation
  - Digikam::BatchTool, [467](#)
- getNetworkManager
  - Digikam::NetworkManager, [2758](#)
- getNewConfiguration
  - Digikam::TableViewColumns::ColumnFileConfigurationWidget, [3299](#)
  - Digikam::TableViewColumns::ColumnGeoConfigurationWidget, [3304](#)
  - Digikam::TableViewColumns::ColumnPhotoConfigurationWidget, [3313](#)
- getNewIdsList
  - Digikam::CollectionScanner, [626](#)
  - Digikam::ScanController, [3012](#)
- getNormalizedBounds
  - Digikam::BackendGoogleMaps, [443](#)
  - Digikam::BackendMarble, [451](#)
- getNumberOfAllItemsAndAlbums
  - Digikam::CoreDB, [704](#)
- getNumberOfImagesInAlbums
  - Digikam::CoreDB, [704](#)
- getNumberOfImagesInTagProperties
  - Digikam::CoreDB, [704](#)
- getNumberOfImagesInTags
  - Digikam::CoreDB, [705](#)
- getNumberOfItemsInAlbum
  - Digikam::CoreDB, [705](#)
- getObsoleteItemIds
  - Digikam::CoreDB, [705](#)
- getOneRelatedImageEach
  - Digikam::CoreDB, [705](#)
- getOrCreate
  - Digikam::TagProperties, [3386](#)
- getOrCreateInternalTag
  - Digikam::TagsCache, [3401](#)
- getOrCreateTag
  - Digikam::TagsCache, [3401](#)
- getOrCreateTagForIdentity
  - Digikam::FaceTags, [1636](#)
- getOrCreateTagForPerson
  - Digikam::FaceTags, [1636](#)
- getOrCreateTagWithProperty
  - Digikam::TagsCache, [3401](#)

- getOriginalImageRegionToRender
  - Digikam::ImageRegionWidget, [1994](#)
- getOriginalRegionImage
  - Digikam::ImageRegionWidget, [1994](#)
- getPALbumsCount
  - Digikam::AlbumManager, [297](#)
- getPixelColor
  - Digikam::DImg, [1082](#)
- getPixelSize
  - Digikam::MetaEngine, [2686](#)
- getPosition
  - Digikam::KDNodeBase, [2481](#)
- getPredefinedClasses
  - Digikam::DNNBaseDetectorModel, [1218](#)
- getPreview
  - Digikam::DKCamera, [1166](#)
  - Digikam::GPCamera, [1789](#)
  - Digikam::UMSCamera, [3559](#)
- getProjection
  - Digikam::BackendMarble, [451](#)
- getQuery
  - Digikam::BdEngineBackend, [486](#)
- getRawProgram
  - Digikam::DPluginRawImport, [1342](#)
- getRecentlyAssignedTags
  - Digikam::AlbumManager, [297](#)
  - Digikam::CoreDB, [705](#)
- getRectBySize
  - Digikam::FocusPoint, [1730](#)
- getRelatedImagesToByType
  - Digikam::CoreDB, [705](#)
- getRelationCloud
  - Digikam::CoreDB, [706](#)
- getRequest
  - Digikam::LookupAltitudeGeonames, [2588](#)
- getRequests
  - Digikam::LookupAltitudeGeonames, [2588](#)
- getResetExifOrientationAllowed
  - Digikam::BatchTool, [467](#)
- getSearchInfo
  - Digikam::CoreDB, [706](#)
- getSearchQuery
  - Digikam::CoreDB, [706](#)
- getSelectedArea
  - Digikam::Canvas, [566](#)
- getSemanticInfo
  - Digikam::BalooWrap, [460](#)
- getSetting
  - Digikam::CoreDB, [706](#)
  - Digikam::SimilarityDb, [3230](#)
- getSpacerAddress
  - Digikam::RGTagModel, [2985](#)
- getSpacers
  - Digikam::RGTagModel, [2987](#)
- getStandardTagIcon
  - Digikam::AlbumThumbnailLoader, [357](#)
- getStateSavingDepth
  - Digikam::StateSavingObject, [3262](#)
- getStatus
  - Digikam::LookupAltitudeGeonames, [2588](#)
- getStdExifTagsList
  - Digikam::MetaEngine, [2686](#)
- getStringComparisonType
  - Digikam::ApplicationSettings, [389](#)
- getSuggestedNames
  - Digikam::FaceTagsEditor, [1641](#)
  - Digikam::ItemInfo, [2299](#)
- getTagDescription
  - Digikam::ExifWidget, [1537](#)
  - Digikam::ICCPProfileWidget, [1935](#)
  - Digikam::IptcWidget, [2167](#)
  - Digikam::MakerNoteWidget, [2603](#)
  - Digikam::XmpWidget, [3639](#)
- getTagIdsWithProperties
  - Digikam::CoreDB, [706](#)
- getTagList
  - Digikam::GPSItemContainer, [1805](#)
- getTagProperties
  - Digikam::CoreDB, [707](#)
- getTagRects
  - Digikam::FaceTagsEditor, [1641](#)
- getTagShortInfos
  - Digikam::CoreDB, [707](#)
- getTagsWithProperty
  - Digikam::CoreDB, [707](#)
- getTagThumbnail
  - Digikam::AlbumThumbnailLoader, [357](#)
  - Digikam::SyncJob, [3279](#)
- getTagThumbnailDirectly
  - Digikam::AlbumThumbnailLoader, [357](#)
- getTagTitle
  - Digikam::ExifWidget, [1537](#)
  - Digikam::ICCPProfileWidget, [1935](#)
  - Digikam::IptcWidget, [2167](#)
  - Digikam::MakerNoteWidget, [2603](#)
  - Digikam::XmpWidget, [3639](#)
- getTagType
  - Digikam::RGTagModel, [2987](#)
- getTAlbumsCount
  - Digikam::AlbumManager, [298](#)
- getTemporaryHaarTitle
  - Digikam::SAlbum, [3004](#)
- getTemporaryTitle
  - Digikam::SAlbum, [3005](#)
- getThreshold
  - Digikam::DNNFaceExtractorBase, [1228](#)
  - Digikam::DNNModelBase, [1230](#)
  - Digikam::DNNOpenFaceExtractor, [1240](#)
  - Digikam::DNNSFaceExtractor, [1244](#)
- getThumbInfo
  - Digikam::CameraThumbsCtrl, [554](#)
- getThumbnail
  - Digikam::GPCamera, [1789](#)
  - Digikam::UMSCamera, [3559](#)
- getThumbnailSize
  - Digikam::TrashView, [3543](#)

- getThumbsInfo
  - Digikam::CameraController, [542](#)
- getTile
  - Digikam::AbstractMarkerTiler, [191](#)
  - Digikam::GPSMarkerTiler, [1821](#)
  - Digikam::ItemMarkerTiler, [2341](#)
- getTileGroupState
  - Digikam::AbstractMarkerTiler, [192](#)
  - Digikam::GPSMarkerTiler, [1822](#)
  - Digikam::ItemMarkerTiler, [2341](#)
- getTileMarkerCount
  - Digikam::GPSMarkerTiler, [1822](#)
  - Digikam::ItemMarkerTiler, [2341](#)
- getTileRepresentativeMarker
  - Digikam::AbstractMarkerTiler, [192](#)
  - Digikam::GPSMarkerTiler, [1822](#)
  - Digikam::ItemMarkerTiler, [2341](#)
- getTileSelectedCount
  - Digikam::GPSMarkerTiler, [1822](#)
  - Digikam::ItemMarkerTiler, [2341](#)
- getTitle
  - Digikam::TableViewColumns::ColumnAudioVideoProperties, [3293](#)
  - Digikam::TableViewColumns::ColumnDigikamProperties, [3298](#)
  - Digikam::TableViewColumns::ColumnFileProperties, [3302](#)
  - Digikam::TableViewColumns::ColumnGeoProperties, [3307](#)
  - Digikam::TableViewColumns::ColumnItemProperties, [3311](#)
  - Digikam::TableViewColumns::ColumnPhotoProperties, [3316](#)
  - Digikam::TableViewColumns::ColumnThumbnail, [3320](#)
- getToggleAction
  - Digikam::ThumbBarDock, [3460](#)
- getTrainingData
  - Digikam::IdentityProvider, [1952](#)
- getUnconfirmedFaceCount
  - Digikam::AlbumManager, [298](#)
- getUniqueFileUrl
  - Digikam::DFileOperations, [1021](#)
- getUniqueFolderUrl
  - Digikam::DFileOperations, [1021](#)
- getUniqueHash
  - Digikam::DImg, [1083](#)
- getUniqueHashVersion
  - Digikam::CoreDB, [707](#)
  - Digikam::DImg, [1083](#)
- getUniqueID
  - Digikam::ProgressManager, [2879](#)
- getUserFilterSettings
  - Digikam::CoreDB, [707](#)
- getUserProperty
  - Digikam::DConfigDlgMngr, [921](#)
- getUserPropertyChangedSignal
  - Digikam::DConfigDlgMngr, [922](#)
- getValue
  - Digikam::DbKeysCollection, [867](#)
- getVideoCodecsProperties
  - Digikam::FFmpegConfigHelper, [1660](#)
- getVideoInformation
  - Digikam::DMetadata, [1186](#)
- getVideoMetadata
  - Digikam::CoreDB, [708](#)
- getXmp
  - Digikam::MetaEngine, [2686](#)
- getXmpKeywords
  - Digikam::DMetadata, [1186](#)
  - Digikam::MetaEngine, [2686](#)
- getXmpSubCategories
  - Digikam::DMetadata, [1186](#)
  - Digikam::MetaEngine, [2686](#)
- getXmpSubjects
  - Digikam::DMetadata, [1186](#)
  - Digikam::MetaEngine, [2686](#)
- getXmpTagDescription
  - Digikam::MetaEngine, [2686](#)
- getXmpTagsDataList
  - Digikam::MetaEngine, [2687](#)
- getXmpTagsList
  - Digikam::MetaEngine, [2687](#)
- getXmpTagString
  - Digikam::MetaEngine, [2687](#)
- getXmpTagStringBag
  - Digikam::MetaEngine, [2687](#)
- getXmpTagStringLangAlt
  - Digikam::MetaEngine, [2687](#)
- getXmpTagStringListLangAlt
  - Digikam::MetaEngine, [2688](#)
- getXmpTagStringSeq
  - Digikam::MetaEngine, [2688](#)
- getXmpTagTitle
  - Digikam::MetaEngine, [2688](#)
- getXmpTagVariant
  - Digikam::MetaEngine, [2688](#)
- getYCbCr
  - Digikam::DColor, [899](#)
- getZoom
  - Digikam::BackendGoogleMaps, [443](#)
  - Digikam::BackendMarble, [451](#)
- GivenAsArgument
  - Digikam::FacePipelineFaceTagsIface, [1593](#)
- globalID
  - Digikam::Album, [257](#)
- GlobalSettings
  - Digikam::ImageQualityConfSelector, [1979](#)
- goldenStarPixmap
  - Digikam::LabelsTreeView, [2500](#)
- GPSTlisting
  - Digikam::GPSDBJobsThread, [1799](#)
- GPSMarkerTiler
  - Digikam::GPSMarkerTiler, [1821](#)
- GPSSearchView
  - Digikam::GPSSearchView, [1833](#)

- granularity
  - Digikam::DImgLoaderObserver, [1106](#)
- graph\_traits
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1841](#)
- GreycstorationFilter
  - Digikam::GreycstorationFilter, [1861](#)
- gridSize
  - Digikam::DItemDelegate, [1152](#)
  - Digikam::ItemViewDelegate, [2453](#)
  - Digikam::ItemViewImportDelegate, [2462](#)
  - ShowFoto::ShowfotoItemViewDelegate, [3708](#)
- groupCaption
  - Digikam::SearchXmlReader, [3148](#)
- groupedImages
  - Digikam::ItemInfo, [2299](#)
- GroupedImagesFinder
  - Digikam::GroupedImagesFinder, [1864](#)
- groupImage
  - Digikam::ItemInfo, [2300](#)
- GroupsOpenRole
  - Digikam::ItemFilterModel, [2243](#)
- groupLabelPixmap
  - Digikam::SearchView, [3139](#)
- groupOperator
  - Digikam::SearchXmlReader, [3149](#)
- GRS80
  - Digikam::Ellipsoid, [1494](#)
- halfSizeColorImage
  - Digikam::DRawDecoderSettings, [1379](#)
- handbookChapter
  - Digikam::DPlugin, [1300](#)
- handbookReference
  - Digikam::DPlugin, [1300](#)
- handbookSection
  - Digikam::DPlugin, [1300](#)
- handle
  - Digikam::IccProfile, [1924](#)
- handleCustomContextMenuAction
  - Digikam::AbstractAlbumTreeView, [153](#)
  - Digikam::AlbumSelectTreeView, [350](#)
  - Digikam::EditableSearchTreeView, [1464](#)
  - Digikam::NormalSearchTreeView, [2780](#)
  - Digikam::TagFilterView, [3350](#)
  - Digikam::TagFolderView, [3356](#)
- handleQueryResult
  - Digikam::BdEngineBackend, [486](#)
- handleWithErrorHandler
  - Digikam::BdEngineBackendPrivate, [492](#)
- HAS\_RESULT
  - Digikam::SearchTextBar, [3125](#)
- hasActions
  - Digikam::DImageHistory, [1068](#)
- hasBeenScanned
  - Digikam::FaceUtils, [1650](#)
- hasChanged
  - Digikam::DConfigDlgMgr, [922](#)
- hasCloseButton
  - Digikam::DDatePicker, [973](#)
- hasComments
  - Digikam::MetaEngine, [2688](#)
- hasDerivedImages
  - Digikam::ItemInfo, [2300](#)
- hasDirtyOrMissingFingerprint
  - Digikam::SimilarityDb, [3230](#)
- hasEdges
  - Digikam::ItemHistoryGraph, [2277](#)
- hasErrors
  - Digikam::DBJobsThread, [863](#)
  - Digikam::IOJobsThread, [2161](#)
- hasExif
  - Digikam::MetaEngine, [2689](#)
- hasFingerprint
  - Digikam::SimilarityDb, [3230](#)
- hasFingerprints
  - Digikam::SimilarityDb, [3231](#)
- hasGroupedImages
  - Digikam::ItemInfo, [2300](#)
- hash
  - Digikam::DbEngineParameters, [845](#)
  - Digikam::DownloadInfo, [1291](#)
  - Digikam::FaceTagsIface, [1646](#)
  - Digikam::ItemInfo, [2300](#)
- hasHiddenGroupedImages
  - Digikam::DigikamItemView, [1064](#)
  - Digikam::GroupingViewImplementation, [1871](#)
  - Digikam::ItemThumbnailBar, [2425](#)
  - Digikam::TableViewTreeView, [3329](#)
- hasHistoryToResolve
  - Digikam::ItemScanner, [2399](#)
- hasIccProfile
  - Digikam::DRawInfo, [1392](#)
- hasImageHistory
  - Digikam::CoreDB, [708](#)
- hasIptc
  - Digikam::MetaEngine, [2689](#)
- hasParameters
  - Digikam::FilterAction, [1706](#)
- hasProperty
  - Digikam::TagProperties, [3386](#)
  - Digikam::TagsCache, [3401](#)
- hasRegionSelection
  - Digikam::GeofaceSharedData, [1775](#)
- hasSearchResult
  - Digikam::AlbumFilterModel, [270](#)
- hasSidecar
  - Digikam::MetaEngine, [2689](#)
- hasTag
  - Digikam::TagsCache, [3402](#)
- hasTags
  - Digikam::CoreDB, [708](#)
- hasThumbnail
  - Digikam::ProgressItem, [2870](#)
- hasTransparentPixels
  - Digikam::DImg, [1083](#)
- hasUnconfirmed



- Digikam::FaceGroup, [1557](#)
- hasUnresolvedEntries
  - Digikam::ItemHistoryGraph, [2277](#)
- hasVisibilityProperty
  - Digikam::DPlugin, [1301](#)
  - Digikam::DPluginBqm, [1308](#)
  - Digikam::DPluginDImg, [1325](#)
- hasVisibleItems
  - Digikam::ItemVisibilityController, [2473](#)
- hasXmp
  - Digikam::MetaEngine, [2689](#)
- HDPLUS
  - Digikam::VidSlideSettings, [3596](#)
- HDTV
  - Digikam::VidSlideSettings, [3596](#)
- header
  - Digikam::DConfigDlgWdgItem, [953](#)
- headerData
  - Digikam::TrackListModel, [3527](#)
- HeaderRole
  - Digikam::DConfigDlgModel, [926](#)
- HEIFFiles
  - Digikam::MimeFilter, [2718](#)
- heightForWidth
  - Digikam::DNotificationWidget, [1261](#)
- hiddenAndRemoved
  - Digikam::ItemVisibilityController, [2474](#)
- hide
  - Digikam::AbstractWidgetDelegateOverlay, [202](#)
  - Digikam::ImportRatingOverlay, [2096](#)
  - Digikam::ItemRatingOverlay, [2387](#)
  - Digikam::PersistentWidgetDelegateOverlay, [2836](#)
  - Digikam::TagsLineEditOverlay, [3414](#)
- hideAndRemoveItem
  - Digikam::ItemVisibilityController, [2474](#)
- hideAnimationFinished
  - Digikam::DNotificationWidget, [1262](#)
- hideEvent
  - Digikam::DPopupFrame, [1350](#)
- HidingStateChanger
  - Digikam::HidingStateChanger, [1883](#)
- hierarchyFromParent
  - ShowFoto::ShowfotoStackViewFavoriteItem, [3734](#)
- highlightLineEdit
  - Digikam::AdvancedRenameWidget, [246](#)
- highlightState
  - Digikam::SearchTextBar, [3125](#)
- HighQualityPreview
  - Digikam::PreviewSettings, [2861](#)
- hintAt
  - Digikam::ImportCategorizedView, [2016](#)
  - Digikam::ItemCategorizedView, [2188](#)
  - ShowFoto::ShowfotoCategorizedView, [3654](#)
- hintAtModificationOfItems
  - Digikam::ScanController, [3012](#)
- hintAtMoveOrCopyOfAlbum
  - Digikam::ScanController, [3012](#)
- hintAtMoveOrCopyOfItems
  - Digikam::ScanController, [3013](#)
- HistogramPainter
  - Digikam::HistogramPainter, [1887](#)
- HistogramRenderingType
  - Digikam, [125](#)
- HistogramScale
  - Digikam, [126](#)
- HistogramWidget
  - Digikam::HistogramWidget, [1892](#)
- HistoryImageld
  - Digikam::HistoryImageld, [1895](#)
- historyImageld
  - Digikam::ItemInfo, [2300](#)
- HistoryLoadingFlag
  - Digikam::ItemHistoryGraph, [2275](#)
- hover
  - Digikam::ActionItemModel, [210](#)
- HoverColor
  - Digikam::SchemeManager, [3023](#)
- HS\_None
  - Digikam, [126](#)
- HSXGA
  - Digikam::VidSlideSettings, [3596](#)
- HttpProxy
  - Digikam::SystemSettings, [3280](#)
- HudSide
  - Digikam, [126](#)
- hue
  - Digikam::DColorValueSelector, [905](#)
  - Digikam::DHueSaturationSelector, [1041](#)
- humanReadableBytesCount
  - Digikam::ItemPropertiesTab, [2380](#)
- HUXGA
  - Digikam::VidSlideSettings, [3596](#)
- HVGA
  - Digikam::VidSlideSettings, [3595](#)
- HXGA
  - Digikam::VidSlideSettings, [3596](#)
- i18nDisplayName
  - Digikam::DImgFilterManager, [1101](#)
- iccData
  - Digikam::DRawInfo, [1392](#)
- IccManager
  - Digikam::IccManager, [1916](#)
- IccPostLoadingManager
  - Digikam::IccPostLoadingManager, [1921](#)
- IccProfile
  - Digikam::IccProfile, [1923](#)
- IccProfilesComboBox
  - Digikam::IccProfilesComboBox, [1927](#)
- icon
  - Digikam::DConfigDlgWdgItem, [953](#)
  - Digikam::DNotificationWidget, [1262](#)
  - Digikam::FaceRejectionOverlayButton, [1621](#)
  - Digikam::ImportRotateOverlayButton, [2105](#)
  - Digikam::ItemFullScreenOverlayButton, [2267](#)
  - Digikam::ItemRotateOverlayButton, [2395](#)
  - Digikam::ItemSelectionOverlayButton, [2408](#)

- Digikam::ItemViewHoverButton, [2457](#)
- iconTypeToIconName
  - Digikam::DConfigDlgTitle::Private, [936](#)
- id
  - Digikam::Album, [258](#)
  - Digikam::CollectionLocation, [609](#)
  - Digikam::Identity, [1950](#)
  - Digikam::ItemInfo, [2300](#)
  - Digikam::ItemScanner, [2399](#)
  - Digikam::ProgressItem, [2870](#)
  - Digikam::RGInfo, [2979](#)
  - Digikam::ThumbnailIdentifier, [3466](#)
  - Digikam::Token, [3511](#)
- identifier
  - Digikam::FilterAction, [1706](#)
- identifierForDetail
  - Digikam::ThumbnailCreator, [3463](#)
- Identity
  - Digikam::Identity, [1950](#)
- identityContains
  - Digikam::FacialRecognitionWrapper::Private, [1656](#)
- ids
  - Digikam::CollectionImageChangeset, [606](#)
  - Digikam::DbKeysCollection, [867](#)
- ifacelid
  - Digikam::DPlugin, [1301](#)
  - Digikam::DPluginBqm, [1309](#)
  - Digikam::DPluginDImg, [1325](#)
  - Digikam::DPluginEditor, [1329](#)
  - Digikam::DPluginGeneric, [1333](#)
  - Digikam::DPluginRawImport, [1342](#)
- ignoredCharacters
  - Digikam::DPlainTextEdit, [1295](#)
  - Digikam::DTextEdit, [1413](#)
- ignoredClicked
  - Digikam::AssignNameWidget, [407](#)
- ignoredWords
  - Digikam::LocalizeContainer, [2579](#)
- ignoreFaces
  - Digikam::DigikamItemView, [1064](#)
- IgnoreRootAlbum
  - Digikam::AbstractAlbumModel, [142](#)
- image
  - Digikam::BatchTool, [467](#)
  - Digikam::EmptyImageListProvider, [1505](#)
  - Digikam::MetaEnginePreviews, [2709](#)
  - Digikam::QListImageListProvider, [2890](#)
- image2Mat
  - Digikam, [129](#)
- image2Mat\_shared
  - Digikam, [129](#)
- ImageAlignment
  - Digikam::DConfigDlgTitle, [930](#)
- imageChange
  - Digikam::CoreDbWatch, [745](#)
  - Digikam::ItemModel, [2349](#)
- ImageChangeset
  - Digikam::ImageChangeset, [1953](#)
- ImageColorWorkSpace
  - Digikam::MetaEngine, [2673](#)
- imageComments
  - Digikam::ItemInfo, [2300](#)
- imageCommonContainer
  - Digikam::ItemInfo, [2301](#)
- imageCopyright
  - Digikam::ItemInfo, [2301](#)
- imageExtendedProperties
  - Digikam::ItemInfo, [2301](#)
- imageFilterModel
  - Digikam::ImageSortFilterModel, [1997](#)
  - Digikam::ItemFilterModel, [2246](#)
- imageHeight
  - Digikam::Canvas, [566](#)
- imageHistory
  - Digikam::ItemInfo, [2301](#)
- imageID
  - Digikam::DRawInfo, [1392](#)
- ImageIface
  - Digikam::ImageIface, [1969](#)
- imageInfo
  - Digikam::ItemModel, [2350](#)
- imageInformationRect
  - Digikam::ImportDelegate, [2037](#)
  - Digikam::ItemDelegate, [2213](#)
  - Digikam::ItemViewDelegate, [2453](#)
  - Digikam::ItemViewImportDelegate, [2463](#)
  - ShowFoto::ShowfotoDelegate, [3666](#)
  - ShowFoto::ShowfotoItemViewDelegate, [3708](#)
- imageInfosAboutToBeAdded
  - Digikam::ItemModel, [2350](#)
- imageInfosAboutToBeRemoved
  - Digikam::ItemModel, [2350](#)
- imageInfosAdded
  - Digikam::ItemFilterModel, [2246](#)
  - Digikam::ItemModel, [2350](#)
- imageInfosCleared
  - Digikam::ItemModel, [2351](#)
  - Digikam::ItemThumbnailModel, [2439](#)
- imageInfosRemoved
  - Digikam::ItemListModel, [2338](#)
  - Digikam::ItemModel, [2351](#)
- imageInfosSorted
  - Digikam::ImageSortFilterModel, [1997](#)
- ImageLeft
  - Digikam::DConfigDlgTitle, [931](#)
- imageLoaded
  - Digikam::LoadSaveThread, [2570](#)
- ImageMagickBackend
  - Digikam::MetaEngine, [2673](#)
- imageModel
  - Digikam::ItemHistoryGraphModel, [2285](#)
- imageModelIndex
  - Digikam::ItemHistoryGraphModel, [2285](#)
- ImageOrientation
  - Digikam::DRawInfo, [1389](#)
  - Digikam::MetaEngine, [2673](#)

- imagePosition
  - Digikam::ItemInfo, [2301](#)
- imageProcessed
  - Digikam::DuplicatesProgressObserver, [1432](#)
- imageProfile
  - Digikam::IccManager, [1917](#)
- ImageQualityParser
  - Digikam::ImageQualityParser, [1981](#)
- ImageQualitySorter
  - Digikam::ImageQualitySorter, [1986](#)
- ImageRight
  - Digikam::DConfigDlgTitle, [931](#)
- images
  - Digikam::EmptyImageListProvider, [1505](#)
  - Digikam::QListImageListProvider, [2890](#)
  - Digikam::RecognitionTrainingProvider, [2946](#)
  - Digikam::TrainingDataProvider, [3534](#)
- imageSaved
  - Digikam::LoadSaveThread, [2570](#)
- imageSavedAs
  - Digikam::DImg, [1083](#)
- ImageSelection
  - Digikam::Imagelface, [1969](#)
- ImageSelectionHistogram
  - Digikam, [126](#)
- imagesFromAlbumsAndTags
  - Digikam::HaarIface, [1878](#)
- imageSize
  - Digikam::DRawInfo, [1392](#)
  - Digikam::ImageZoomSettings, [2007](#)
- imageStartedLoading
  - Digikam::LoadSaveThread, [2570](#)
- imageStartedSaving
  - Digikam::LoadSaveThread, [2571](#)
- imagesUrls
  - Digikam::AlbumLabelsSearchHandler, [282](#)
- imageTagChange
  - Digikam::ItemModel, [2351](#)
- imageTagPair
  - Digikam::ItemInfo, [2301](#)
- imageWidth
  - Digikam::Canvas, [566](#)
- ImportContextMenuHelper
  - Digikam::ImportContextMenuHelper, [2023](#)
- importFilterModel
  - Digikam::ImportCategorizedView, [2017](#)
  - Digikam::ImportFilterModel, [2055](#)
  - Digikam::ImportSortFilterModel, [2111](#)
- ImportFilterModelPointerRole
  - Digikam::ImportFilterModel, [2054](#)
- ImportFilterModelRoles
  - Digikam::ImportFilterModel, [2053](#)
- ImportItemModelPointerRole
  - Digikam::ImportItemModel, [2068](#)
- ImportItemModelRoles
  - Digikam::ImportItemModel, [2068](#)
- ImportThumbnailDelegatePrivate
  - Digikam::ImportThumbnailDelegatePrivate, [2128](#)
- ImportThumbnailModel
  - Digikam::ImportThumbnailModel, [2132](#)
- importThumbnailModel
  - Digikam::ImportCategorizedView, [2017](#)
- Inactive
  - Digikam::FocusPoint, [1729](#)
- InactiveText
  - Digikam::SchemeManager, [3024](#)
- includeChildrenCount
  - Digikam::AbstractCountingAlbumModel, [181](#)
- IncludeFadingOut
  - Digikam::ItemVisibilityController, [2473](#)
- IncludeFadingOutMode
  - Digikam::ItemVisibilityController, [2472](#)
- IncludeLeadingSlash
  - Digikam::TagsCache, [3400](#)
- IncludeRootAlbum
  - Digikam::AbstractAlbumModel, [142](#)
- incrementedCounter
  - Digikam::DefaultVersionNamingScheme, [1000](#)
  - Digikam::VersionNamingScheme, [3580](#)
- indent
  - Digikam::DSelector, [1401](#)
- index
  - Digikam::DConfigDlgWdgModel, [960](#)
  - Digikam::TrackListModel, [3527](#)
- indexActivated
  - Digikam::ImportCategorizedView, [2017](#)
  - Digikam::ItemCategorizedView, [2188](#)
  - ShowFoto::ShowfotoCategorizedView, [3654](#)
- indexForAlbum
  - Digikam::AbstractAlbumModel, [144](#)
- indexForCamItemInfo
  - Digikam::ImportItemModel, [2069](#)
- indexForCategoryAt
  - Digikam::ItemViewCategorized, [2445](#)
- indexForInfo
  - Digikam::ItemHistoryGraphModel, [2285](#)
- indexForItemInfo
  - Digikam::ItemModel, [2351](#)
- indexForPath
  - Digikam::ItemModel, [2351](#)
- indexForShowfotoItemInfo
  - ShowFoto::ShowfotoItemModel, [3698](#)
- indexForUrl
  - Digikam::ImportItemModel, [2069](#)
  - ShowFoto::ShowfotoItemModel, [3698](#)
- indexFromImageId
  - Digikam::TableViewModel, [3325](#)
- indexImage
  - Digikam::HaarIface, [1878](#)
- indexVisuallyAt
  - Digikam::AbstractAlbumTreeView, [154](#)
- indicesEqual
  - Digikam::AbstractMarkerTiler, [192](#)
  - Digikam::GPSTiler, [1822](#)
  - Digikam::ItemMarkerTiler, [2341](#)
- InFocus

- Digikam::FocusPoint, [1729](#)
- infoForId
  - Digikam::ItemInfoCache, [2310](#)
- infoForPath
  - Digikam::ItemInfoCache, [2310](#)
- infoFromItem
  - Digikam::TableViewModel, [3325](#)
- infofance
  - Digikam::DigikamApp, [1045](#)
  - Digikam::DPluginBqm, [1309](#)
  - Digikam::DPluginEditor, [1330](#)
  - Digikam::DPluginGeneric, [1334](#)
  - Digikam::DXmlGuiWindow, [1447](#)
  - Digikam::ImageWindow, [2005](#)
  - Digikam::ImportUI, [2136](#)
  - Digikam::LightTableWindow, [2536](#)
  - Digikam::QueueMgrWindow, [2897](#)
  - ShowFoto::Showfoto, [3648](#)
- InfoMessage
  - Digikam::DConfigDlgTitle, [931](#)
- informationList
  - Digikam::CameraMessageBox, [550](#)
- infosLessThan
  - Digikam::ImportFilterModel, [2055](#)
  - Digikam::ItemFilterModel, [2246](#)
  - ShowFoto::ShowfotoFilterModel, [3677](#)
- init
  - Digikam::DConfigDlgMgr, [922](#)
  - Digikam::DPlainTextEdit::Private, [1297](#)
  - Digikam::DPluginLoader, [1337](#)
  - Digikam::DTextEdit::Private, [1415](#)
  - Digikam::EditorTool, [1474](#)
  - Digikam::lccTransform, [1943](#)
- initDbEngineErrorHandler
  - Digikam::CoreDbAccess, [723](#)
  - Digikam::SimilarityDbAccess, [3234](#)
- initExifTool
  - Digikam::ExifToolProcess, [1529](#)
- initFilter
  - Digikam::DImgThreadedFilter, [1119](#)
- initFrom
  - Digikam::HistogramPainter, [1888](#)
- initialCounter
  - Digikam::DefaultVersionNamingScheme, [1000](#)
  - Digikam::VersionNamingScheme, [3580](#)
- initialize
  - Digikam::CurvesContainer, [750](#)
- initializeExiv2
  - Digikam::MetaEngine, [2689](#)
- initializeGPSInfo
  - Digikam::MetaEngine, [2689](#)
- initializeNoThumbnailStorage
  - Digikam::ThumbnailLoadThread, [3482](#)
- initializeThumbnailDatabase
  - Digikam::ThumbnailLoadThread, [3482](#)
- initMaps
  - Digikam::DConfigDlgMgr, [922](#)
- initSchema
  - Digikam::CoreDbBackend, [728](#)
  - Digikam::FaceDbBackend, [1551](#)
  - Digikam::SimilarityDbBackend, [3239](#)
  - Digikam::ThumbsDbBackend, [3493](#)
- initSlave
  - Digikam::DImgThreadedFilter, [1119](#)
- Input
  - Digikam::lccProfile, [1923](#)
- InputColorSpace
  - Digikam::DRawDecoderSettings, [1376](#)
- inputColorSpace
  - Digikam::DRawDecoderSettings, [1379](#)
- inputProfile
  - Digikam::DRawDecoderSettings, [1380](#)
- inputProfiles
  - Digikam::lccSettings, [1938](#)
- insert
  - Digikam::KDNodeBase, [2481](#)
- insertFaceVector
  - Digikam::FaceDb, [1544](#)
- insertItem
  - Digikam::DExpanderBox, [1015](#)
- insertMessageOsdToFrame
  - Digikam::FrameOsd, [1739](#)
- insertOsdToFrame
  - Digikam::FrameOsd, [1739](#)
- insertPage
  - Digikam::DConfigDlg, [915](#)
  - Digikam::DConfigDlgWdg, [949](#)
  - Digikam::DConfigDlgWdgModel, [960](#), [961](#)
- insertSqueezedItem
  - Digikam::SqueezedComboBox, [3254](#)
- insertSqueezedList
  - Digikam::SqueezedComboBox, [3254](#)
- installLineEdit
  - Digikam::TreeViewLineEditComboBox, [3550](#)
- installQtTranslationFiles
  - Digikam, [130](#)
- installView
  - Digikam::AbstractAlbumTreeViewSelectComboBox, [162](#)
  - Digikam::AlbumSelectComboBox, [332](#)
  - Digikam::ChoiceSearchComboBox, [595](#)
  - Digikam::ListViewComboBox, [2543](#)
  - Digikam::StayPoppedUpComboBox, [3268](#)
  - Digikam::TreeViewComboBox, [3547](#)
  - Digikam::TreeViewLineEditComboBox, [3550](#)
- INSTANCE
  - Digikam::StateSavingObject, [3261](#)
- instance
  - Digikam::AlbumManager, [298](#)
  - Digikam::AlbumThumbnailLoader, [358](#)
  - Digikam::DatabaseServerStarter, [789](#)
  - Digikam::DBJobsManager, [860](#)
  - Digikam::DMetadataSettings, [1190](#)
  - Digikam::DNNModelManager, [1234](#)
  - Digikam::DPluginLoader, [1338](#)
  - Digikam::GeolocationSettings, [1778](#)

- Digikam::IccSettings, [1938](#)
- Digikam::IOJobsManager, [2157](#)
- Digikam::ItemSortCollator, [2413](#)
- Digikam::LocalizeSettings, [2583](#)
- Digikam::MetaEngineSettings, [2713](#)
- Digikam::NetworkManager, [2758](#)
- Digikam::ProgressManager, [2879](#)
- instructions
  - Digikam::ItemCopyright, [2206](#)
- integrityCheck
  - Digikam::CoreDB, [708](#)
  - Digikam::FaceDb, [1544](#)
  - Digikam::FacialRecognitionWrapper, [1654](#)
  - Digikam::IdentityProvider, [1952](#)
  - Digikam::SimilarityDb, [3231](#)
  - Digikam::ThumbsDb, [3488](#)
- intellectualGenre
  - Digikam::ItemExtendedProperties, [2229](#)
- interface
  - Digikam::Canvas, [566](#)
- Intermediate
  - Digikam::HistoryImageId, [1895](#)
- intermediateDirectory
  - Digikam::DefaultVersionNamingScheme, [1000](#)
- intermediateFileName
  - Digikam::DefaultVersionNamingScheme, [1001](#)
  - Digikam::VersionNamingScheme, [3581](#)
- internalServerMysqlInitCmd
  - Digikam::DbEngineParameters, [846](#)
- INTERNATIONAL\_1924
  - Digikam::Ellipsoid, [1494](#)
- intersectionSet
  - Digikam::DCategorizedView::Private, [888](#)
- intersects
  - Digikam::TagRegion, [3394](#)
- invalidate
  - Digikam::ItemInfoCache, [2310](#)
- invalidatePaintingCache
  - Digikam::ImportDelegate, [2037](#)
  - Digikam::ItemDelegate, [2213](#)
  - Digikam::ItemViewImportDelegate, [2463](#)
- InvalidBehavior
  - Digikam::IccSettingsContainer, [1941](#)
- InvalidType
  - Digikam::DPluginAction, [1304](#)
  - Digikam::IccProfile, [1923](#)
- inverseFlattening
  - Digikam::Ellipsoid, [1494](#)
- invertSelection
  - Digikam::TableView, [3284](#)
- ioFileSettings
  - Digikam::BatchTool, [467](#)
- iptcCorePropertyName
  - Digikam::ItemScanner, [2399](#)
- isAccessible
  - Digikam::ThumbnailInfo, [3471](#)
- isAlbumRoot
  - Digikam::CollectionManager, [615](#)
- isAlbumUrl
  - Digikam::CoreDbUrl, [742](#)
- isAncestorOf
  - Digikam::Album, [258](#)
- isAnimatedImage
  - Digikam::DImg, [1084](#)
- IsAppendRole
  - Digikam::SetupCollectionModel, [3171](#)
- isAssigned
  - Digikam::ItemTagPair, [2418](#)
- isCacheable
  - Digikam::LoadingCache, [2545](#)
- isCanceled
  - Digikam::DuplicatesProgressObserver, [1432](#)
  - Digikam::IOJobsThread, [2162](#)
- isCancelled
  - Digikam::BatchTool, [467](#)
- isCategorizedModel
  - Digikam::DCategorizedSortFilterProxyModel, [879](#)
- IsCategoryRole
  - Digikam::SetupCollectionModel, [3171](#)
- isCheckedable
  - Digikam::DConfigDlgWdgItem, [953](#)
  - Digikam::LabelsTreeView, [2500](#)
- isChecked
  - Digikam::DConfigDlgWdgItem, [954](#)
- isClearButtonEnabled
  - Digikam::DPlainTextEdit, [1295](#)
  - Digikam::DTextEdit, [1413](#)
- isCloseButtonVisible
  - Digikam::DNotificationWidget, [1262](#)
- isCompatible
  - Digikam::BdEngineBackend, [486](#)
- isComplexAction
  - Digikam::FilterActionFilter, [1712](#)
- isConnected
  - Digikam::GPSModelIndexProxyMapper, [1827](#)
- isDecodable
  - Digikam::DRawInfo, [1393](#)
- isDefault
  - Digikam::DConfigDlgMngr, [922](#)
- IsDeleteRole
  - Digikam::SetupCollectionModel, [3171](#)
- isEmpty
  - Digikam::ActionThreadBase, [220](#)
  - Digikam::CurvesContainer, [750](#)
  - Digikam::DImageHistory, [1068](#)
  - Digikam::DTrashItemModel, [1424](#)
  - Digikam::ItemPosition, [2355](#)
  - Digikam::MetaEngine, [2689](#)
  - Digikam::ProgressManager, [2879](#)
  - Digikam::Template, [3440](#)
- isEnabled
  - Digikam::DConfigDlgWdgItem, [954](#)
  - Digikam::IccSettings, [1938](#)
- isExpanded
  - Digikam::Sidebar, [3218](#)
- isFaceTagModel

- Digikam::AbstractAlbumModel, [144](#)
- isFieldElement
  - Digikam::SearchXmlReader, [3149](#)
- isFiltering
  - Digikam::AlbumFilterModel, [270](#)
  - Digikam::CheckableAlbumFilterModel, [590](#)
  - Digikam::SearchFilterModel, [3101](#)
  - Digikam::TagPropertiesFilterModel, [3391](#)
- isGrouped
  - Digikam::ItemInfo, [2301](#)
- isGroupElement
  - Digikam::SearchXmlReader, [3149](#)
- isGroupOpen
  - Digikam::ItemFilterModel, [2246](#)
- isHideAnimationRunning
  - Digikam::DNotificationWidget, [1262](#)
- isInitialized
  - Digikam::SimilarityDbAccess, [3234](#)
- isInSingleFileMode
  - Digikam::StackedView, [3257](#)
- isInternalTag
  - Digikam::TagsCache, [3402](#)
- isInTransaction
  - Digikam::BdEngineBackend, [487](#)
- isIvfDefinitive
  - Digikam::Ellipsoid, [1494](#)
- isLinear
  - Digikam::ImageCurves, [1956](#)
- isLoadingState
  - Digikam::LabelsTreeView, [2500](#)
- isLocationAvailable
  - Digikam::ItemInfo, [2302](#)
- isMigrationChecked
  - Digikam::MigrateFromDigikam4Page, [2717](#)
- isMinimized
  - Digikam::Sidebar::Private, [3220](#)
- isMovingAlbum
  - Digikam::AlbumManager, [298](#)
- isNoTransform
  - Digikam::MetaEngineRotation, [2711](#)
- isNull
  - Digikam::CamItemInfo, [557](#)
  - Digikam::DeltaTime, [1007](#)
  - Digikam::DImageHistory, [1068](#)
  - Digikam::ItemInfo, [2302](#)
  - Digikam::Template, [3440](#)
  - ShowFoto::ShowfotoItemInfo, [3694](#)
- isOpen
  - Digikam::IccProfile, [1924](#)
- isPerson
  - Digikam::FaceTags, [1636](#)
- isPreviewImage
  - Digikam::LoadingDescription, [2554](#)
- isRawConversion
  - Digikam::DImgFilterManager, [1101](#)
- isRawFile
  - Digikam::BatchTool, [467](#)
- isReadableImageFile
  - Digikam, [130](#)
- isReadOnly
  - Digikam::Canvas, [566](#)
  - Digikam::DDateEdit, [969](#)
  - Digikam::DImg, [1084](#)
- isReady
  - Digikam::BackendGoogleMaps, [443](#)
  - Digikam::BackendMarble, [452](#)
- isReducedVersion
  - Digikam::LoadingDescription, [2554](#)
- isRefreshing
  - Digikam::ImportItemModel, [2069](#)
  - Digikam::ItemModel, [2351](#)
- isRemoved
  - Digikam::ItemInfo, [2302](#)
- isReproducible
  - Digikam::FilterActionFilter, [1712](#)
- isRestoreCheckState
  - Digikam::AbstractCheckableAlbumTreeView, [175](#)
- isRestoringSelectionFromHistory
  - Digikam::AlbumLabelsSearchHandler, [282](#)
- isRoot
  - Digikam::Album, [258](#)
- isRunning
  - Digikam::DatabaseServer, [788](#)
  - Digikam::DOnlineTranslator, [1273](#)
- isRunningInApplImageBundle
  - Digikam, [130](#)
- isRunningOnNativeKDE
  - Digikam, [130](#)
- isSameProfileAs
  - Digikam::IccProfile, [1924](#)
- isShowAnimationRunning
  - Digikam::DNotificationWidget, [1262](#)
- isSixteenBit
  - Digikam::ImageHistogram, [1967](#)
- isSourceTranscriptionEnabled
  - Digikam::DOnlineTranslator, [1274](#)
- isSourceTranslitEnabled
  - Digikam::DOnlineTranslator, [1274](#)
- isSphere
  - Digikam::Ellipsoid, [1494](#)
- isSRGB
  - Digikam::IccManager, [1917](#)
- isStoredLosslessly
  - Digikam::CurvesContainer, [750](#)
- isSupported
  - Digikam::DImgBuiltinFilter, [1091](#)
  - Digikam::DImgFilterGenerator, [1098](#)
  - Digikam::DImgFilterManager, [1101](#)
  - Digikam::FilterActionFilter, [1712](#)
- isSupportTranslation
  - Digikam::DOnlineTranslator, [1274](#)
- isTabRaised
  - Digikam::DMultiTabBar, [1200](#)
- isTagListDirty
  - Digikam::GPSItemContainer, [1805](#)
- isTemporarySearch

- Digikam::SAlbum, [3005](#)
- isThumbnail
  - Digikam::LoadingDescription, [2554](#)
- isTranslationOptionsEnabled
  - Digikam::DOnlineTranslator, [1274](#)
- isTranslationTranslitEnabled
  - Digikam::DOnlineTranslator, [1275](#)
- isTrashAlbum
  - Digikam::Album, [259](#)
- IsUpdateRole
  - Digikam::SetupCollectionModel, [3171](#)
- isUsedByLabelsTree
  - Digikam::Album, [259](#)
- isUtf8
  - Digikam::MetaEngine::Private, [2705](#)
- isValid
  - Digikam::DbEngineParameters, [845](#)
  - Digikam::DImageHistory, [1068](#)
  - Digikam::DImgBuiltinFilter, [1092](#)
  - Digikam::Rule, [2996](#)
- isValidId
  - Digikam::IdentityProvider, [1953](#)
- isValue
  - Digikam::DbEngineActionType, [835](#)
- isVisible
  - Digikam::ItemInfo, [2302](#)
  - Digikam::SearchField, [3038](#)
  - Digikam::VisibilityController, [3603](#)
- item
  - Digikam::DConfigDlgWdgModel, [961](#)
  - Digikam::GraphicsDImgView, [1856](#)
  - Digikam::SqueezedComboBox, [3255](#)
- itemChanged
  - Digikam::ItemPropertiesSideBarDB, [2376](#)
- ItemComments
  - Digikam::ItemComments, [2194](#), [2195](#)
- itemCoordinates
  - Digikam::GeoModelHelper, [1781](#)
  - Digikam::GPSBookmarkModelHelper, [1791](#)
  - Digikam::GPSGeofaceModelHelper, [1802](#)
  - Digikam::ItemGPSModelHelper, [2273](#)
  - Digikam::MapViewModelHelper, [2617](#)
- ItemCopyMoveHint
  - Digikam::ItemCopyMoveHint, [2203](#)
- ItemCopyright
  - Digikam::ItemCopyright, [2205](#)
- itemCount
  - Digikam::FindDuplicatesAlbumItem, [1721](#)
- itemDelegate
  - Digikam::DConfigDlgView, [939](#)
- ItemDelegateOverlayContainer
  - Digikam::ItemDelegateOverlayContainer, [2221](#)
- ItemExtendedProperties
  - Digikam::ItemExtendedProperties, [2229](#)
- ItemFilterModelRoles
  - Digikam::ItemFilterModel, [2243](#)
- itemFlags
  - Digikam::GPSBookmarkModelHelper, [1791](#)
- itemForAction
  - Digikam::ActionItemModel, [210](#)
- itemHighlighted
  - Digikam::SqueezedComboBox, [3255](#)
- itemIcon
  - Digikam::GeoModelHelper, [1782](#)
  - Digikam::GPSBookmarkModelHelper, [1792](#)
- ItemInfo
  - Digikam::ItemInfo, [2295](#)
- itemInfo
  - Digikam::DBInfoface, [855](#)
  - Digikam::DMetaInfoface, [1195](#)
- itemInfosAboutToBeAdded
  - Digikam::ImportItemModel, [2069](#)
  - ShowFoto::ShowfotoItemModel, [3698](#)
- itemInfosAboutToBeRemoved
  - Digikam::ImportItemModel, [2069](#)
  - ShowFoto::ShowfotoItemModel, [3699](#)
- itemInfosAdded
  - Digikam::ImportItemModel, [2070](#)
  - ShowFoto::ShowfotoItemModel, [3699](#)
- itemInfosRemoved
  - Digikam::ImportItemModel, [2070](#)
  - ShowFoto::ShowfotoItemModel, [3699](#)
- ItemListType
  - Digikam::QueueListView, [2893](#)
- ItemModelPointerRole
  - Digikam::ItemModel, [2348](#)
- ItemModelRoles
  - Digikam::ItemModel, [2348](#)
- ItemModified
  - Digikam::ItemChangeHint, [2192](#)
- ItemOrderRole
  - Digikam::CategorizedItemModel, [577](#)
- ItemPosition
  - Digikam::ItemPosition, [2354](#)
- ItemQueryPostHook
  - Digikam::ItemQueryPostHook, [2384](#)
- ItemRescan
  - Digikam::ItemChangeHint, [2192](#)
- items
  - Digikam::DDatePickerPopup, [978](#)
  - Digikam::ItemVisibilityController, [2474](#)
- itemScanInfo
  - Digikam::ItemScanner, [2400](#)
- ItemScanner
  - Digikam::ItemScanner, [2397](#), [2398](#)
- itemsVisible
  - Digikam::DPluginConfView, [1311](#)
- itemsWithVisiblyProperty
  - Digikam::DPluginConfView, [1311](#)
- ItemTagPair
  - Digikam::ItemTagPair, [2417](#)
- ItemThumbnailModel
  - Digikam::ItemThumbnailModel, [2439](#)
- itemView
  - Digikam::DWItemDelegate, [1436](#)
- ItemVisibilityControllerPropertyObject

- Digikam::ItemVisibilityControllerPropertyObject, [2477](#)
- jobData
  - Digikam::IOJobsThread, [2162](#)
- jobId
  - Digikam::ItemExtendedProperties, [2229](#)
- JPEGPreview
  - Digikam::DNGWriter, [1213](#)
- JpegRotator
  - Digikam::JPEGUtils::JpegRotator, [2478](#)
- JPEGSUBSampling
  - Digikam::IOFileSettings, [2154](#)
- KDTreeBase
  - Digikam::KDTreeBase, [2487](#)
- KeepProfile
  - Digikam::ICCSettingsContainer, [1941](#)
- KeepSignals
  - Digikam::WorkerObject, [3617](#)
- keyPressed
  - Digikam::ItemViewCategorized, [2445](#)
- keyPressEvent
  - Digikam::DPopupFrame, [1351](#)
  - Digikam::PanIconFrame, [2815](#)
- keywords
  - Digikam::DisjointMetadata, [1140](#)
- keywordSearch
  - Digikam::SearchXmlWriter, [3153](#)
- label
  - Digikam::CollectionLocation, [610](#)
  - Digikam::ProgressItem, [2870](#)
- labelClicked
  - Digikam::AssignNameWidget, [407](#)
- language
  - Digikam::DOnlineTranslator, [1275](#)
  - Digikam::ItemComments, [2196](#)
- LanguageChoiceBehavior
  - Digikam::ItemComments, [2194](#)
- languageCode
  - Digikam::DOnlineTranslator, [1275](#)
- languageName
  - Digikam::DOnlineTranslator, [1276](#)
- languageNameRFC3066
  - Digikam::AltLangStrEdit, [371](#)
- lastCheckDate
  - Digikam::OnlineVersionChecker, [2795](#)
- lastChild
  - Digikam::Album, [259](#)
- lastDescriptions
  - Digikam::ThumbnailLoadThread, [3482](#)
- lastError
  - Digikam::BdEngineBackend, [487](#)
  - Digikam::CoreDbAccess, [723](#)
- lastSavedFilePath
  - Digikam::DImg, [1084](#)
- lastSelectedItemUrl
  - Digikam::TrashView, [3543](#)
- lastSQLException
  - Digikam::BdEngineBackend, [487](#)
- latitude
  - Digikam::DRawInfo, [1393](#)
  - Digikam::ItemPosition, [2355](#)
- latitudeFormatted
  - Digikam::ItemPosition, [2355](#)
- latitudeNumber
  - Digikam::ItemPosition, [2355](#)
- latitudeUserPresentableNumbers
  - Digikam::ItemPosition, [2356](#)
- layoutMargin
  - Digikam, [130](#)
- layoutSpacing
  - Digikam, [130](#)
- LcmsLock
  - Digikam::LcmsLock, [2502](#)
- LeadingSlashPolicy
  - Digikam::TagsCache, [3400](#)
- leafImages
  - Digikam::ItemHistoryGraph, [2277](#)
- LeaveFileUntagged
  - Digikam::ICCSettingsContainer, [1941](#)
- leaves
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1842](#)
- leftCharacters
  - Digikam::DPlainTextEdit, [1296](#)
  - Digikam::DTextEdit, [1413](#)
- leftMargin
  - Digikam::DCategoryDrawer, [893](#)
  - Digikam::DRawInfo, [1393](#)
- lensDescription
  - Digikam::LensFunface, [2513](#)
- lensModel
  - Digikam::DRawInfo, [1393](#)
- lessThan
  - Digikam::AlbumFilterModel, [270](#)
  - Digikam::CamItemSortSettings, [560](#)
  - Digikam::DCategorizedSortFilterProxyModel, [879](#)
  - Digikam::ItemSortSettings, [2415](#), [2416](#)
  - ShowFoto::ShowfotoItemSortSettings, [3703](#)
- lessThanByOrder
  - Digikam::CamItemSortSettings, [560](#)
  - Digikam::ItemSortSettings, [2416](#)
  - ShowFoto::ShowfotoItemSortSettings, [3704](#)
- levelsChannelReset
  - Digikam::ImageLevels, [1973](#)
- LibHeifBackend
  - Digikam::MetaEngine, [2673](#)
- LibRawBackend
  - Digikam::MetaEngine, [2673](#)
- librawUseGomp
  - Digikam::DRawDecoder, [1370](#)
- librawVersion
  - Digikam::DRawDecoder, [1370](#)
- LibsInfoDlg
  - Digikam::LibsInfoDlg, [2521](#)



- lift
  - Digikam::CoreDbOperationGroup, [734](#)
  - Digikam::FaceDbOperationGroup, [1552](#)
- LightShade
  - Digikam::SchemeManager, [3024](#)
- linkActivated
  - Digikam::DNotificationWidget, [1263](#)
- LinkBackground
  - Digikam::SchemeManager, [3022](#)
- linkHovered
  - Digikam::DNotificationWidget, [1263](#)
- LinkText
  - Digikam::SchemeManager, [3024](#)
- LinScale
  - Digikam::TimeLineWidget, [3508](#)
- LinScaleHistogram
  - Digikam, [126](#)
- list
  - Digikam::ItemLister, [2321](#)
- ListAFPoints
  - Digikam::FocusPointsExtractor, [1738](#)
- listAreaRange
  - Digikam::ItemLister, [2321](#)
- listDateRange
  - Digikam::ItemLister, [2321](#)
- listDTrashItems
  - Digikam::IOJobsThread, [2162](#)
- listHaarSearch
  - Digikam::ItemLister, [2322](#)
- listImageTagPropertySearch
  - Digikam::ItemLister, [2322](#)
- listPAlbum
  - Digikam::ItemLister, [2322](#)
- listPath
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1842](#)
- listSearch
  - Digikam::ItemLister, [2322](#)
- listTag
  - Digikam::ItemLister, [2323](#)
- ListViewComboBox
  - Digikam::ListViewComboBox, [2542](#)
- load
  - Digikam::DMetadata, [1186](#)
  - Digikam::ExifToolParser, [1522](#)
  - Digikam::LoadSaveThread, [2571](#)
  - Digikam::ManagedLoadSaveThread, [2609](#)
  - Digikam::MetadataHub, [2635](#)
  - Digikam::MetaEngine, [2690](#)
  - Digikam::PreviewLoadThread, [2858](#)
  - Digikam::ThumbnailCreator, [3463](#)
  - Digikam::ThumbnailLoadThread, [3482](#)
  - Digikam::WorkflowManager, [3624](#)
- LOAD\_CHUNKS
  - Digikam::ExifToolProcess, [1527](#)
- LOAD\_METADATA
  - Digikam::ExifToolProcess, [1527](#)
- LoadAll
  - Digikam::DImgLoader, [1103](#)
- loadAllProfilesProperties
  - Digikam::IccSettings, [1938](#)
- loadChunk
  - Digikam::ExifToolParser, [1523](#)
- loadColumnProfile
  - Digikam::TableViewModel, [3325](#)
- loadDetail
  - Digikam::ThumbnailCreator, [3463](#)
- loadEcmQtTranslationFiles
  - Digikam, [130](#)
- loadEmbeddedPreview
  - Digikam::DRawDecoder, [1370](#), [1371](#)
- Loader
  - Digikam::MLPipelineFoundation, [2729](#)
- loader
  - Digikam::DPluginDImg, [1325](#)
  - Digikam::FacePipelineDetect, [1579](#)
  - Digikam::FacePipelineDetectRecognize, [1583](#)
  - Digikam::FacePipelineEdit, [1588](#)
  - Digikam::FacePipelineRecognize, [1601](#)
  - Digikam::FacePipelineReset, [1605](#)
  - Digikam::FacePipelineRetrain, [1609](#)
- loaderName
  - Digikam::DPluginDImg, [1326](#)
- loadFast
  - Digikam::PreviewLoadThread, [2858](#)
- loadFastButLarge
  - Digikam::PreviewLoadThread, [2859](#)
- loadFastSynchronously
  - Digikam::PreviewLoadThread, [2859](#)
- LoadFlag
  - Digikam::DImgLoader, [1103](#)
- loadFromData
  - Digikam::MetaEngine, [2690](#)
- loadFromDataAndMerge
  - Digikam::MetaEngine, [2690](#)
- loadFromDisk
  - Digikam::ItemScanner, [2400](#)
- loadFromSidecarAndMerge
  - Digikam::MetaEngine, [2690](#)
- loadFromURL
  - Digikam::ExifWidget, [1537](#)
  - Digikam::ICCPProfileWidget, [1935](#)
  - Digikam::IptcWidget, [2167](#)
  - Digikam::MakerNoteWidget, [2603](#)
  - Digikam::XmpWidget, [3639](#)
- loadFullImage
  - Digikam::DRawDecoder, [1371](#)
- loadHalfPreview
  - Digikam::DRawDecoder, [1371](#), [1372](#)
- loadHighQuality
  - Digikam::PreviewLoadThread, [2859](#)
- LoadICCData
  - Digikam::DImgLoader, [1103](#)
- LoadImageData
  - Digikam::DImgLoader, [1103](#)
- loadImageData

- Digikam::GPSItemContainer, [1805](#)
- Digikam::ItemGPS, [2271](#)
- LoadImageHistory
  - Digikam::DImgLoader, [1103](#)
- loadImagesFromCurrentAlbum
  - Digikam::DItemsList, [1159](#)
- LoadingDescription
  - Digikam::LoadingDescription, [2553](#)
- LoadingMode
  - Digikam::ManagedLoadSaveThread, [2607](#)
- LoadingModeNormal
  - Digikam::ManagedLoadSaveThread, [2607](#)
- LoadingModeShared
  - Digikam::ManagedLoadSaveThread, [2607](#)
- LoadingPolicy
  - Digikam::ManagedLoadSaveThread, [2607](#)
- LoadingPolicyAppend
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadingPolicyFirstRemovePrevious
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadingPolicyPreload
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadingPolicyPrepend
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadingPolicySimpleAppend
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadingPolicySimplePrepend
  - Digikam::ManagedLoadSaveThread, [2608](#)
- loadingProgress
  - Digikam::LoadSaveThread, [2571](#)
- LoadingTaskFilter
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadingTaskFilterAll
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadingTaskFilterPreloading
  - Digikam::ManagedLoadSaveThread, [2608](#)
- LoadItemInfo
  - Digikam::DImgLoader, [1103](#)
- loadItemInfo
  - Digikam::DImg, [1084](#)
- loadItemInfos
  - Digikam::LightTableWindow, [2536](#)
- loadItemsForCollection
  - Digikam::DTrashItemModel, [1424](#)
- LoadLeavesHistory
  - Digikam::ItemHistoryGraph, [2275](#)
- LoadMetadata
  - Digikam::DImgLoader, [1103](#)
- loadModels
  - Digikam::DNNFaceExtractorBase, [1228](#)
  - Digikam::DNNOpenFaceExtractor, [1240](#)
  - Digikam::DNNResnetDetector, [1242](#)
  - Digikam::DNNFaceExtractor, [1244](#)
  - Digikam::DNNYoloDetector, [1246](#)
  - Digikam::OpenfacePreprocessor, [2803](#)
- loadPlugins
  - Digikam::DPluginConfViewBqm, [1314](#)
  - Digikam::DPluginConfViewDImg, [1316](#)
  - Digikam::DPluginConfViewEditor, [1318](#)
  - Digikam::DPluginConfViewGeneric, [1320](#)
  - Digikam::DPluginLoader::Private, [1339](#)
- LoadPreview
  - Digikam::DImgLoader, [1103](#)
- loadQImage
  - Digikam::Haarface, [1878](#)
- loadRawPreview
  - Digikam::DRawDecoder, [1372](#)
- LoadRelationCloud
  - Digikam::ItemHistoryGraph, [2275](#)
- loadSaveNotifier
  - Digikam::SharedLoadingTask, [3194](#)
- loadSettings
  - Digikam::TableViewColumnProfile, [3290](#)
- loadState
  - Digikam::AlbumSelectors, [341](#)
  - Digikam::StateSavingObject, [3262](#)
- loadStdQtTranslationFiles
  - Digikam, [131](#)
- LoadSubjectHistory
  - Digikam::ItemHistoryGraph, [2275](#)
- loadTags
  - Digikam::MetadataHub, [2635](#)
- loadToDImg
  - Digikam::BatchTool, [467](#)
- loadTrainingData
  - Digikam::FaceClassifier, [1541](#)
- LoadUniqueHash
  - Digikam::DImgLoader, [1103](#)
- loadUsingFFmpeg
  - Digikam::DMetadata, [1186](#)
- loadUsingRawEngine
  - Digikam::DMetadata, [1187](#)
- localFileRename
  - Digikam::DFileOperations, [1021](#)
- localizedCameraModel
  - Digikam::DRawInfo, [1393](#)
- location
  - Digikam::ItemExtendedProperties, [2229](#)
- LocationAllRight
  - Digikam::CollectionManager, [613](#)
- LocationAvailable
  - Digikam::CollectionLocation, [609](#)
- LocationCheckResult
  - Digikam::CollectionManager, [613](#)
- LocationDeleted
  - Digikam::CollectionLocation, [609](#)
- locationForAlbumRoot
  - Digikam::CollectionManager, [616](#)
- locationForAlbumRootId
  - Digikam::CollectionManager, [616](#)
- locationForUrl
  - Digikam::CollectionManager, [616](#)
- LocationHasProblems
  - Digikam::CollectionManager, [613](#)
- LocationHidden
  - Digikam::CollectionLocation, [609](#)

- LocationInvalidCheck
  - Digikam::CollectionManager, [613](#)
- LocationNotAllowed
  - Digikam::CollectionManager, [613](#)
- LocationNull
  - Digikam::CollectionLocation, [609](#)
- locationPropertiesChanged
  - Digikam::CollectionManager, [616](#)
- locationStatusChanged
  - Digikam::CollectionManager, [616](#)
- LocationUnavailable
  - Digikam::CollectionLocation, [609](#)
- LogScale
  - Digikam::TimeLineWidget, [3508](#)
- LogScaleHistogram
  - Digikam, [126](#)
- longestPath
  - Digikam::Graph< VertexProperties, EdgeProperties >::Path, [1850](#)
- longestPathTouching
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1842](#)
- longitudeNumber
  - Digikam::ItemInfo, [2302](#)
- lookupCacheKeys
  - Digikam::LoadingDescription, [2554](#)
- LTLeftPanelRole
  - Digikam::ItemModel, [2348](#)
- LTRightPanelRole
  - Digikam::ItemModel, [2348](#)
- m\_A
  - Digikam::GeodeticCalculator, [1767](#)
- m\_authors
  - Digikam::Template, [3440](#)
- m\_authorsPosition
  - Digikam::Template, [3440](#)
- m\_bin
  - Digikam::Haar::WeightBin, [1875](#)
- m\_cancel
  - Digikam::ActionJob, [212](#)
  - Digikam::DRawDecoder, [1374](#)
- m\_contactInfo
  - Digikam::Template, [3440](#)
- m\_copyright
  - Digikam::Template, [3440](#)
- m\_credit
  - Digikam::Template, [3441](#)
- m\_decoderSettings
  - Digikam::DRawDecoder, [1374](#)
- m\_destImage
  - Digikam::DImgThreadedFilter, [1121](#)
- m\_destinationValid
  - Digikam::GeodeticCalculator, [1767](#)
- m\_directionValid
  - Digikam::GeodeticCalculator, [1767](#)
- m\_distance
  - Digikam::GeodeticCalculator, [1768](#)
- m\_eccentricitySquared
  - Digikam::GeodeticCalculator, [1768](#)
- m\_ellipsoid
  - Digikam::GeodeticCalculator, [1768](#)
- m\_instructions
  - Digikam::Template, [3441](#)
- m\_inverseFlattening
  - Digikam::Ellipsoid, [1496](#)
- m\_ivfDefinitive
  - Digikam::Ellipsoid, [1496](#)
- m\_lat1
  - Digikam::GeodeticCalculator, [1768](#)
- m\_lat2
  - Digikam::GeodeticCalculator, [1768](#)
- m\_locationInfo
  - Digikam::Template, [3441](#)
- m\_master
  - Digikam::DImgThreadedFilter, [1121](#)
- m\_maxOrthodromicDistance
  - Digikam::GeodeticCalculator, [1768](#)
- m\_name
  - Digikam::DImgThreadedFilter, [1121](#)
- m\_orgImage
  - Digikam::DImgThreadedFilter, [1121](#)
- m\_originalUUID
  - Digikam::HistoryImageId, [1895](#)
- m\_progressBegin
  - Digikam::DImgThreadedFilter, [1122](#)
- m\_rightUsageTerms
  - Digikam::Template, [3441](#)
- m\_semiMajorAxis
  - Digikam::Ellipsoid, [1497](#)
  - Digikam::GeodeticCalculator, [1768](#)
- m\_semiMinorAxis
  - Digikam::Ellipsoid, [1497](#)
  - Digikam::GeodeticCalculator, [1769](#)
- m\_settingsWidget
  - Digikam::BatchTool, [472](#)
- m\_slave
  - Digikam::DImgThreadedFilter, [1122](#)
- m\_source
  - Digikam::Template, [3441](#)
- m\_subjects
  - Digikam::Template, [3441](#)
- m\_templateTitle
  - Digikam::Template, [3441](#)
- m\_timer
  - Digikam::ActionJob, [212](#)
- m\_TOLERANCE\_0
  - Digikam::GeodeticCalculator, [1769](#)
- m\_TOLERANCE\_CHECK
  - Digikam::GeodeticCalculator, [1769](#)
- m\_uuid
  - Digikam::HistoryImageId, [1895](#)
- macOSBundlePrefix
  - Digikam, [131](#)
- mainMarbleWidget
  - Digikam::GeolocationSettings, [1778](#)
- make

- Digikam::DRawInfo, [1393](#)
- makeColumnVisible
  - Digikam::DFontProperties, [1030](#)
- makeDescription
  - Digikam::LensFunface, [2513](#)
- makeQMapFromXML
  - Digikam::BackendGeonamesRG, [436](#)
  - Digikam::BackendGeonamesUSRG, [439](#)
  - Digikam::BackendOsmRG, [458](#)
- makeStaleAlbum
  - Digikam::CoreDB, [708](#)
- ManagedLoadSaveThread
  - Digikam::ManagedLoadSaveThread, [2609](#)
- manualOrder
  - Digikam::ItemInfo, [2302](#)
- mapImagesUrl
  - Digikam::CoreDbUrl, [742](#)
- mapImageToZoom
  - Digikam::ImageZoomSettings, [2008](#)
- mapIndexForDragDrop
  - Digikam::DragDropViewImplementation, [1365](#)
  - Digikam::ItemViewCategorized, [2446](#)
  - Digikam::TableViewTreeView, [3329](#)
  - Digikam::VersionsTreeView, [3586](#)
- mapLeftToRight
  - Digikam::GPSModelIndexProxyMapper, [1826](#)
- mapListToSource
  - Digikam::ImageSortFilterModel, [1997](#)
- mapRightToLeft
  - Digikam::GPSModelIndexProxyMapper, [1826](#)
- mapSelectionLeftToRight
  - Digikam::GPSModelIndexProxyMapper, [1826](#)
- mapSelectionRightToLeft
  - Digikam::GPSModelIndexProxyMapper, [1826](#)
- mapSize
  - Digikam::BackendGoogleMaps, [444](#)
  - Digikam::BackendMarble, [452](#)
- mapToOriginalSize
  - Digikam::TagRegion, [3395](#)
- mapToSourceImportModel
  - Digikam::ImportSortFilterModel, [2111](#)
- mapToSourceShowfotoModel
  - ShowFoto::ShowfotoSortFilterModel, [3732](#)
- mapWidget
  - Digikam::BackendGoogleMaps, [444](#)
  - Digikam::BackendMarble, [452](#)
  - Digikam::MapBackend, [2612](#)
- mapWidgetDocked
  - Digikam::BackendGoogleMaps, [444](#)
  - Digikam::BackendMarble, [452](#)
- MapWidgetView
  - Digikam::MapWidgetView, [2628](#)
- mapZoomToImage
  - Digikam::ImageZoomSettings, [2008](#)
- marbleCustomPaint
  - Digikam::BackendMarble, [452](#)
- markAllAsIgnored
  - Digikam::FaceGroup, [1557](#)
- markAsScanned
  - Digikam::FaceUtils, [1650](#)
- markDerivedFrom
  - Digikam::ItemInfo, [2302](#)
- mat2Image
  - Digikam, [131](#)
- mat2Image\_shared
  - Digikam, [131](#)
- MatchContainingFragment
  - Digikam::TaggingActionFactory, [3360](#)
- matches
  - Digikam::AlbumFilterModel, [270](#)
  - Digikam::CheckableAlbumFilterModel, [590](#)
  - Digikam::CoreDbNameFilter, [734](#)
  - Digikam::GroupItemFilterSettings, [1871](#)
  - Digikam::ItemFilterSettings, [2258](#)
  - Digikam::SearchFilterModel, [3101](#)
  - Digikam::TagPropertiesFilterModel, [3391](#)
  - Digikam::TagsManagerFilterModel, [3422](#)
  - Digikam::VersionItemFilterSettings, [3576](#)
- MatchResult
  - Digikam::AlbumFilterModel, [269](#)
- matchResult
  - Digikam::AlbumFilterModel, [271](#)
- MatchStartingWithFragment
  - Digikam::TaggingActionFactory, [3360](#)
- maxCell
  - Digikam::DDateTable::Private, [985](#)
- MaxFaceListSize
  - Digikam::FaceGroup::Private, [1559](#)
- maximumBoundValues
  - Digikam::BdEngineBackend, [487](#)
- maximumNumberOfThreads
  - Digikam::ActionThreadBase, [220](#)
- maximumSize
  - Digikam::ImportThumbnailDelegate, [2124](#)
  - Digikam::ItemThumbnailDelegate, [2430](#)
  - ShowFoto::ShowfotoThumbnailDelegate, [3756](#)
- maximumThumbnailSize
  - Digikam::ThumbnailLoadThread, [3483](#)
- MaxMouseDistance
  - Digikam::FaceGroup::Private, [1559](#)
- maxValue
  - Digikam::DIntRangeBox, [1131](#)
- maxZoomFactor
  - Digikam::SinglePhotoPreviewLayout, [3243](#)
- MeaningOfDirection
  - Digikam, [126](#)
- media
  - Digikam::DOnlineTts, [1289](#)
- medianFilterPasses
  - Digikam::DRawDecoderSettings, [1380](#)
- MEDIUM
  - Digikam::DNGWriter, [1213](#)
- menubarVisibility
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- MenuCategoryFlag
  - Digikam::ActionItemModel, [209](#)

- merge
  - Digikam::Template, [3440](#)
- mergeAll
  - Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >, [2707](#)
- mergeFields
  - Digikam::MetaEngineMergeHelper< Data, Key, KeyString, KeyStringList >, [2707](#)
- mergeTAlbum
  - Digikam::AlbumManager, [298](#)
- meridianArcLength
  - Digikam::GeodeticCalculator, [1765](#)
- meridianArcLengthRadians
  - Digikam::GeodeticCalculator, [1765](#)
- message
  - Digikam::DNotificationPopup, [1250–1254](#)
- MessageType
  - Digikam::DConfigDlgTitle, [931](#)
  - Digikam::DNotificationWidget, [1260](#)
- messageType
  - Digikam::DNotificationWidget, [1263](#)
- MetadataAvailable
  - Digikam::DisjointMetadataDataFields, [1146](#)
  - Digikam::MetadataHub, [2634](#)
- MetadataDisjoint
  - Digikam::DisjointMetadataDataFields, [1146](#)
- MetadataEditingAborted
  - Digikam::ItemMetadataAdjustmentHint, [2344](#)
- MetadataEditingFinished
  - Digikam::ItemMetadataAdjustmentHint, [2344](#)
- MetadataHub
  - Digikam::MetadataHub, [2634](#)
- MetadataInvalid
  - Digikam::DisjointMetadataDataFields, [1146](#)
  - Digikam::MetadataHub, [2634](#)
- MetaDataMap
  - Digikam::MetaEngine, [2672](#)
- MetadataRemover
  - Digikam::MetadataRemover, [2653](#)
- MetadataSynchronizer
  - Digikam::MetadataSynchronizer, [2662](#)
- metadataTemplate
  - Digikam::DisjointMetadata, [1140](#)
  - Digikam::ItemInfo, [2303](#)
- MetadataTool
  - Digikam::BatchTool, [465](#)
- MetadataWritingMode
  - Digikam::MetaEngine, [2673](#)
- metadataWritingMode
  - Digikam::MetaEngine, [2690](#)
- MetaEngine
  - Digikam::MetaEngine, [2674](#)
- MetaEnginePreviews
  - Digikam::MetaEnginePreviews, [2708](#)
- MetaEngineRotation
  - Digikam::MetaEngineRotation, [2710](#)
- meteringMode
  - Digikam::DRawInfo, [1393](#)
- middleButtonPressed
  - Digikam::AbstractCheckableAlbumTreeView, [175](#)
- MidlightShade
  - Digikam::SchemeManager, [3024](#)
- MidShade
  - Digikam::SchemeManager, [3024](#)
- migrateAlbumRoot
  - Digikam::CoreDB, [709](#)
- migrateToVolume
  - Digikam::CollectionManager, [616](#)
- migrationCandidates
  - Digikam::CollectionManager, [617](#)
- mimeType
  - Digikam::ThumbnailInfo, [3471](#)
- mimeTypeTypes
  - Digikam::AbstractItemDragDropHandler, [188](#)
  - Digikam::AlbumDragDropHandler, [265](#)
  - Digikam::AlbumModelDragDropHandler, [314](#)
  - Digikam::ImportDragDropHandler, [2047](#)
  - Digikam::ItemDragDropHandler, [2228](#)
  - Digikam::TagDragDropHandler, [3341](#)
  - ShowFoto::ShowfotoDragDropHandler, [3671](#)
- minimumSizeHint
  - Digikam::DNotificationWidget, [1264](#)
  - Digikam::DPointSelect, [1346](#)
- minValue
  - Digikam::DIntRangeBox, [1131](#)
- MJPEG
  - Digikam::VidSlideSettings, [3594](#)
- MKV
  - Digikam::VidSlideSettings, [3595](#)
- MLPipelineStage
  - Digikam::MLPipelineFoundation, [2729](#)
- mocMetaObject
  - Digikam::ParallelAdapter< A >, [2821](#)
  - Digikam::ParallelWorkers, [2825](#)
- modDateTime
  - Digikam::ItemInfo, [2303](#)
- MODE
  - Digikam::GreycstorationFilter, [1861](#)
- Mode
  - Digikam::TagsPopupMenu, [3423](#)
- model
  - Digikam::AlbumSelectComboBox, [332](#)
  - Digikam::DConfigDlgView, [940](#)
  - Digikam::DRawInfo, [1393](#)
  - Digikam::GeoModelHelper, [1782](#)
  - Digikam::GPSBookmarkModelHelper, [1792](#)
  - Digikam::GPSGeoIfaceModelHelper, [1802](#)
  - Digikam::ItemGPSModelHelper, [2274](#)
  - Digikam::MapViewModelHelper, [2617](#)
  - Digikam::TrashView, [3543](#)
- modelChanged
  - Digikam::ImportCategorizedView, [2017](#)
  - ShowFoto::ShowfotoCategorizedView, [3654](#)
- modelDescription
  - Digikam::LensFunIface, [2513](#)
- modelFlags

- Digikam::GPSBookmarkModelHelper, [1792](#)
- Digikam::GPSGeofaceModelHelper, [1802](#)
- ModelIndexBasedComboBox
  - Digikam::ModelIndexBasedComboBox, [2736](#)
- modificationDate
  - Digikam::ThumbnailInfo, [3471](#)
- ModifiedScan
  - Digikam::CollectionScanner, [624](#)
- modulateProgress
  - Digikam::DImgThreadedFilter, [1119](#)
- monitorProfile
  - Digikam::IccSettings, [1939](#)
- monitorProfileFromSystem
  - Digikam::IccSettings, [1939](#)
- monthIndexForDate
  - Digikam::DateAlbumModel, [807](#)
- moreCompleteLoadingAvailable
  - Digikam::LoadSaveThread, [2571](#)
- mostRemoteNodes
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1842](#)
- mouseButtonDoubleClicked
  - Digikam::DCategoryDrawer, [893](#)
- mouseButtonPressed
  - Digikam::DCategoryDrawer, [894](#)
- mouseButtonReleased
  - Digikam::DCategoryDrawer, [894](#)
- mouseLeft
  - Digikam::DCategoryDrawer, [895](#)
- mouseModeChanged
  - Digikam::BackendGoogleMaps, [444](#)
  - Digikam::BackendMarble, [452](#)
  - Digikam::MapBackend, [2612](#)
- mouseMoved
  - Digikam::DCategoryDrawer, [895](#)
  - Digikam::DItemDelegate, [1152](#)
  - Digikam::ItemDelegateOverlay, [2218](#)
  - Digikam::ItemViewDelegate, [2453](#)
  - Digikam::ItemViewImportDelegate, [2463](#)
  - ShowFoto::ShowfotoItemViewDelegate, [3708](#)
- mouseMoveEvent
  - Digikam::ProxyLineEdit, [2888](#)
- mousePressEvent
  - Digikam::ClickDragReleaseItem, [602](#)
  - Digikam::DDateTable, [982](#)
  - Digikam::ProxyLineEdit, [2888](#)
- moveCurrentReferredImage
  - Digikam::DImageHistory, [1069](#)
- Moved
  - Digikam::CollectionImageChangeset, [605](#)
- moveItem
  - Digikam::CoreDB, [709](#)
- moveNear
  - Digikam::DNotificationPopup, [1255](#)
- moveTAlbum
  - Digikam::AlbumManager, [299](#)
- MoveToIntermediate
  - Digikam::VersionFileOperation, [3574](#)
- MP4
  - Digikam::VidSlideSettings, [3595](#)
- MPEG2
  - Digikam::VidSlideSettings, [3594](#)
- MPEG4
  - Digikam::VidSlideSettings, [3594](#)
- MPG
  - Digikam::VidSlideSettings, [3595](#)
- mSecTimeStamp
  - Digikam::DMetadata, [1187](#)
- MULTIPLIER\_16BIT
  - Digikam::ImageCurves, [1957](#)
- multithreadedSteps
  - Digikam::DImgThreadedFilter, [1119](#)
- name
  - Digikam::CoreDbUrl, [742](#)
  - Digikam::DConfigDlgWdgItem, [954](#)
  - Digikam::DownloadInfo, [1291](#)
  - Digikam::DPlugin, [1301](#)
  - Digikam::ItemInfo, [2303](#)
- NameMatchMode
  - Digikam::TaggingActionFactory, [3360](#)
- namespaceName
  - Digikam::NamespaceEntry, [2755](#)
- Natural
  - Digikam::ApplicationSettings, [388](#)
- naturalCompare
  - Digikam::CamItemSortSettings, [560](#)
  - Digikam::ItemSortSettings, [2416](#)
  - ShowFoto::ShowfotoItemSortSettings, [3704](#)
- needCheckRawDecoding
  - Digikam::LoadingDescription, [2554](#)
- needsPostLoadingManagement
  - Digikam::IccManager, [1917](#)
- NegativeBackground
  - Digikam::SchemeManager, [3022](#)
- NegativeText
  - Digikam::SchemeManager, [3024](#)
- Network
  - Digikam::CollectionLocation, [609](#)
- NetworkError
  - Digikam::DOnlineTranslator, [1272](#)
- NEUTRAL
  - Digikam::SearchTextBar, [3125](#)
- NeutralBackground
  - Digikam::SchemeManager, [3022](#)
- NeutralText
  - Digikam::SchemeManager, [3024](#)
- new\_failureTolerant
  - Digikam::DImgLoader, [1104](#)
- NewerCreationDate
  - Digikam::Haarface, [1876](#)
- NewerModificationDate
  - Digikam::Haarface, [1876](#)
- NewFile
  - Digikam::VersionFileOperation, [3574](#)
- newFile
  - Digikam::ItemScanner, [2400](#)

- newFileFullScan
  - Digikam::ItemScanner, [2400](#)
- newImages
  - Digikam::RecognitionTrainingProvider, [2946](#)
  - Digikam::TrainingDataProvider, [3534](#)
- NewPicture
  - Digikam::CamItemInfo, [557](#)
- newSearch
  - Digikam::NormalSearchTreeView, [2781](#)
- next
  - Digikam::Album, [259](#)
- nextIndexHint
  - Digikam::ImportCategorizedView, [2017](#)
  - Digikam::ItemCategorizedView, [2188](#)
  - Digikam::ItemViewCategorized, [2446](#)
  - ShowFoto::ShowfotoCategorizedView, [3655](#)
- nextInOrder
  - Digikam::ImportCategorizedView, [2017](#)
  - Digikam::ItemCategorizedView, [2188](#)
  - ShowFoto::ShowfotoCategorizedView, [3655](#)
- NO\_ACTION
  - Digikam::ExifToolProcess, [1527](#)
- NO\_RESULT
  - Digikam::SearchTextBar, [3125](#)
- NoBackend
  - Digikam::MetaEngine, [2673](#)
- NoCategories
  - Digikam::ItemSortSettings, [2414](#)
- nodeCompare
  - Digikam::KDNodeOpenFace, [2484](#)
  - Digikam::KDNodeSFace, [2486](#)
- NoError
  - Digikam::DOnlineTranslator, [1272](#)
  - Digikam::DOnlineTts, [1287](#)
- NoErrors
  - Digikam::BdEngineBackend, [482](#)
  - Digikam::DatabaseServerError, [789](#)
- NoiseReduction
  - Digikam::DRawDecoderSettings, [1376](#)
- NoLeadingSlash
  - Digikam::TagsCache, [3400](#)
- NoMatch
  - Digikam::AlbumFilterModel, [269](#)
- NonAssignedItems
  - Digikam::AutotagsAssignment, [431](#)
  - Digikam::ImageQualitySorter, [1986](#)
- NonDeterministicRandomData
  - Digikam::NonDeterministicRandomData, [2770](#)
- nonDeterministicSeed
  - Digikam::RandomNumberGenerator, [2909](#)
- NONE
  - Digikam::DNGWriter, [1213](#)
  - Digikam::FileSaveOptionsBox, [1687](#)
- None
  - Digikam::EffectMgr, [1490](#)
  - Digikam::MLPipelineFoundation, [2729](#)
- NoPreviewMode
  - Digikam::PreviewToolBar, [2864](#)
- NoPreviewZoomCtrl
  - Digikam::DZoomBar, [1458](#)
- Normal
  - Digikam::ApplicationSettings, [388](#)
- NormalBackground
  - Digikam::SchemeManager, [3022](#)
- NormalScan
  - Digikam::CollectionScanner, [624](#)
- NormalSearchTreeView
  - Digikam::NormalSearchTreeView, [2779](#)
- NormalText
  - Digikam::SchemeManager, [3024](#)
- NormalWrite
  - Digikam::FacePipeline, [1566](#)
  - Digikam::FacePipelineBase, [1575](#)
- NotificationPolicy
  - Digikam::LoadSaveThread, [2570](#)
- NotificationPolicyDirect
  - Digikam::LoadSaveThread, [2570](#)
- NotificationPolicyTimeLimited
  - Digikam::LoadSaveThread, [2570](#)
- notifyFileChanged
  - Digikam::LoadingCache, [2546](#)
  - Digikam::LoadingCacheFileWatch, [2550](#)
- notifyNewLoadingProcess
  - Digikam::LoadingCache, [2546](#)
  - Digikam::SharedLoadingTask, [3195](#)
- NoTransformation
  - Digikam::MetaEngineRotation, [2710](#)
- NotSupported
  - Digikam::DatabaseServerError, [789](#)
- NREstimate
  - Digikam::NREstimate, [2785](#)
- NRThreshold
  - Digikam::DRawDecoderSettings, [1380](#)
- NRType
  - Digikam::DRawDecoderSettings, [1380](#)
- NTSC
  - Digikam::VidSlideSettings, [3595](#)
- Null
  - Digikam::RatingComboBox, [2918](#)
- NUM\_CHANNELS
  - Digikam::ImageCurves, [1957](#)
- number
  - Digikam::RandomNumberGenerator, [2909](#)
- NUMBER\_OF\_POINTS
  - Digikam::ImageCurves, [1957](#)
- numberOfComments
  - Digikam::ItemComments, [2196](#)
- numberOfFaces
  - Digikam::FaceTagsEditor, [1641](#)
- numDayColumns
  - Digikam::DDateTable::Private, [985](#)
- numDaysThisMonth
  - Digikam::DDateTable::Private, [986](#)
- numWeekRows
  - Digikam::DDateTable::Private, [986](#)
- OlderOrLarger

- Digikam::Haarface, [1876](#)
- oneAlbumRoot
  - Digikam::CollectionManager, [617](#)
- OneCategory
  - Digikam::ItemSortSettings, [2414](#)
- onIndicesClicked
  - Digikam::AbstractMarkerTiler, [192](#)
  - Digikam::GeoModelHelper, [1782](#)
  - Digikam::GPSMarkerTiler, [1823](#)
  - Digikam::ItemMarkerTiler, [2341](#)
  - Digikam::MapViewModelHelper, [2617](#)
- onIndicesMoved
  - Digikam::GPSGeofaceModelHelper, [1802](#)
  - Digikam::ItemMarkerTiler, [2342](#)
- Open
  - Digikam::BdEngineBackend, [482](#)
- open
  - Digikam::BdEngineBackend, [487](#)
  - Digikam::IccProfile, [1924](#)
- openAlbum
  - Digikam::ItemAlbumModel, [2179](#)
- OpenCV\_KNN
  - Digikam::OpenCVDNNFaceRecognizer, [2800](#)
- OpenFace
  - Digikam::FaceScanSettings, [1624](#)
- openFilesWithDefaultApplication
  - Digikam::DFileOperations, [1022](#)
- openInFileManager
  - Digikam::DFileOperations, [1022](#)
- openOnlineDocumentation
  - Digikam, [131](#)
- OpenSchemaChecked
  - Digikam::BdEngineBackend, [482](#)
- openSetupPage
  - Digikam::DBInfoInterface, [855](#)
  - Digikam::DInfoInterface, [1128](#)
  - ShowFoto::ShowfotoInfoInterface, [3693](#)
- Operation
  - Digikam::CollectionImageChangeset, [605](#)
  - Digikam::ImageTagChangeset, [1998](#)
  - Digikam::TagChangeset, [3330](#)
- OperationType
  - Digikam, [126](#)
- operationTypeExplanation
  - Digikam::ApplicationSettings, [389](#)
- operationTypeTitle
  - Digikam::ApplicationSettings, [389](#)
- operator!=
  - Digikam::CamItemInfo, [557](#)
- operator<
  - Digikam::AlbumInfo, [280](#)
  - Digikam::SearchInfo, [3106](#)
- operator<<
  - Digikam, [131](#)
  - Digikam::CollectionImageChangeset, [606](#)
  - Digikam::DImageHistory, [1069](#)
  - Digikam::ImageTagChangeset, [1998](#)
- operator()
  - Digikam::RedEye::RegressionTree, [2950](#)
- operator=
  - Digikam::DImg, [1084](#)
  - Digikam::FocusPoint, [1730](#)
  - Digikam::SchemeManager, [3027](#)
- operator==
  - Digikam::BatchToolSet, [473](#)
  - Digikam::CamItemInfo, [557](#)
  - Digikam::ChoiceSearchModel::Entry, [599](#)
  - Digikam::DImg, [1084](#)
  - Digikam::DRawDecoderSettings, [1377](#)
  - Digikam::DRawDecoding, [1386](#)
  - Digikam::HotPixelProps, [1902](#)
  - Digikam::IccProfile, [1925](#)
  - Digikam::LoadingDescription, [2555](#)
  - Digikam::Template, [3440](#)
  - ShowFoto::ShowfotoItemInfo, [3694](#)
- operator\*=
  - Digikam::MetaEngineRotation, [2711](#)
- operator |
  - Digikam, [132](#)
- optimalWorkerCount
  - Digikam::ParallelWorkers, [2826](#)
- optimalWorkerCountReached
  - Digikam::ParallelWorkers, [2826](#)
- optimizeTimeLoading
  - Digikam::DRawDecoderSettings, [1377](#)
  - Digikam::DRawDecoding, [1386](#)
- orientation
  - Digikam::DImg, [1085](#)
  - Digikam::DRawInfo, [1394](#)
  - Digikam::ItemInfo, [2303](#)
- orientationHint
  - Digikam::DatabaseLoadSaveFileInfoProvider, [780](#)
  - Digikam::LoadSaveFileInfoProvider, [2562](#)
  - Digikam::ThumbnailInfo, [3471](#)
- Original
  - Digikam::HistoryImageId, [1895](#)
- original
  - Digikam::Imageface, [1970](#)
- originalBitDepth
  - Digikam::DImg, [1085](#)
- originalColorModel
  - Digikam::DImg, [1085](#)
- originalFilePath
  - Digikam::DImg, [1085](#)
- originalIccProfile
  - Digikam::Imageface, [1970](#)
- originalImageSize
  - Digikam::ImageZoomSettings, [2008](#)
- originalRatioSize
  - Digikam::DImg, [1085](#)
- originalRawFileName
  - Digikam::DRawInfo, [1394](#)
- originalRect
  - Digikam::DImgChildItem, [1095](#)
- originalSize
  - Digikam::DImg, [1085](#)



- Digikam::Imagelface, [1970](#)
- orthodromicDistance
  - Digikam::Ellipsoid, [1495](#)
  - Digikam::GeodeticCalculator, [1766](#)
- osdFont
  - Digikam::FrameOsdSettings, [1740](#)
- Output
  - Digikam::IccProfile, [1923](#)
- output
  - Digikam::ProcessLauncher, [2867](#)
- OutputColorSpace
  - Digikam::DRawDecoderSettings, [1376](#)
- outputColorSpace
  - Digikam::DRawDecoderSettings, [1380](#)
- outputDir
  - Digikam::VidSlideSettings, [3597](#)
- outputProfile
  - Digikam::DRawDecoderSettings, [1380](#)
- outputProfiles
  - Digikam::IccSettings, [1939](#)
- outputSize
  - Digikam::DRawInfo, [1394](#)
- outputSuffix
  - Digikam::BatchTool, [468](#)
- OverwriteAllFaces
  - Digikam::FacePipeline, [1566](#)
  - Digikam::FacePipelineBase, [1575](#)
- OverwriteUnconfirmed
  - Digikam::FacePipeline, [1566](#)
  - Digikam::FacePipelineBase, [1575](#)
- owner
  - Digikam::DRawInfo, [1394](#)
- pageRemoved
  - Digikam::DConfigDlg, [916](#)
  - Digikam::DConfigDlgWdg, [950](#)
- pageToggled
  - Digikam::DConfigDlgWdg, [950](#)
- pageWidget
  - Digikam::DConfigDlg, [916](#)
- paint
  - Digikam::ComboBoxDelegate, [650](#)
  - Digikam::Imagelface, [1970](#)
  - Digikam::TableViewColumn, [3287](#)
  - Digikam::TableViewColumns::ColumnThumbnail, [3320](#)
- PAL
  - Digikam::VidSlideSettings, [3595](#)
- ParallelAdapter
  - Digikam::ParallelAdapter< A >, [2821](#)
- ParallelWorkers
  - Digikam::ParallelWorkers, [2825](#)
- parameter
  - Digikam::FilterAction, [1706](#), [1707](#)
- parameters
  - Digikam::CoreDbAccess, [723](#)
  - Digikam::CoreDbUrl, [742](#)
  - Digikam::DImgLoaderSettings, [1107](#)
  - Digikam::SimilarityDbAccess, [3235](#)
- ParametersError
  - Digikam::DOnlineTranslator, [1272](#)
- parametersForSQLite
  - Digikam::DbEngineParameters, [845](#)
- parametersSuccessfullyRead
  - Digikam::DImgThreadedFilter, [1119](#)
  - Digikam::IccTransformFilter, [1948](#)
- parent
  - Digikam::Album, [259](#)
  - Digikam::ProgressItem, [2870](#)
  - Digikam::TableViewModel, [3325](#)
- parentDImgItem
  - Digikam::DImgChildItem, [1095](#)
- ParentMatch
  - Digikam::AlbumFilterModel, [269](#)
- ParentMenuCategory
  - Digikam::ActionItemModel, [210](#)
- parentObject
  - Digikam::IccProfilesMenuAction, [1930](#)
- parentTag
  - Digikam::TagsCache, [3402](#)
- parentTags
  - Digikam::TagsCache, [3402](#)
- ParentToChild
  - Digikam, [126](#)
- parentWidget
  - Digikam::AbstractWidgetDelegateOverlay, [202](#)
- parse
  - Digikam::AdvancedRenameWidget, [246](#)
- parseAlbumItemsRecursive
  - Digikam::DBInfolface, [856](#)
  - Digikam::DMetaInfolface, [1195](#)
- parseChildren
  - Digikam::DConfigDlgMgr, [922](#)
- parseOperation
  - Digikam::CameraNameOption, [552](#)
  - Digikam::CaseModifier, [574](#)
  - Digikam::DatabaseOption, [783](#)
  - Digikam::DateOption, [817](#)
  - Digikam::DefaultValueModifier, [998](#)
  - Digikam::DirectoryNameOption, [1136](#)
  - Digikam::FilePropertiesOption, [1683](#)
  - Digikam::MetadataOption, [2645](#)
  - Digikam::Modifier, [2740](#)
  - Digikam::Option, [2806](#)
  - Digikam::RangeModifier, [2914](#)
  - Digikam::RemoveDoublesModifier, [2967](#)
  - Digikam::ReplaceModifier, [2974](#)
  - Digikam::Rule, [2997](#)
  - Digikam::SequenceNumberOption, [3157](#)
  - Digikam::TrimmedModifier, [3552](#)
  - Digikam::UniqueModifier, [3568](#)
- parser
  - Digikam::AdvancedRenameWidget, [247](#)
- parseString
  - Digikam::AdvancedRenameWidget, [247](#)
- parseStringsValid
  - Digikam::Parser, [2828](#)

- ParsingError
  - Digikam::DOnlineTranslator, [1272](#)
- partialScan
  - Digikam::CollectionScanner, [626](#)
- PartialWrite
  - Digikam::DisjointMetadata, [1139](#)
  - Digikam::MetadataHub, [2634](#)
- passShortcutActionsToWidget
  - Digikam::DBInfofance, [856](#)
  - Digikam::DInfoInterface, [1128](#)
- path
  - Digikam::DownloadInfo, [1291](#)
- Pending
  - Digikam::QueueListView, [2893](#)
- pendingCount
  - Digikam::ActionThreadBase, [220](#)
- pendingItemInfoListFromCurrentQueue
  - Digikam::BqmInfofance, [529](#)
- permissionsString
  - Digikam::ItemPropertiesTab, [2380](#)
- PersistentWidgetDelegateOverlay
  - Digikam::PersistentWidgetDelegateOverlay, [2836](#)
- personParentTag
  - Digikam::FaceTags, [1637](#)
- PhaseOut
  - Digikam::WorkerObject, [3617](#)
- PHYSICAL
  - Digikam::Album, [255](#)
- pickLabel
  - Digikam::DisjointMetadata, [1140](#)
  - Digikam::ItemInfo, [2303](#)
- pickLabelForTag
  - Digikam::TagsCache, [3402](#)
- pickLabelFromTags
  - Digikam::TagsCache, [3402](#)
- pickLabelInterval
  - Digikam::DisjointMetadata, [1141](#)
- pickLabels
  - Digikam::PickLabelWidget, [2845](#)
- pickLabelTags
  - Digikam::TagsCache, [3403](#)
- pixelAntiAliasing
  - Digikam::DPixelsAliasFilter, [1292](#)
- pixelAntiAliasing16
  - Digikam::DPixelsAliasFilter, [1292](#)
- pixelAspectRatio
  - Digikam::DRawInfo, [1394](#)
- pixmap
  - Digikam::DConfigDlgTitle, [932](#)
- pixmapForDrag
  - Digikam::AbstractAlbumTreeView, [154](#)
  - Digikam::DragDropViewImplementation, [1365](#)
  - Digikam::ImportDelegate, [2037](#)
  - Digikam::ItemDelegate, [2213](#)
  - Digikam::ItemViewCategorized, [2446](#)
  - Digikam::TableViewTreeView, [3330](#)
  - Digikam::VersionsTreeView, [3587](#)
  - ShowFoto::ShowfotoDelegate, [3666](#)
- pixmapForItem
  - Digikam::DTrashItemModel, [1424](#)
- pixmapFromRepresentativeIndex
  - Digikam::AbstractMarkerTiler, [192](#)
  - Digikam::GeoModelHelper, [1782](#)
  - Digikam::GPSGeoifaceModelHelper, [1802](#)
  - Digikam::GPSMarkerTiler, [1823](#)
  - Digikam::ItemGPSModelHelper, [2274](#)
  - Digikam::ItemMarkerTiler, [2342](#)
  - Digikam::MapViewModelHelper, [2618](#)
- pixmapRect
  - Digikam::ImportDelegate, [2037](#)
  - Digikam::ItemDelegate, [2213](#)
  - Digikam::ItemViewDelegate, [2453](#)
  - Digikam::ItemViewImportDelegate, [2463](#)
  - ShowFoto::ShowfotoDelegate, [3666](#)
  - ShowFoto::ShowfotoItemViewDelegate, [3709](#)
- pixmapToThumbnailSize
  - Digikam::ThumbnailLoadThread, [3483](#)
- PlainMessage
  - Digikam::DConfigDlgTitle, [931](#)
- plugDatabaseFilter
  - Digikam::FacePipeline, [1567](#)
- pluginAuthors
  - Digikam::DPlugin, [1301](#)
- pluginEntriesList
  - Digikam::DPluginLoader::Private, [1339](#)
- pluginId
  - Digikam::DPluginAction, [1304](#)
- populateOSD
  - Digikam::FrameOsd, [1739](#)
- popup
  - Digikam::DPopupFrame, [1351](#)
  - Digikam::PanIconFrame, [2815](#)
- popupMenuEnabled
  - Digikam::DDateTable, [982](#)
- PopupStyle
  - Digikam::DNotificationPopup, [1249](#)
- posFromDate
  - Digikam::DDateTable, [982](#)
- position
  - Digikam::DMultiTabBar, [1200](#)
- positionChanged
  - Digikam::DImgChildItem, [1095](#)
- PositionKeys
  - Digikam::PositionKeys, [2848](#)
- positionOnImageChanged
  - Digikam::DImgChildItem, [1095](#)
- positionSelf
  - Digikam::DNotificationPopup, [1255](#)
- PositiveBackground
  - Digikam::SchemeManager, [3022](#)
- PositiveText
  - Digikam::SchemeManager, [3024](#)
- possibleCacheKeys
  - Digikam::LoadingDescription, [2555](#)
- possibleValuesForEnumField
  - Digikam::DMetadata, [1187](#)

- postLoadingManage
  - Digikam::IccPostLoadingManager, [1921](#)
- postProcess
  - Digikam::ThumbnailLoadingTask, [3475](#)
- postProcessingSettingsIsDirty
  - Digikam::DRawDecoding, [1386](#)
- postProgress
  - Digikam::DImgThreadedFilter, [1119](#)
- predict
  - Digikam::FaceClassifier, [1541](#)
- PreferFolder
  - Digikam::Haarface, [1876](#)
- pregenerate
  - Digikam::ThumbnailCreator, [3463](#)
- pregenerateGroup
  - Digikam::ThumbnailLoadThread, [3483](#)
- preload
  - Digikam::ThumbnailLoadThread, [3483](#)
- preloadThumbnails
  - Digikam::ItemThumbnailModel, [2439](#)
- premultiply
  - Digikam::DColor, [899](#)
- prepareAddExcludeDecoration
  - Digikam::AbstractCheckableAlbumModel, [169](#)
- prepareForDeletion
  - Digikam::Album, [259](#)
- prepareForDetection
  - Digikam::AbstractDetector, [187](#)
- prepareForDisplay
  - Digikam::ItemHistoryGraph, [2277](#)
- prepareForRecognition
  - Digikam::OpenCVDNNFaceRecognizer, [2800](#)
- prepareFrames
  - Digikam::VidSlideThread, [3601](#)
- prepareItemCounts
  - Digikam::AlbumManager, [299](#)
- PrepareMetadataFlag
  - Digikam::DImg, [1075](#)
- prepareMetadataToSave
  - Digikam::DImg, [1085](#)
- prepareQuery
  - Digikam::BdEngineBackend, [487](#)
- prepareRatingPixmaps
  - Digikam::ItemViewImportDelegate, [2463](#)
- prepareTiles
  - Digikam::AbstractMarkerTiler, [192](#)
  - Digikam::GPSMarkerTiler, [1823](#)
  - Digikam::ItemMarkerTiler, [2342](#)
- prePopulated
  - Digikam::BookmarksMenu, [518](#)
  - Digikam::ModelMenu, [2738](#)
- preprocess
  - Digikam::ImportItemModel, [2070](#)
  - Digikam::ItemModel, [2352](#)
  - Digikam::RecognitionPreprocessor, [2944](#)
  - ShowFoto::ShowfotoItemModel, [3699](#)
- PreserveEmbeddedProfile
  - Digikam::ICCSettingsContainer, [1941](#)
- prev
  - Digikam::Album, [260](#)
- preview
  - Digikam::Imagelface, [1970](#)
- PreviewBothImagesHorz
  - Digikam::PreviewToolBar, [2864](#)
- PreviewBothImagesHorzCont
  - Digikam::PreviewToolBar, [2864](#)
- PreviewBothImagesVert
  - Digikam::PreviewToolBar, [2864](#)
- PreviewBothImagesVertCont
  - Digikam::PreviewToolBar, [2864](#)
- PreviewCameraMode
  - Digikam::ImportStackedView, [2113](#)
- previewItem
  - Digikam::GraphicsDImgView, [1856](#)
- PreviewLoadThread
  - Digikam::PreviewLoadThread, [2858](#)
- PreviewMode
  - Digikam::PreviewToolBar, [2864](#)
- PreviewOriginalImage
  - Digikam::PreviewToolBar, [2864](#)
- previewReference
  - Digikam::Imagelface, [1970](#)
- previewSize
  - Digikam::Imagelface, [1970](#)
- previewSupported
  - Digikam::DPluginDImg, [1326](#)
- PreviewTargetImage
  - Digikam::PreviewToolBar, [2864](#)
- PreviewToggleOnMouseOver
  - Digikam::PreviewToolBar, [2864](#)
- PreviewType
  - Digikam::Imagelface, [1969](#)
- PreviewZoomCtrl
  - Digikam::DZoomBar, [1458](#)
- printComments
  - Digikam::FrameOsd, [1739](#)
- printExiv2ExceptionError
  - Digikam::MetaEngine::Private, [2705](#)
- printExiv2MessageHandler
  - Digikam::MetaEngine::Private, [2705](#)
- printTags
  - Digikam::FrameOsd, [1739](#)
- proceed
  - Digikam::EmptyImageListProvider, [1505](#)
  - Digikam::QListImageListProvider, [2890](#)
- process
  - Digikam::FacePipeline, [1568](#)
  - Digikam::ItemFilterModelFilterer, [2251](#)
  - Digikam::ItemFilterModelPreparer, [2254](#)
  - Digikam::RecognitionWorker, [2949](#)
  - Digikam::TrainerWorker, [3533](#)
- PROCESS\_CANCELED
  - Digikam::DNGWriter, [1212](#)
- PROCESS\_COMPLETE
  - Digikam::DNGWriter, [1212](#)
- PROCESS\_CONTINUE

- Digikam::DNGWriter, [1212](#)
- PROCESS\_FAILED
  - Digikam::DNGWriter, [1212](#)
- profilesForDescription
  - Digikam::IccSettings, [1939](#)
- ProfileType
  - Digikam::IccProfile, [1922](#)
- progress
  - Digikam::DImgThreadedFilter, [1120](#)
  - Digikam::ProgressItem, [2871](#)
- progressCompleted
  - Digikam::DProgressWdg, [1360](#)
- progressInfo
  - Digikam::DImgLoaderObserver, [1106](#)
  - Digikam::IccTransformFilter, [1949](#)
  - Digikam::LoadingTask, [2561](#)
  - Digikam::SavingTask, [3007](#)
  - Digikam::SharedLoadingTask, [3195](#)
- progressItemAdded
  - Digikam::ProgressItem, [2871](#)
  - Digikam::ProgressManager, [2880](#)
- progressItemCanceled
  - Digikam::ProgressItem, [2871](#)
  - Digikam::ProgressManager, [2880](#)
- progressItemCompleted
  - Digikam::ProgressItem, [2871](#)
  - Digikam::ProgressManager, [2880](#)
- progressItemLabel
  - Digikam::ProgressItem, [2871](#)
  - Digikam::ProgressManager, [2880](#)
- progressItemProgress
  - Digikam::ProgressItem, [2872](#)
  - Digikam::ProgressManager, [2880](#)
- progressItemStatus
  - Digikam::ProgressItem, [2872](#)
  - Digikam::ProgressManager, [2881](#)
- progressItemThumbnail
  - Digikam::ProgressItem, [2872](#)
  - Digikam::ProgressManager, [2881](#)
- progressItemUsesBusyIndicator
  - Digikam::ProgressItem, [2872](#)
  - Digikam::ProgressManager, [2881](#)
- progressScheduled
  - Digikam::DProgressWdg, [1360](#)
- progressStatusChanged
  - Digikam::DProgressWdg, [1360](#)
- progressThumbnailChanged
  - Digikam::DProgressWdg, [1360](#)
- properties
  - Digikam::TagsCache, [3403](#)
- propertiesAssigned
  - Digikam::ItemVisibilityController, [2474](#)
- PropertiesChanged
  - Digikam::TagChangeset, [3330](#)
- property
  - Digikam::DConfigDlgMgr, [923](#)
- propertyMap
  - Digikam::DConfigDlgMgr, [923](#)
- propertyValue
  - Digikam::TagsCache, [3403](#)
- provider
  - Digikam::ItemCopyright, [2206](#)
- ProxyClickLineEdit
  - Digikam::ProxyClickLineEdit, [2886](#)
- ProxyLineEdit
  - Digikam::ProxyLineEdit, [2888](#)
- ProxyType
  - Digikam::SystemSettings, [3280](#)
- publicTags
  - Digikam::TagsCache, [3403](#)
- pureColorMask
  - Digikam::DImg, [1086](#)
- purgePathFromReferredImages
  - Digikam::DImageHistory, [1069](#)
- putData
  - Digikam::UndoCache, [3564](#)
- putImage
  - Digikam::LoadingCache, [2546](#)
  - Digikam::LoadingCacheInterface, [2551](#)
- putImageData
  - Digikam::DImg, [1086](#)
- putThumbnail
  - Digikam::LoadingCache, [2546](#)
- qHash
  - Digikam, [132](#)
- QIMAGE
  - Digikam::DImg, [1075](#)
- QPointSquareDistance
  - Digikam, [132](#)
- QSXGA
  - Digikam::VidSlideSettings, [3596](#)
- QSXGAPLUS
  - Digikam::VidSlideSettings, [3596](#)
- Quality
  - Digikam::PreviewSettings, [2861](#)
- QualityScanMode
  - Digikam::ImageQualitySorter, [1985](#)
- queryErrorHandling
  - Digikam::BdEngineBackend, [488](#)
- queryOperationWakeAll
  - Digikam::BdEngineBackendPrivate, [492](#)
- querySendNotifyEvent
  - Digikam::SharedLoadingTask, [3195](#)
- QueryStateEnum
  - Digikam::BdEngineBackend, [482](#)
- QueuePoolItemsList
  - Digikam, [124](#)
- queuesMap
  - Digikam::QueueMgrWindow, [2897](#)
- QUXGA
  - Digikam::VidSlideSettings, [3596](#)
- QVGA
  - Digikam::VidSlideSettings, [3595](#)
- QXGA
  - Digikam::VidSlideSettings, [3596](#)

- radius
  - Digikam::CMat, [604](#)
- radiusOfCurvature
  - Digikam::Ellipsoid, [1495](#)
- RandomNumberGenerator
  - Digikam::RandomNumberGenerator, [2909](#)
- randomString
  - Digikam::WSToolUtils, [3635](#)
- RasterGraphics
  - Digikam::MimeFilter, [2718](#)
- rating
  - Digikam::DisjointMetadata, [1141](#)
  - Digikam::ItemInfo, [2303](#)
- RatingComboBoxWidget
  - Digikam::RatingComboBoxWidget, [2923](#)
- ratingInterval
  - Digikam::DisjointMetadata, [1141](#)
- ratingPixmap
  - Digikam::ItemViewDelegate, [2453](#)
- ratingRect
  - Digikam::ItemViewDelegate, [2454](#)
  - Digikam::ItemViewImportDelegate, [2463](#)
- RatingValue
  - Digikam::RatingComboBox, [2918](#)
- rawColors
  - Digikam::DRawInfo, [1394](#)
- rawDataUniqueID
  - Digikam::DRawInfo, [1394](#)
- RawDecodingCustomSettings
  - Digikam::LoadingDescription, [2553](#)
- RawDecodingDefaultSettings
  - Digikam::LoadingDescription, [2553](#)
- RawDecodingGlobalSettings
  - Digikam::LoadingDescription, [2553](#)
- RawDecodingHint
  - Digikam::LoadingDescription, [2552](#)
- rawDecodingSettings
  - Digikam::BatchTool, [468](#)
  - Digikam::DImg, [1086](#)
- RawDecodingTimeOptimized
  - Digikam::LoadingDescription, [2553](#)
- rawFileIdentify
  - Digikam::DRawDecoder, [1372](#)
- RAWFiles
  - Digikam::MimeFilter, [2718](#)
- rawFiles
  - Digikam::DRawDecoder, [1373](#)
- rawFilesList
  - Digikam::DRawDecoder, [1373](#)
- rawFilesVersion
  - Digikam::DRawDecoder, [1373](#)
- rawImages
  - Digikam::DRawInfo, [1395](#)
- rawPrm
  - Digikam::DRawDecoding, [1387](#)
- RawProcessingFilter
  - Digikam::RawProcessingFilter, [2938](#)
- RAWQuality
  - Digikam::DRawDecoderSettings, [1380](#)
- reactivateToolDialog
  - Digikam::DPluginGeneric, [1334](#)
- read
  - Digikam::DatabaseBlob, [775](#)
  - Digikam::SearchFieldAlbum, [3041](#)
  - Digikam::SearchFieldCheckBox, [3045](#)
  - Digikam::SearchFieldChoice, [3049](#)
  - Digikam::SearchFieldColorDepth, [3053](#)
  - Digikam::SearchFieldKeyword, [3062](#)
  - Digikam::SearchFieldLabels, [3065](#)
  - Digikam::SearchFieldMonthDay, [3069](#)
  - Digikam::SearchFieldPageOrientation, [3073](#)
  - Digikam::SearchFieldRangeDate, [3076](#)
  - Digikam::SearchFieldRangeDouble, [3080](#)
  - Digikam::SearchFieldRangeInt, [3084](#)
  - Digikam::SearchFieldRangeTime, [3088](#)
  - Digikam::SearchFieldRating, [3092](#)
  - Digikam::SearchFieldText, [3096](#)
- READ\_FORMATS
  - Digikam::ExifToolProcess, [1527](#)
- readableFormats
  - Digikam::ExifToolParser, [1523](#)
- ReadConfirmedFaces
  - Digikam::FacePipeline, [1566](#)
- readdNewTags
  - Digikam::RGTagModel, [2987](#)
- readdTag
  - Digikam::RGTagModel, [2987](#)
- ReadFacesForTraining
  - Digikam::FacePipeline, [1566](#)
- readFromConfig
  - Digikam::DbEngineParameters, [845](#)
- readFullScreenSettings
  - Digikam::DXmlGuiWindow, [1447](#)
- readMsgBoxShouldBeShown
  - Digikam::ApplicationSettings, [390](#)
  - Digikam::DMessageBox, [1170](#)
- readNext
  - Digikam::SearchXmlReader, [3149](#)
- readParameters
  - Digikam::AntiVignettingFilter, [381](#)
  - Digikam::AutoExpoFilter, [422](#)
  - Digikam::AutoLevelsFilter, [426](#)
  - Digikam::BCGFilter, [478](#)
  - Digikam::BlurFilter, [508](#)
  - Digikam::BlurFXFilter, [512](#)
  - Digikam::BorderFilter, [525](#)
  - Digikam::BWSepiaFilter, [536](#)
  - Digikam::CBFilter, [581](#)
  - Digikam::CharcoalFilter, [587](#)
  - Digikam::ColorFXFilter, [640](#)
  - Digikam::ContentAwareFilter, [658](#)
  - Digikam::CurvesFilter, [754](#)
  - Digikam::DistortionFXFilter, [1150](#)
  - Digikam::EmbossFilter, [1501](#)
  - Digikam::EqualizeFilter, [1509](#)
  - Digikam::FilmFilter, [1697](#)

- Digikam::FilmGrainFilter, [1701](#)
- Digikam::FilterActionFilter, [1712](#)
- Digikam::FreeRotationFilter, [1746](#)
- Digikam::GreycstorationFilter, [1862](#)
- Digikam::HotPixelFixer, [1901](#)
- Digikam::HSLFilter, [1911](#)
- Digikam::IccTransformFilter, [1949](#)
- Digikam::InfraredFilter, [2146](#)
- Digikam::InvertFilter, [2153](#)
- Digikam::LensDistortionFilter, [2506](#)
- Digikam::LensFunFilter, [2512](#)
- Digikam::LevelsFilter, [2519](#)
- Digikam::LocalContrastFilter, [2577](#)
- Digikam::MixerFilter, [2723](#)
- Digikam::NormalizeFilter, [2774](#)
- Digikam::NRFilter, [2789](#)
- Digikam::OilPaintFilter, [2794](#)
- Digikam::RainDropFilter, [2908](#)
- Digikam::RawProcessingFilter, [2939](#)
- Digikam::RefocusFilter, [2960](#)
- Digikam::SharpenFilter, [3204](#)
- Digikam::ShearFilter, [3209](#)
- Digikam::StretchFilter, [3273](#)
- Digikam::TextureFilter, [3455](#)
- Digikam::TonalityFilter, [3516](#)
- Digikam::UnsharpMaskFilter, [3573](#)
- Digikam::WBFilter, [3610](#)
- readParametersError
  - Digikam::IccTransformFilter, [1949](#)
- readSearch
  - Digikam::SearchWindow, [3143](#)
- readSettings
  - Digikam::DRawDecoderWidget, [1384](#)
  - Digikam::FrameOsdSettings, [1740](#)
  - Digikam::VidSlideSettings, [3597](#)
- readSettingsFromGroup
  - Digikam::BackendGoogleMaps, [444](#)
  - Digikam::BackendMarble, [452](#)
  - Digikam::RGWidget, [2990](#)
- readToEndOfElement
  - Digikam::SearchXmlReader, [3149](#)
- readToFirstField
  - Digikam::SearchXmlReader, [3149](#)
- readToList
  - Digikam::BdEngineBackend, [488](#)
- readToStartOfElement
  - Digikam::SearchXmlReader, [3149](#)
- ReadUnconfirmedFaces
  - Digikam::FacePipeline, [1566](#)
- readWithExifTool
  - Digikam::MetaEngine, [2690](#)
- readyForIncrementalRefresh
  - Digikam::ImportItemModel, [2070](#)
  - Digikam::ItemModel, [2352](#)
  - ShowFoto::ShowfotoItemModel, [3699](#)
- realZoomFactor
  - Digikam::ImageZoomSettings, [2008](#)
- rebuildDuplicatesAlbums
  - Digikam::Haarface, [1879](#)
- receive
  - Digikam::ItemLISTERJobGrowingPartsSendingReceiver, [2325](#)
  - Digikam::ItemLISTERJobPartsSendingReceiver, [2327](#)
  - Digikam::ItemLISTERValueListReceiver, [2333](#)
- recognize
  - Digikam::OpenCVDNNFaceRecognizer, [2800](#)
- recognizeAccuracy
  - Digikam::FaceScanSettings, [1625](#)
- recognizeFaces
  - Digikam::FacialRecognitionWrapper, [1654](#)
- RecognizeMarkedFaces
  - Digikam::FaceScanSettings, [1625](#)
- RecognizeOnly
  - Digikam::FaceScanSettings, [1624](#)
- recommendedImageSize
  - Digikam::FaceDetector, [1554](#)
- recommendedImageSizeForDetection
  - Digikam::OpenCVDNNFaceDetector, [2798](#)
- recordChangeset
  - Digikam::CoreDbBackend, [728](#)
- recordHint
  - Digikam::CollectionScannerHintContainerImplementation, [633](#)
- recordHints
  - Digikam::CollectionScannerHintContainerImplementation, [633](#), [634](#)
- rect
  - Digikam::DImageChildItem, [1095](#)
- RECURSIVE
  - Digikam::StateSavingObject, [3261](#)
- RedEyeCorrectionFilter
  - Digikam::RedEyeCorrectionFilter, [2955](#)
- reduceEdges
  - Digikam::ItemHistoryGraph, [2277](#)
- referredImages
  - Digikam::DImageHistory, [1069](#)
- ReflImageSelMethod
  - Digikam::Haarface, [1876](#)
- refresh
  - Digikam::AlbumManager, [299](#)
  - Digikam::CollectionManager, [617](#)
  - Digikam::ItemAlbumModel, [2179](#)
- refreshThumbnails
  - Digikam::DTrashItemModel, [1424](#)
- regenerateTiles
  - Digikam::GPSMarkerTiler, [1824](#)
  - Digikam::ItemMarkerTiler, [2342](#)
- regExp
  - Digikam::Rule, [2997](#)
- regionSelectionChanged
  - Digikam::BackendGoogleMaps, [444](#)
  - Digikam::BackendMarble, [452](#)
- regionSelectionMoved
  - Digikam::PanIconWidget, [2818](#)
- registerButton

- Digikam::Rule, [2997](#)
- registeredImageIds
  - Digikam::SimilarityDb, [3232](#)
- registerEditorPlugins
  - Digikam::DPluginLoader, [1338](#)
- registerExtraPluginsActions
  - Digikam::EditorWindow, [1487](#)
- registerGenericPlugins
  - Digikam::DPluginLoader, [1338](#)
- registerLabelsActions
  - Digikam::TagsActionMngr, [3397](#)
- registerMenu
  - Digikam::Rule, [2999](#)
- registerPluginsActions
  - Digikam::DXmlGuiWindow, [1447](#)
- registerRawImportPlugins
  - Digikam::DPluginLoader, [1338](#)
- registerSettingsWidget
  - Digikam::BatchTool, [468](#)
- registerTagsActionCollections
  - Digikam::TagsActionMngr, [3397](#)
- registerWidget
  - Digikam::GeolocationSettings, [1778](#)
- registerXmpNameSpace
  - Digikam::MetaEngine, [2691](#)
- reInitialize
  - Digikam::ThumbBarDock, [3460](#)
- rejectAll
  - Digikam::FaceGroup, [1558](#)
- rejected
  - Digikam::AssignNameWidget, [408](#)
- rejectFaces
  - Digikam::DigikamItemView, [1064](#)
- relationCloud
  - Digikam::ItemHistoryGraph, [2277](#)
  - Digikam::ItemInfo, [2303](#)
- relativePath
  - Digikam::ItemInfo, [2304](#)
- relativeRect
  - Digikam::DImgChildItem, [1096](#)
- RelativeSize
  - Digikam::AlbumThumbnailLoader, [356](#)
- relativeToAbsolute
  - Digikam::TagRegion, [3395](#)
- releaseWidget
  - Digikam::BackendGoogleMaps, [444](#)
  - Digikam::BackendMarble, [453](#)
- reload
  - Digikam::BackendGoogleMaps, [445](#)
  - Digikam::BackendMarble, [453](#)
- remove
  - Digikam::FacePipeline, [1568](#)
  - Digikam::ItemComments, [2197](#)
  - Digikam::ItemPosition, [2356](#)
  - Digikam::OpenCVDNNFaceRecognizer, [2800](#)
- removeAction
  - Digikam::DNotificationWidget, [1264](#)
- removeAll
  - Digikam::ItemComments, [2197](#)
  - Digikam::ItemCopyright, [2206](#)
- removeAllComments
  - Digikam::ItemComments, [2197](#)
- removeAllFaces
  - Digikam::FaceTagsEditor, [1641](#)
- removeAllImageComments
  - Digikam::CoreDB, [709](#)
- removeAllItemCopyrightProperties
  - Digikam::CoreDB, [709](#)
- removeAllTags
  - Digikam::ItemInfo, [2304](#)
- removeAlphaChannel
  - Digikam::DImg, [1086](#)
- removeAltitude
  - Digikam::ItemPosition, [2356](#)
- removeAndCopyFile
  - Digikam::DFileOperations, [1022](#)
- removeButton
  - Digikam::DMultiTabBar, [1201](#)
- removeByFilePath
  - Digikam::ThumbsDb, [3488](#)
- removeByUniqueHash
  - Digikam::ThumbsDb, [3488](#)
- Removed
  - Digikam::CollectionImageChangeset, [605](#)
- RemovedAll
  - Digikam::CollectionImageChangeset, [605](#)
- RemovedDeleted
  - Digikam::CollectionImageChangeset, [605](#)
- removeExifColorSpace
  - Digikam::DMetadata, [1187](#)
- removeExifTag
  - Digikam::MetaEngine, [2691](#)
- removeExifThumbnail
  - Digikam::MetaEngine, [2691](#)
- removeExtraData
  - Digikam::Album, [260](#)
- removeFace
  - Digikam::FaceTagsEditor, [1642](#)
- removeFaces
  - Digikam::DigikamItemView, [1064](#)
- removeFaceTraining
  - Digikam::FaceTagsIface, [1646](#)
- removeFaceVector
  - Digikam::FaceDb, [1544](#), [1546](#)
- removeFromGroup
  - Digikam::ItemInfo, [2304](#)
- removeFromXmpTagStringBag
  - Digikam::DMetadata, [1187](#)
  - Digikam::MetaEngine, [2691](#)
- removeGPSInfo
  - Digikam::MetaEngine, [2691](#)
- removeImage
  - Digikam::LoadingCache, [2546](#)
- removeImageComment
  - Digikam::CoreDB, [709](#)
- removeImageFingerprint

- Digikam::SimilarityDb, [3232](#)
- removeImageRelation
  - Digikam::CoreDB, [710](#)
- removeImages
  - Digikam::LoadingCache, [2546](#)
- removeImageSimilarity
  - Digikam::SimilarityDb, [3232](#)
- removeImageTagProperties
  - Digikam::CoreDB, [710](#)
- removeIndex
  - Digikam::ImportItemModel, [2070](#)
  - Digikam::ItemModel, [2352](#)
  - ShowFoto::ShowfotoItemModel, [3699](#)
- removeIptcTag
  - Digikam::MetaEngine, [2691](#)
- removeItemAllTags
  - Digikam::CoreDB, [710](#)
- removeItemCopyrightProperties
  - Digikam::CoreDB, [710](#)
- removeItemFacesMap
  - Digikam::DMetadata, [1188](#)
- removeItemPosition
  - Digikam::CoreDB, [711](#)
- removeItemPositionAltitude
  - Digikam::CoreDB, [711](#)
- removeItems
  - Digikam::CoreDB, [711](#)
  - Digikam::DTrashItemModel, [1425](#)
- removeItemsFromAlbum
  - Digikam::CoreDB, [711](#)
- removeItemsPermanently
  - Digikam::CoreDB, [712](#)
- removeItemTag
  - Digikam::CoreDB, [712](#)
- removeListener
  - Digikam::SharedLoadingTask, [3195](#)
- removeLoadingProcess
  - Digikam::LoadingCache, [2547](#)
- removeLocation
  - Digikam::CollectionManager, [617](#)
- removeMarkerIndexFromGrid
  - Digikam::ItemMarkerTiler, [2342](#)
- removeMetadataTemplate
  - Digikam::ItemInfo, [2304](#)
- removeNormalTag
  - Digikam::FaceTagsEditor, [1642](#)
  - Digikam::FaceUtils, [1650](#)
- removeNormalTags
  - Digikam::FaceUtils, [1651](#)
- RemoveOldMetadataPreviews
  - Digikam::DImg, [1076](#)
- removePage
  - Digikam::DConfigDlg, [916](#)
  - Digikam::DConfigDlgWdg, [950](#)
  - Digikam::DConfigDlgWdgModel, [961](#)
- removeTab
  - Digikam::DMultiTabBar, [1201](#)
- removeTag
  - Digikam::ItemInfo, [2304](#)
- removeTagProperties
  - Digikam::CoreDB, [712](#)
- removeTags
  - Digikam::FileActionMngrDatabaseWorker, [1673](#)
- removeTagsFromItems
  - Digikam::CoreDB, [712](#)
- removeThumbnail
  - Digikam::LoadingCache, [2547](#)
- removeThumbnails
  - Digikam::LoadingCache, [2547](#)
- removeUngroupedModel
  - Digikam::MapWidget, [2624](#)
- removeXmpKeywords
  - Digikam::DMetadata, [1188](#)
  - Digikam::MetaEngine, [2692](#)
- removeXmpSubCategories
  - Digikam::DMetadata, [1188](#)
  - Digikam::MetaEngine, [2692](#)
- removeXmpSubjects
  - Digikam::DMetadata, [1188](#)
  - Digikam::MetaEngine, [2692](#)
- removeXmpTag
  - Digikam::MetaEngine, [2692](#)
- renameAlbum
  - Digikam::CoreDB, [713](#)
- renameFile
  - Digikam::DFileOperations, [1022](#)
  - Digikam::IOJobsThread, [2162](#)
- renameItem
  - Digikam::CoreDB, [713](#)
- renamePAlbum
  - Digikam::AlbumManager, [299](#)
- renameTAlbum
  - Digikam::AlbumManager, [300](#)
- render
  - Digikam::HistogramPainter, [1888](#)
- renderingMode
  - Digikam::EditorToolThreaded, [1481](#)
- Replace
  - Digikam::VersionFileOperation, [3574](#)
- ReplaceAllEntries
  - Digikam::ItemCopyright, [2204](#)
- replaceColorLabel
  - Digikam::DisjointMetadata, [1141](#)
- replaceComments
  - Digikam::ItemComments, [2197](#)
- replaceFrom
  - Digikam::ItemComments, [2197](#)
  - Digikam::ItemCopyright, [2206](#)
- ReplaceLanguageEntry
  - Digikam::ItemCopyright, [2204](#)
- replacementQtMetacall
  - Digikam::ParallelWorkers, [2826](#)
- ReplaceMode
  - Digikam::ItemCopyright, [2204](#)
- replaceProfiles
  - Digikam::IccProfilesMenuAction, [1930](#)



- replaceProfilesSqueezed
  - Digikam::IccProfilesComboBox, [1928](#)
- repositionRect
  - Digikam::BlackFrameToolTip, [502](#)
  - Digikam::FreeSpaceToolTip, [1749](#)
  - Digikam::ItemViewToolTip, [2468](#)
- ReproducibleFilter
  - Digikam::FilterAction, [1706](#)
- requestActiveTab
  - Digikam::SidebarWidget, [3226](#)
- requestIncrementalRefresh
  - Digikam::ImportItemModel, [2070](#)
  - Digikam::ItemModel, [2352](#)
  - ShowFoto::ShowfotoItemModel, [3700](#)
- Rescan
  - Digikam::CollectionScanner, [624](#)
  - Digikam::FaceScanSettings, [1624](#)
- rescan
  - Digikam::ItemScanner, [2400](#)
- reseed
  - Digikam::RandomNumberGenerator, [2910](#)
- reset
  - Digikam::AltLangStrEdit, [371](#)
  - Digikam::CurvesWidget, [758](#)
  - Digikam::CustomStepsDoubleSpinBox, [761](#)
  - Digikam::CustomStepsIntSpinBox, [763](#)
  - Digikam::DImg, [1087](#)
  - Digikam::Rule, [2999](#)
  - Digikam::SearchFieldAlbum, [3041](#)
  - Digikam::SearchFieldCheckBox, [3045](#)
  - Digikam::SearchFieldChoice, [3049](#)
  - Digikam::SearchFieldComboBox, [3056](#)
  - Digikam::SearchFieldLabels, [3065](#)
  - Digikam::SearchFieldMonthDay, [3069](#)
  - Digikam::SearchFieldRangeDate, [3076](#)
  - Digikam::SearchFieldRangeDouble, [3080](#)
  - Digikam::SearchFieldRangeInt, [3084](#)
  - Digikam::SearchFieldRangeTime, [3088](#)
  - Digikam::SearchFieldRating, [3092](#)
  - Digikam::SearchFieldText, [3096](#)
  - Digikam::SearchWindow, [3143](#)
  - Digikam::UniqueModifier, [3569](#)
- resetChecked
  - Digikam::ChoiceSearchModel, [597](#)
- ResetExifOrientationTag
  - Digikam::DImg, [1076](#)
- resetMetaData
  - Digikam::DImg, [1087](#)
- resetPAAlbumSelection
  - Digikam::AlbumSelectors, [341](#)
- resetPostProcessingSettings
  - Digikam::DRawDecoding, [1386](#)
- resetSelection
  - Digikam::AlbumSelectors, [341](#)
- resetTAlbumSelection
  - Digikam::AlbumSelectors, [342](#)
- resetTime
  - Digikam::CoreDbOperationGroup, [735](#)
  - Digikam::FaceDbOperationGroup, [1552](#)
- resetUI
  - Digikam::CurvesWidget, [758](#)
- Resize
  - Digikam::DImgBuiltinFilter, [1090](#)
- resize
  - Digikam::DImg, [1087](#)
- resizeEvent
  - Digikam::DPopupFrame, [1351](#)
  - Digikam::PanelIconFrame, [2816](#)
- RESNET50
  - Digikam, [127](#)
- resolvedImageHistory
  - Digikam::ItemScanner, [2400](#)
- resolveHistoryImageld
  - Digikam::ItemScanner, [2401](#)
- resolveImageHistory
  - Digikam::ItemScanner, [2401](#)
- restart
  - Digikam::ScanController, [3013](#)
- restartCollectionScan
  - Digikam::ScanController, [3013](#)
- restore
  - Digikam::Sidebar, [3218](#), [3219](#)
- RESTORE\_PREVIEW
  - Digikam::ExifToolProcess, [1527](#)
- restoreCurve
  - Digikam::CurvesWidget, [758](#)
- restoreDTrashItems
  - Digikam::IOJobsThread, [2162](#)
- restoreRGTagList
  - Digikam::GPSItemContainer, [1805](#)
- restoreSelectionFromHistory
  - Digikam::AlbumLabelsSearchHandler, [282](#)
  - Digikam::LabelsTreeView, [2500](#)
- restoreSettings
  - Digikam::TagList, [3363](#)
- restoreState
  - Digikam::SidebarSplitter, [3222](#)
- result
  - Digikam::DetectionBenchmark, [1011](#)
  - Digikam::RecognitionBenchmark, [2943](#)
- ResultStatus
  - Digikam::ExifToolProcess, [1527](#)
- resumeCollectionScan
  - Digikam::ScanController, [3013](#)
- retrain
  - Digikam::FaceClassifier, [1541](#)
- RetrainAll
  - Digikam::FaceScanSettings, [1625](#)
- retrieveAlbum
  - Digikam::AbstractAlbumModel, [145](#)
- retrieveCamItemInfo
  - Digikam::ImportItemModel, [2071](#)
- retrieveImage
  - Digikam::LoadingCache, [2547](#)
- retrieveItemInfo
  - Digikam::ItemModel, [2352](#)

- retrieveLoadingProcess
  - Digikam::LoadingCache, [2547](#)
- retrieveShowfotoItemInfo
  - ShowFoto::ShowfotoItemModel, [3700](#)
- retrieveSignatureFromDB
  - Digikam::Haarface, [1879](#)
- retrieveThumbnail
  - Digikam::LoadingCache, [2547](#)
- retrieveThumbnailPixmap
  - Digikam::ImportDelegate, [2037](#)
  - Digikam::ItemDelegate, [2213](#)
  - ShowFoto::ShowfotoDelegate, [3666](#)
- ReturnMatchingDefaultOrFirstLanguage
  - Digikam::ItemComments, [2194](#)
- ReturnMatchingLanguageOnly
  - Digikam::ItemComments, [2194](#)
- ReturnMatchingOrDefaultLanguage
  - Digikam::ItemComments, [2194](#)
- returnPressed
  - Digikam::DPlainTextEdit, [1296](#)
  - Digikam::DTextEdit, [1413](#)
- reverseExifRotate
  - Digikam::DImg, [1087](#)
- reverseFilter
  - Digikam::DImgBuiltinFilter, [1092](#)
- reverseRotateAndFlip
  - Digikam::DImg, [1087](#)
- reverseToOrientation
  - Digikam::TagRegion, [3395](#)
- RGBBackend
  - Digikam::RGBBackend, [2977](#)
- RGBInterpolate4Colors
  - Digikam::DRawDecoderSettings, [1381](#)
- rgData
  - Digikam::RGInfo, [2979](#)
- RGInfo
  - Digikam::RGInfo, [2978](#)
- RGTagModel
  - Digikam::RGTagModel, [2982](#)
- RGWidget
  - Digikam::RGWidget, [2989](#)
- rightMargin
  - Digikam::DCategoryDrawer, [895](#)
- rightsUsageTerms
  - Digikam::ItemCopyright, [2206](#)
- Role
  - Digikam::DConfigDlgModel, [926](#)
  - Digikam::FacePipelineFaceTagsIface, [1593](#)
- rollbackTransaction
  - Digikam::BdEngineBackend, [488](#)
- rootAlbumAvailable
  - Digikam::AbstractAlbumModel, [145](#)
- RootAlbumBehavior
  - Digikam::AbstractAlbumModel, [142](#)
- rootAlbumBehavior
  - Digikam::AbstractAlbumModel, [145](#)
- rootAlbumIndex
  - Digikam::AbstractAlbumModel, [145](#)
- rootImages
  - Digikam::ItemHistoryGraph, [2277](#)
- roots
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1842](#)
- rootsOf
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1842](#)
- Rotate180
  - Digikam::MetaEngineRotation, [2710](#)
- Rotate270
  - Digikam::MetaEngineRotation, [2710](#)
- Rotate90
  - Digikam::MetaEngineRotation, [2710](#)
- rotate90
  - Digikam::EditorCore, [1468](#)
- rotateAndFlip
  - Digikam::DImg, [1087](#)
- rotateExifQImage
  - Digikam::MetaEngine, [2692](#)
- rotateFaces
  - Digikam::FaceTagsEditor, [1642](#)
- RotationBehaviorFlag
  - Digikam::MetaEngineSettingsContainer, [2715](#)
- row\_stride
  - Digikam::CMat, [604](#)
- rowCount
  - ShowFoto::ShowfotoItemModel, [3700](#)
- rowFromAlbum
  - Digikam::Album, [260](#)
- rows
  - Digikam::Mat, [2630](#)
- rowsRemoved
  - Digikam::ItemViewCategorized, [2446](#)
- run
  - Digikam::ActionThreadBase, [220](#)
  - Digikam::DImgThreadedFilter, [1120](#)
  - Digikam::DPluginRawImport, [1342](#)
  - Digikam::DynamicThread, [1453](#)
  - Digikam::ExifToolThread, [1533](#)
  - Digikam::ImageHistogram, [1967](#)
  - Digikam::LoadSaveThread, [2571](#)
  - Digikam::ScanStateFilter, [3020](#)
- runFiles
  - Digikam::DServiceMenu, [1402](#)
- runningFlag
  - Digikam::DynamicThread, [1453](#)
- s\_bingLanguageCodes
  - Digikam::DOnlineTranslator::Private, [1281](#)
- s\_emotionCodes
  - Digikam::DOnlineTts::Private, [1290](#)
- s\_googleLanguageCodes
  - Digikam::DOnlineTranslator::Private, [1281](#)
- s\_inlineTranslateString
  - Digikam, [132](#)
- s\_lingvaLanguageCodes
  - Digikam::DOnlineTranslator::Private, [1281](#)
- s\_metaEngineMutex

- Digikam, [136](#)
- s\_metaEngineSupportBmff
  - Digikam, [136](#)
- s\_metaEngineWarnOrError
  - Digikam, [136](#)
- s\_rawFileExtensionsdWithDesc
  - Digikam, [132](#)
- s\_rawFileExtensionsVersion
  - Digikam, [133](#)
- s\_rfc3066ForXMP
  - Digikam, [136](#)
- s\_rfc3066LanguageCodesBing
  - Digikam::DOnlineTranslator::Private, [1282](#)
- s\_rfc3066LanguageCodesGoogle
  - Digikam::DOnlineTranslator::Private, [1282](#)
- s\_rfc3066LanguageCodesLingva
  - Digikam::DOnlineTranslator::Private, [1282](#)
- s\_rfc3066LanguageCodesYandex
  - Digikam::DOnlineTranslator::Private, [1282](#)
- s\_setXmpTagStringFromEntry
  - Digikam, [133](#)
- s\_stage
  - Digikam, [136](#)
- s\_voiceCodes
  - Digikam::DOnlineTts::Private, [1290](#)
- s\_yandexLanguageCodes
  - Digikam::DOnlineTranslator::Private, [1282](#)
- safelyRemoveAlbums
  - Digikam::CollectionScanner, [626](#)
- sameReferredImage
  - Digikam::ItemScanner, [2401](#)
- sampleText
  - Digikam::DFontProperties, [1031](#)
- saturation
  - Digikam::DColorValueSelector, [906](#)
  - Digikam::DHueSaturationSelector, [1041](#)
- save
  - Digikam::LoadSaveThread, [2571](#)
  - Digikam::ManagedLoadSaveThread, [2609](#)
  - Digikam::MetaEngine, [2692](#)
  - Digikam::WorkflowManager, [3624](#)
- SaveAndDelete
  - Digikam::VersionFileOperation, [3574](#)
- saveChanges
  - Digikam::GPSItemContainer, [1805](#)
  - Digikam::ItemGPS, [2271](#)
- saveCurve
  - Digikam::CurvesWidget, [759](#)
- saveDestinationUrl
  - Digikam::EditorWindow, [1487](#)
- savedFormat
  - Digikam::DImg, [1087](#)
- savefromDImg
  - Digikam::BatchTool, [468](#)
- saveLevelsToGimpLevelsFile
  - Digikam::ImageLevels, [1973](#)
- saveMsgBoxShouldBeShown
  - Digikam::ApplicationSettings, [390](#)
- Digikam::DMessageBox, [1170](#)
- saveSettingsToGroup
  - Digikam::BackendGoogleMaps, [445](#)
  - Digikam::BackendMarble, [453](#)
  - Digikam::RGWidget, [2990](#)
- saveState
  - Digikam::AlbumSelectors, [342](#)
  - Digikam::SidebarSplitter, [3222](#), [3223](#)
  - Digikam::StateSavingObject, [3263](#)
- savingProgress
  - Digikam::LoadSaveThread, [2572](#)
- ScaleMode
  - Digikam::TimeLineWidget, [3508](#)
- scanAlbums
  - Digikam::CoreDB, [713](#)
- ScanAll
  - Digikam::FacePipeline, [1566](#)
  - Digikam::FacePipelineBase, [1574](#)
- ScanDeferredFiles
  - Digikam::NewItemFinder, [2762](#)
- scanFile
  - Digikam::CollectionScanner, [626](#), [627](#)
- scannedFiles
  - Digikam::CollectionScanner, [627](#)
- scannedInfo
  - Digikam::ScanController, [3013](#)
- ScanNew
  - Digikam::FacePipelineBase, [1574](#)
- scanSearches
  - Digikam::CoreDB, [713](#)
- scanTags
  - Digikam::CoreDB, [713](#)
- ScanTask
  - Digikam::FaceScanSettings, [1624](#)
- scene
  - Digikam::ItemExtendedProperties, [2229](#)
- schedule
  - Digikam::ParallelWorkers, [2826](#)
  - Digikam::WorkerObject, [3618](#)
- ScheduleCollectionScan
  - Digikam::NewItemFinder, [2762](#)
- scheduleCollectionScan
  - Digikam::ScanController, [3013](#)
- scheduleCollectionScanExternal
  - Digikam::ScanController, [3014](#)
- scheduleCollectionScanRelaxed
  - Digikam::ScanController, [3014](#)
- SchemeManager
  - Digikam::SchemeManager, [3024](#)
- screenCoordinates
  - Digikam::BackendGoogleMaps, [445](#)
  - Digikam::BackendMarble, [453](#)
- scrollPointOnPoint
  - Digikam::GraphicsDImgView, [1856](#)
- scrollToRelaxed
  - Digikam::ItemViewCategorized, [2446](#)
- scrollToSelectedAlbum
  - Digikam::AbstractAlbumTreeView, [154](#)

- SDTV1
  - Digikam::VidSlideSettings, [3595](#)
- SDTV2
  - Digikam::VidSlideSettings, [3596](#)
- SDTV3
  - Digikam::VidSlideSettings, [3596](#)
- SEARCH
  - Digikam::Album, [255](#)
- search
  - Digikam::SearchWindow, [3143](#)
- searchEdited
  - Digikam::SearchWindow, [3143](#)
- searchesListing
  - Digikam::SearchesDBJobsThread, [3034](#)
- searchId
  - Digikam::CoreDbUrl, [742](#)
- SearchModel
  - Digikam::SearchModel, [3112](#)
- SearchModificationHelper
  - Digikam::SearchModificationHelper, [3115](#)
- searchTextSettings
  - Digikam::AlbumFilterModel, [271](#)
- searchTextSettingsAboutToChange
  - Digikam::AlbumFilterModel, [271](#)
- searchTextSettingsChanged
  - Digikam::AlbumFilterModel, [271](#)
- searchUrl
  - Digikam::CoreDbUrl, [742](#)
- SearchWindow
  - Digikam::SearchWindow, [3143](#)
- SearchXmlCachingReader
  - Digikam::SearchXmlCachingReader, [3146](#)
- SearchXmlWriter
  - Digikam::SearchXmlWriter, [3152](#)
- seed
  - Digikam::RandomNumberGenerator, [2910](#)
- seedByTime
  - Digikam::RandomNumberGenerator, [2910](#)
- seedNonDeterministic
  - Digikam::RandomNumberGenerator, [2910](#)
- selectAll
  - Digikam::DPluginConfView, [1312](#)
  - Digikam::TableView, [3284](#)
- selectBbox
  - Digikam::DNNFaceDetectorBase, [1220](#)
- Selected
  - Digikam::FocusPoint, [1729](#)
  - Digikam::QueueListView, [2893](#)
  - Digikam::TimeLineWidget, [3509](#)
- selected
  - Digikam::AssignNameWidget, [408](#)
  - Digikam::ImportCategorizedView, [2018](#)
  - ShowFoto::ShowfotoCategorizedView, [3655](#)
- selectedAlbumIds
  - Digikam::AlbumSelectors, [342](#)
- selectedAlbums
  - Digikam::AlbumSelectors, [342](#)
- selectedAlbumsAndTags
  - Digikam::AlbumSelectors, [342](#)
- selectedAlbumsChanged
  - Digikam::AbstractAlbumTreeView, [154](#)
- selectedDateRange
  - Digikam::TimeLineWidget, [3509](#)
- SelectedInFocus
  - Digikam::FocusPoint, [1729](#)
- selectedItemInfoListFromCurrentQueue
  - Digikam::BqmlInfolface, [529](#)
- selectedLabels
  - Digikam::LabelsTreeView, [2501](#)
- selectedTagIds
  - Digikam::AlbumSelectors, [342](#)
- selectedTags
  - Digikam::AlbumSelectors, [342](#)
- selectedUrls
  - Digikam::ItemIconView, [2290](#)
- Selection
  - Digikam::SchemeManager, [3023](#)
  - Digikam::VidSlideSettings, [3594](#)
- selection
  - Digikam::ImageIface, [1971](#)
- selectionChanged
  - Digikam::ItemViewCategorized, [2447](#)
- selectionCleared
  - Digikam::ItemViewCategorized, [2447](#)
- selectionForRect
  - Digikam::DCategorizedView::Private, [889](#)
- SelectionMode
  - Digikam::TimeLineWidget, [3509](#)
- selectionModel
  - Digikam::GeoModelHelper, [1783](#)
  - Digikam::GPSBookmarkModelHelper, [1792](#)
  - Digikam::GPSGeoIfaceModelHelper, [1802](#)
  - Digikam::ItemGPSModelHelper, [2274](#)
  - Digikam::MapViewModelHelper, [2618](#)
- selectionRect
  - Digikam::ImageIface, [1971](#)
- semiMajorAxis
  - Digikam::Ellipsoid, [1495](#)
- semiMinorAxis
  - Digikam::Ellipsoid, [1496](#)
- sendViewportEventToView
  - Digikam::AbstractAlbumTreeViewSelectComboBox, [162](#)
  - Digikam::ListViewComboBox, [2543](#)
  - Digikam::StayPoppedUpComboBox, [3268](#)
  - Digikam::TreeViewComboBox, [3547](#)
- sensitivity
  - Digikam::DRawInfo, [1395](#)
- serialNumber
  - Digikam::DRawInfo, [1395](#)
- serverPrivatePath
  - Digikam::DbEngineParameters, [845](#)
- ServiceError
  - Digikam::DOnlineTranslator, [1272](#)
- servicesForOpenWith
  - Digikam::DServiceMenu, [1403](#)

setAction  
  Digikam::DImgBuiltinFilter, 1092

setActionCategory  
  Digikam::DPluginAction, 1304

setActionValue  
  Digikam::DbEngineActionType, 835

setActive  
  Digikam::AbstractMarkerTiler, 193  
  Digikam::AbstractWidgetDelegateOverlay, 203  
  Digikam::ActionVersionsOverlay, 225  
  Digikam::AlbumFolderViewSideBarWidget, 277  
  Digikam::AssignNameOverlay, 403  
  Digikam::BackendGoogleMaps, 445  
  Digikam::BackendMarble, 453  
  Digikam::DateFolderViewSideBarWidget, 814  
  Digikam::FaceRejectionOverlay, 1619  
  Digikam::FuzzySearchSideBarWidget, 1756  
  Digikam::GPSMarkerTiler, 1824  
  Digikam::GPSSearchSideBarWidget, 1831  
  Digikam::GPSSearchView, 1834  
  Digikam::GroupIndicatorOverlay, 1868  
  Digikam::HoverButtonDelegateOverlay, 1906  
  Digikam::ImportCoordinatesOverlay, 2030  
  Digikam::ImportDownloadOverlay, 2044  
  Digikam::ImportLockOverlay, 2081  
  Digikam::ImportRatingOverlay, 2096  
  Digikam::ImportRotateOverlay, 2102  
  Digikam::ItemCoordinatesOverlay, 2201  
  Digikam::ItemDelegateOverlay, 2218  
  Digikam::ItemFullScreenOverlay, 2265  
  Digikam::ItemMarkerTiler, 2343  
  Digikam::ItemRatingOverlay, 2387  
  Digikam::ItemRotateOverlay, 2392  
  Digikam::ItemSelectionOverlay, 2406  
  Digikam::LabelsSideBarWidget, 2497  
  Digikam::MapBackend, 2612  
  Digikam::MapWidgetView, 2629  
  Digikam::PeopleSideBarWidget, 2833  
  Digikam::PersistentWidgetDelegateOverlay, 2836  
  Digikam::SearchSideBarWidget, 3121  
  Digikam::ShowHideVersionsOverlay, 3213  
  Digikam::SidebarWidget, 3226  
  Digikam::TagsLineEditOverlay, 3415  
  Digikam::TagViewSideBarWidget, 3435  
  Digikam::ThumbnailImageCatcher, 3469  
  Digikam::TimelineSideBarWidget, 3506  
  ShowFoto::ShowfotoCoordinatesOverlay, 3659

setActiveTab  
  Digikam::Sidebar, 3219

setAddExcludeTristate  
  Digikam::AbstractCheckableAlbumModel, 169

setAlbumCaption  
  Digikam::CoreDB, 714

setAlbumCategory  
  Digikam::CoreDB, 715

setAlbumDate  
  Digikam::CoreDB, 715

setAlbumIcon  
  Digikam::CoreDB, 715

setAlbumManagerCurrentAlbum  
  Digikam::AbstractAlbumTreeView, 154

setAlbumModel  
  Digikam::ContextMenuHelper, 669

setAlbumModels  
  Digikam::AddTagsComboBox, 231  
  Digikam::AddTagsLineEdit, 234  
  Digikam::AssignNameWidget, 408

setAlbumModificationDate  
  Digikam::CoreDB, 715

setAlbumRootCaseSensitivity  
  Digikam::CoreDB, 716

setAlbumRootLabel  
  Digikam::CoreDB, 716

setAlbumRootPath  
  Digikam::CoreDB, 716

setAlbumRootsToSearch  
  Digikam::Haarface, 1879

setAlbumRootType  
  Digikam::CoreDB, 716

setAlbumSelected  
  Digikam::AlbumSelectors, 342

setAllOpen  
  Digikam::GroupItemFilterSettings, 1871

setAllSelectedText  
  Digikam::AlbumSelectComboBox, 333

setAltitude  
  Digikam::ItemPosition, 2356

setAnchor  
  Digikam::DNotificationPopup, 1255

setArrowDirection  
  Digikam::DSelector, 1401

setAuthorsList  
  Digikam::CaptionsMap, 571

setAutoDelete  
  Digikam::DNotificationPopup, 1255

setAutoHideTimeout  
  Digikam::DConfigDlgTitle, 932

setAutoSuggest  
  Digikam::FaceGroup, 1558

setBackend  
  Digikam::MapWidget, 2624

setBackgroundColor  
  Digikam::DFontProperties, 1031

setBaseAlbum  
  Digikam::WSAlbum, 3625

setBlockedEventTypes  
  Digikam::DWItemDelegate, 1437

setBlockUpdateSignalOnDrag  
  Digikam::DAbstractSliderSpinBox, 767

setBranchHistory  
  Digikam::BatchTool, 468

setBuddy  
  Digikam::DConfigDlgTitle, 932

setButtonBox  
  Digikam::DConfigDlg, 916

setButtonsExclusive

- Digikam::ColorLabelWidget, [647](#)
- Digikam::PickLabelWidget, [2845](#)
- setCacheOptions
  - Digikam::LoadingCacheInterface, [2551](#)
- setCacheSize
  - Digikam::LoadingCache, [2547](#)
- setCameraThumbsController
  - Digikam::ImportItemModel, [2071](#)
  - Digikam::ImportThumbnailModel, [2132](#)
- setCamItemInfos
  - Digikam::ImportItemModel, [2071](#)
- setCaseSensitive
  - Digikam::SearchTextBar, [3125](#)
- setCategorizedModel
  - Digikam::DCategorizedSortFilterProxyModel, [880](#)
- setCategory
  - Digikam::ItemScanner, [2401](#)
- setCenter
  - Digikam::BackendGoogleMaps, [445](#)
  - Digikam::BackendMarble, [453](#)
- setCenterPosition
  - Digikam::FocusPoint, [1730](#)
- setChannelFromBinary
  - Digikam::ImageCurves, [1956](#)
- setChannelType
  - Digikam::HistogramPainter, [1888](#)
- setCheckable
  - Digikam::AlbumSelectComboBox, [333](#)
  - Digikam::DConfigDlgWdgItem, [954](#)
- setChecked
  - Digikam::ChoiceSearchModel, [598](#)
  - Digikam::DConfigDlgWdgItem, [954](#)
- setCheckNewTags
  - Digikam::TagCheckView, [3337](#)
- setCheckOnMiddleClick
  - Digikam::AbstractCheckableAlbumTreeView, [175](#)
- setChoice
  - Digikam::ChoiceSearchModel, [598](#)
- setChooserMode
  - Digikam::DColorValueSelector, [906](#)
  - Digikam::DHueSaturationSelector, [1041](#)
- setCloseButton
  - Digikam::DDatePicker, [974](#)
- setCloseButtonVisible
  - Digikam::DNotificationWidget, [1264](#)
- setCloseOnActivate
  - Digikam::AlbumSelectComboBox, [333](#)
- setColor
  - Digikam::DColor, [899](#)
  - Digikam::DFontProperties, [1031](#)
- setColorLabel
  - Digikam::ItemInfo, [2304](#)
- setColorLabels
  - Digikam::ColorLabelWidget, [648](#)
- setColorValue
  - Digikam::DColorValueSelector, [906](#)
  - Digikam::DHueSaturationSelector, [1042](#)
- setComment
  - Digikam::DConfigDlgTitle, [933](#)
- setComments
  - Digikam::MetaEngine, [2693](#)
- setConfigGroup
  - Digikam::DateFolderView, [811](#)
  - Digikam::DConfigDlg, [917](#)
  - Digikam::FilterSideBarWidget, [1717](#)
  - Digikam::FuzzySearchView, [1759](#)
  - Digikam::GPSSearchView, [1834](#)
  - Digikam::StateSavingObject, [3263](#)
- setConfigGroupName
  - Digikam::DXmlGuiWindow, [1448](#)
- setConfiguration
  - Digikam::TableViewColumns::ColumnAudioVideoProperties, [3293](#)
  - Digikam::TableViewColumns::ColumnFileProperties, [3303](#)
  - Digikam::TableViewColumns::ColumnGeoProperties, [3308](#)
  - Digikam::TableViewColumns::ColumnPhotoProperties, [3317](#)
- setConsoleTraces
  - Digikam::ProcessLauncher, [2867](#)
- setConstraintInterface
  - Digikam::TaggingActionFactory, [3360](#)
- setContainerWidget
  - Digikam::VisibilityController, [3603](#)
- setContextMenuItems
  - Digikam::TagFolderView, [3357](#)
  - Digikam::TagMngrTreeView, [3372](#)
- setContextMenuIcon
  - Digikam::AbstractAlbumTreeView, [154](#)
- setContextParentTag
  - Digikam::TagCompleter, [3339](#)
- setContinueOnError
  - Digikam::FilterActionFilter, [1712](#)
- setControlButtonsPlacement
  - Digikam::DItemsList, [1159](#)
- setControlWidgets
  - Digikam::AdvancedRenameWidget, [247](#)
- setCopyrightNotice
  - Digikam::ItemCopyright, [2206](#)
- setCoreDatabasePath
  - Digikam::DbEngineParameters, [845](#)
- setCoreDbWatch
  - Digikam::CoreDbBackend, [728](#)
- setCountHash
  - Digikam::AbstractCountingAlbumModel, [181](#)
- setCreator
  - Digikam::ItemCopyright, [2207](#)
- setCurrent
  - Digikam::SqueezedComboBox, [3255](#)
- setCurrentAlbums
  - Digikam::AbstractAlbumTreeView, [154](#)
  - Digikam::AlbumManager, [300](#)
- setCurrentInfo
  - Digikam::ImportCategorizedView, [2018](#)
  - Digikam::ItemCategorizedView, [2188](#)

- ShowFoto::ShowfotoCategorizedView, [3655](#)
- setCurrentLanguage
  - Digikam::DPlainTextEdit, [1296](#)
  - Digikam::DTextEdit, [1414](#)
- setCurrentOrientation
  - Digikam::JPEGUtils::JpegRotator, [2479](#)
- setCurrentPage
  - Digikam::DConfigDlg, [917](#)
  - Digikam::DConfigDlgView, [940](#)
  - Digikam::DConfigDlgWdg, [950](#)
- setCurrentProfile
  - Digikam::IccProfilesComboBox, [1928](#)
- setCurrentTag
  - Digikam::AddTagsComboBox, [231](#)
  - Digikam::AddTagsLineEdit, [234](#)
  - Digikam::AssignNameWidget, [408](#)
- setCurrentUrl
  - Digikam::ImportCategorizedView, [2018](#)
  - Digikam::ItemCategorizedView, [2188](#)
  - ShowFoto::ShowfotoCategorizedView, [3655](#)
- setCurrentUrlWhenAvailable
  - Digikam::ItemCategorizedView, [2189](#)
- setCurrentWhenAvailable
  - Digikam::ImportCategorizedView, [2018](#)
  - Digikam::ItemCategorizedView, [2189](#)
  - ShowFoto::ShowfotoCategorizedView, [3655](#)
- setCustomDatePainting
  - Digikam::DDateTable, [982](#)
- setData
  - Digikam::AbstractCheckableAlbumModel, [169](#)
  - Digikam::ImportThumbnailModel, [2132](#)
  - Digikam::ItemThumbnailModel, [2439](#)
  - ShowFoto::ShowfotoThumbnailModel, [3763](#)
- setDatabase
  - Digikam::AlbumManager, [300](#)
- setDate
  - Digikam::DDateEdit, [969](#)
  - Digikam::DDatePicker, [974](#)
  - Digikam::DDateTable, [982](#)
- setDateTime
  - Digikam::DDateTimeEdit, [989](#)
  - Digikam::DisjointMetadata, [1141](#)
  - Digikam::ItemInfo, [2305](#)
- setDayFilter
  - Digikam::ItemFilterModel, [2246](#)
- setDbEngineErrorHandler
  - Digikam::BdEngineBackend, [488](#)
- setDefaultAlbumModel
  - Digikam::AlbumSelectComboBox, [333](#)
- setDefaultFieldOperator
  - Digikam::SearchXmlWriter, [3153](#)
- setDefaultMaximumNumberOfThreads
  - Digikam::ActionThreadBase, [220](#)
- setDefaultViewOptions
  - Digikam::DItemDelegate, [1153](#)
  - Digikam::ImportDelegate, [2037](#)
  - Digikam::ImportThumbnailDelegate, [2124](#)
  - Digikam::ItemDelegate, [2213](#)
  - Digikam::ItemThumbnailDelegate, [2430](#)
  - Digikam::ItemViewDelegate, [2454](#)
  - Digikam::ItemViewImportDelegate, [2464](#)
- ShowFoto::ShowfotoDelegate, [3667](#)
- ShowFoto::ShowfotoItemViewDelegate, [3709](#)
- ShowFoto::ShowfotoThumbnailDelegate, [3756](#)
- setDefaultWidget
  - Digikam::DConfigDlgView, [940](#)
- setDescriptionBoxVisible
  - Digikam::ColorLabelWidget, [648](#)
  - Digikam::PickLabelWidget, [2846](#)
- setDestinationFile
  - Digikam::JPEGUtils::JpegRotator, [2479](#)
- setDestinationGeographicPoint
  - Digikam::GeodeticCalculator, [1766](#)
- setDirection
  - Digikam::GeodeticCalculator, [1766](#)
- setDirectSourceImportModel
  - Digikam::ImportFilterModel, [2055](#)
  - Digikam::ImportSortFilterModel, [2111](#)
- setDirectSourceItemModel
  - Digikam::ImageSortFilterModel, [1997](#)
  - Digikam::ItemFilterModel, [2246](#)
- setDirectSourceShowfotoModel
  - ShowFoto::ShowfotoFilterModel, [3677](#)
  - ShowFoto::ShowfotoSortFilterModel, [3732](#)
- setDisplayingWidget
  - Digikam::PreviewLoadThread, [2859](#)
  - Digikam::ThumbnailLoadThread, [3483](#)
- setDisplayWidget
  - Digikam::ImageZoomSettings, [2008](#)
- setDocumentName
  - Digikam::JPEGUtils::JpegRotator, [2479](#)
- setDoNotEmbedOutputProfile
  - Digikam::IccTransform, [1943](#)
- setDownloaded
  - Digikam::CoreDbDownloadHistory, [733](#)
- setDragDropHandler
  - Digikam::AbstractAlbumModel, [145](#)
  - Digikam::DragDropModelImplementation, [1362](#)
- setDRawDecoderSettings
  - Digikam::BatchTool, [468](#)
- setDrawDraggedItems
  - Digikam::DCategorizedView, [885](#)
- setDropIndex
  - Digikam::AbstractAlbumModel, [145](#)
- setEasingCurve
  - Digikam::ItemVisibilityController, [2474](#)
- setEmbeddedProfile
  - Digikam::IccTransform, [1943](#)
- setEmitDataChanged
  - Digikam::ImportThumbnailModel, [2133](#)
  - Digikam::ItemThumbnailModel, [2440](#)
  - ShowFoto::ShowfotoThumbnailModel, [3763](#)
- setEnabledContextMenu
  - Digikam::AbstractAlbumTreeView, [155](#)
- setEnabled
  - Digikam::DConfigDlgWdgItem, [954](#)

- Digikam::DIntRangeBox, [1132](#)
- setEnabledDrag
  - Digikam::AbstractAlbumModel, [145](#)
- setEnabledToolTips
  - Digikam::AlbumSelectionTreeView, [339](#)
- setEngineApiKey
  - Digikam::DOnlineTranslator, [1276](#)
- setEngineUrl
  - Digikam::DOnlineTranslator, [1276](#)
- setEntryPrefix
  - Digikam::StateSavingObject, [3263](#)
- setErrorDescription
  - Digikam::BatchTool, [469](#)
- setExceptionList
  - Digikam::VersionItemFilterSettings, [3576](#)
- setExif
  - Digikam::MetaEngine, [2693](#)
- setExifComment
  - Digikam::MetaEngine, [2693](#)
- setExifOrient
  - Digikam::Canvas, [566](#)
- setExifOrientation
  - Digikam::FileActionMngrDatabaseWorker, [1673](#)
- setExifRotate
  - Digikam::ThumbnailCreator, [3463](#)
- setExifTagData
  - Digikam::MetaEngine, [2693](#)
- setExifTagLong
  - Digikam::MetaEngine, [2693](#)
- setExifTagRational
  - Digikam::MetaEngine, [2693](#)
- setExifTagString
  - Digikam::MetaEngine, [2694](#)
- setExifTagURational
  - Digikam::MetaEngine, [2694](#)
- setExifTagUShort
  - Digikam::MetaEngine, [2694](#)
- setExifTagVariant
  - Digikam::MetaEngine, [2694](#)
- setExifThumbnail
  - Digikam::MetaEngine, [2694](#)
- setExifToolProgram
  - Digikam::ExifToolProcess, [1529](#)
- setExifXmpTagDataVariant
  - Digikam, [133](#)
- setExpandNewCurrentItem
  - Digikam::AbstractAlbumTreeView, [155](#)
- setExpandOnSingleClick
  - Digikam::AbstractAlbumTreeView, [155](#)
- setExposureSettings
  - Digikam::Canvas, [567](#)
- setExtraData
  - Digikam::Album, [260](#)
- setFaceDetectionSize
  - Digikam::DNNFaceDetectorYuNet, [1226](#)
- setFaceType
  - Digikam::DConfigDlg, [917](#)
  - Digikam::DConfigDlgView, [940](#)
- setFieldOperator
  - Digikam::SearchXmlWriter, [3153](#)
- setFilePath
  - Digikam::MetaEngine, [2695](#)
- setFileWatch
  - Digikam::LoadingCache, [2548](#)
- setFilter
  - Digikam::DPluginConfView, [1312](#)
  - ShowFoto::ShowfotoStackViewFavoriteList, [3737](#)
- setFilterActions
  - Digikam::FilterActionFilter, [1713](#)
- setFilterBehavior
  - Digikam::AlbumFilterModel, [272](#)
- setFilterModel
  - Digikam::AddTagsLineEdit, [234](#)
  - Digikam::SearchTextBarDb, [3129](#)
- setFilterSearchType
  - Digikam::SearchFilterModel, [3101](#)
- setFilterSettings
  - Digikam::CoreDB, [717](#)
- setFilterVersion
  - Digikam::DImgThreadedFilter, [1120](#)
- setFocusOnWidget
  - Digikam::AssignNameOverlay, [403](#)
  - Digikam::PersistentWidgetDelegateOverlay, [2837](#)
- setFont
  - Digikam::DFontProperties, [1031](#)
- setFontSize
  - Digikam::DDatePicker, [974](#)
  - Digikam::DDateTable, [982](#)
- setForeignKeyChecks
  - Digikam::BdEngineBackend, [488](#)
- setFromTemplate
  - Digikam::ItemCopyright, [2207](#)
- setFullScreenOptions
  - Digikam::DXmlGuiWindow, [1448](#)
- setGPSInfo
  - Digikam::MetaEngine, [2695](#)
- setGraphicsView
  - Digikam::SinglePhotoPreviewLayout, [3244](#)
- setGroupCaption
  - Digikam::SearchXmlWriter, [3153](#)
- setGroupedModel
  - Digikam::MapWidget, [2624](#)
- setGroupingOperateOnAll
  - Digikam::ApplicationSettings, [390](#)
- setGroupItemFilterSettings
  - Digikam::ItemFilterModel, [2247](#)
- setGroupList
  - Digikam::ExifToolListView, [1516](#)
- setGroupOperator
  - Digikam::SearchXmlWriter, [3153](#)
- setHeader
  - Digikam::DConfigDlgWdgItem, [954](#)
- setHighlightOnResult
  - Digikam::SearchTextBar, [3126](#)
- setHighlightPixmap
  - Digikam::ThumbnailLoadThread, [3483](#)



- setHighlightSelection
  - Digikam::HistogramPainter, [1889](#)
- setHistogram
  - Digikam::HistogramPainter, [1889](#)
- setHistory
  - Digikam::ItemHistoryGraphModel, [2285](#)
- setHistoryBranchAfter
  - Digikam::DImg, [1088](#)
- setHSL
  - Digikam::DColor, [899](#)
- setHudWidget
  - Digikam::RegionFrameItem, [2963](#)
- setHue
  - Digikam::DColorValueSelector, [906](#)
  - Digikam::DHueSaturationSelector, [1042](#)
- setIccProfile
  - Digikam::DMetadata, [1188](#)
- setICCSettings
  - Digikam::Canvas, [567](#)
- setIcon
  - Digikam::DConfigDlgWdgItem, [955](#)
  - Digikam::DNotificationWidget, [1264](#)
- setIconShowOverlays
  - Digikam::ApplicationSettings, [390](#)
- setImage
  - Digikam::ClockPhotoDialog, [603](#)
  - Digikam::GraphicsDImgItem, [1853](#)
- setImageComment
  - Digikam::CoreDB, [717](#)
- setImageData
  - Digikam::BatchTool, [469](#)
- setImageDateTime
  - Digikam::MetaEngine, [2695](#)
- setImageProperty
  - Digikam::CoreDB, [717](#)
- setImages
  - Digikam::EmptyImageListProvider, [1505](#)
  - Digikam::QListImageListProvider, [2890](#)
- setImageSize
  - Digikam::ImageZoomSettings, [2008](#)
- setImageSmoothScale
  - Digikam::ImageZoomSettings, [2009](#)
- setImageSorting
  - Digikam::ApplicationSettings, [390](#)
- setImageTagPropertiesJoined
  - Digikam::ItemQueryBuilder, [2383](#)
- setImportFilterModel
  - Digikam::ImportContextMenuHelper, [2027](#)
- setIndent
  - Digikam::DSelector, [1401](#)
- setInfo
  - Digikam::FaceGroup, [1558](#)
  - Digikam::FocusPointGroup, [1732](#)
- setInitialSelectedItem
  - Digikam::ItemViewCategorized, [2447](#)
- setInitPreview
  - Digikam::EditorTool, [1474](#)
- setInputUrl
  - Digikam::BatchTool, [469](#)
- setIntent
  - Digikam::IccTransform, [1943](#)
- setIntermediate
  - Digikam::WSCoboBoxIntermediate, [3626](#)
- setInternalServerPath
  - Digikam::DbEngineParameters, [845](#)
- setInternalValue
  - Digikam::DAbstractSliderSpinBox, [767](#)
  - Digikam::DDoubleSliderSpinBox, [993](#)
  - Digikam::DSliderSpinBox, [1406](#)
- setInterval
  - Digikam::DIntRangeBox, [1132](#)
- setIOFileSettings
  - Digikam::BatchTool, [469](#)
- setIptc
  - Digikam::MetaEngine, [2695](#)
- setIptcKeywords
  - Digikam::MetaEngine, [2696](#)
- setIptcSubCategories
  - Digikam::MetaEngine, [2696](#)
- setIptcSubjects
  - Digikam::MetaEngine, [2696](#)
- setIptcTagData
  - Digikam::MetaEngine, [2696](#)
- setIptcTagsStringList
  - Digikam::MetaEngine, [2696](#)
- setIptcTagString
  - Digikam::MetaEngine, [2697](#)
- setIsLessThanHandler
  - Digikam::DItemsList, [1159](#)
- setIsToolBox
  - Digikam::DExpanderBoxExclusive, [1018](#)
- setItem
  - Digikam::GraphicsDImgView, [1856](#)
- setItemAlbum
  - Digikam::CoreDB, [718](#)
- setItemColorWorkSpace
  - Digikam::MetaEngine, [2697](#)
- setItemCopyrightProperty
  - Digikam::CoreDB, [718](#)
- setItemDelegate
  - Digikam::DConfigDlgView, [940](#)
- setItemDimensions
  - Digikam::MetaEngine, [2697](#)
- setItemFacesMap
  - Digikam::DMetadata, [1188](#)
- setItemFilterModel
  - Digikam::ContextMenuHelper, [669](#)
- setItemFilterSettings
  - Digikam::ItemAlbumFilterModel, [2173](#)
  - Digikam::ItemFilterModel, [2247](#)
- setItemHistory
  - Digikam::CoreDB, [718](#)
- setItemIccProfile
  - Digikam::MetaEngine, [2697](#)
- setItemInfo
  - Digikam::BatchTool, [469](#)

- Digikam::DBInfolface, [856](#)
- Digikam::DMetaInfolface, [1195](#)
- setItemInfos
  - Digikam::ItemModel, [2352](#)
- setItemManualOrder
  - Digikam::CoreDB, [718](#)
- setItemModel
  - Digikam::ModelCompleter, [2733](#)
- setItemModificationDate
  - Digikam::CoreDB, [718](#)
- setItemOrientation
  - Digikam::MetaEngine, [2697](#)
- setItemPreview
  - Digikam::MetaEngine, [2697](#)
- setItemProgramId
  - Digikam::MetaEngine, [2698](#)
- setItemSortSettings
  - Digikam::ItemFilterModel, [2247](#)
- setItemStatus
  - Digikam::CoreDB, [718](#)
- setItemThatShallBeShown
  - Digikam::ItemVisibilityController, [2474](#)
- setKeepsFilePathCache
  - Digikam::ItemModel, [2352](#)
- setKeepsFilePathCache
  - Digikam::ImportItemModel, [2071](#)
  - ShowFoto::ShowfotoItemModel, [3700](#)
- setLabel
  - Digikam::CollectionManager, [617](#)
  - Digikam::ProgressItem, [2873](#)
- setLabelText
  - Digikam::ChoiceSearchComboBox, [595](#)
- setLastChainedTool
  - Digikam::BatchTool, [469](#)
- setLastError
  - Digikam::CoreDbAccess, [723](#)
  - Digikam::FaceDbAccess, [1547](#)
  - Digikam::SimilarityDbAccess, [3235](#)
  - Digikam::ThumbsDbAccess, [3489](#)
- setLatitude
  - Digikam::ItemPosition, [2356](#)
- setLayoutStyle
  - Digikam::AdvancedRenameWidget, [247](#)
- setLeftRightItems
  - Digikam::LightTableWindow, [2537](#)
- setLevelGammaValue
  - Digikam::ImageLevels, [1973](#)
- setLineEditText
  - Digikam::TreeViewLineEditComboBox, [3550](#)
- setLinesVisible
  - Digikam::AltLangStrEdit, [371](#)
  - Digikam::DPlainTextEdit, [1296](#)
  - Digikam::DTextEdit, [1414](#)
- setListOnlyAvailable
  - Digikam::ItemLister, [2323](#)
- setListTemporarySearches
  - Digikam::SearchFilterModel, [3101](#)
- setLoadingPolicy
  - Digikam::ManagedLoadSaveThread, [2609](#)
- setLoadingProperties
  - Digikam::ThumbnailCreator, [3463](#)
- setLockItem
  - Digikam::GPCamera, [1789](#)
  - Digikam::UMSCamera, [3560](#)
- setLogFilePath
  - Digikam::NREstimate, [2785](#)
- setMacOSEnvironment
  - Digikam, [134](#)
- setMainWidget
  - Digikam::DPopupFrame, [1351](#)
  - Digikam::PanelIconFrame, [2816](#)
- setManualOrder
  - Digikam::ItemInfo, [2305](#)
- setMarkerColor
  - Digikam::DPointSelect, [1346](#)
- setMarkerPixmap
  - Digikam::BackendGoogleMaps, [445](#)
- setMaximumNumberOfThreads
  - Digikam::ActionThreadBase, [221](#)
- setMaxLength
  - Digikam::DPlainTextEdit, [1296](#)
  - Digikam::DTextEdit, [1414](#)
- setMessageType
  - Digikam::DNotificationWidget, [1264](#)
- setMetadataTemplate
  - Digikam::ItemInfo, [2305](#)
- setMetadataWritingMode
  - Digikam::MetaEngine, [2698](#)
- setModDateTime
  - Digikam::ItemInfo, [2305](#)
- setMode
  - Digikam::AssignNameWidget, [408](#)
- setModel
  - Digikam::DConfigDlgView, [941](#)
  - Digikam::SearchTextBarDb, [3129](#)
- setModelsFiltered
  - Digikam::ImportThumbnailBar, [2119](#)
  - Digikam::ItemThumbnailBar, [2425](#)
  - ShowFoto::ShowfotoThumbnailBar, [3751](#)
- setModificationTime
  - Digikam::DFileOperations, [1022](#)
- setName
  - Digikam::DConfigDlgWdgltem, [955](#)
  - Digikam::ItemInfo, [2305](#)
- setNbNeighbors
  - Digikam::OpenCVDNNFaceRecognizer, [2801](#)
- setNeedFileCount
  - Digikam::CollectionScanner, [627](#)
- setNeedResetExifOrientation
  - Digikam::BatchTool, [469](#)
- setNodeId
  - Digikam::KDNodeBase, [2482](#)
- setNoProfileIfEmpty
  - Digikam::IccProfilesComboBox, [1928](#)
- setNoSelectionText
  - Digikam::AlbumSelectComboBox, [333](#)

- setNotificationEnabled
  - Digikam::MaintenanceTool, [2599](#)
- setObserver
  - Digikam::CollectionScanner, [627](#)
  - Digikam::RawProcessingFilter, [2939](#)
- setOnlyLargeThumbnails
  - Digikam::ThumbnailCreator, [3464](#)
- setOpen
  - Digikam::GroupItemFilterSettings, [1871](#)
- setOpenCLEnvironment
  - Digikam, [134](#)
- setOrientation
  - Digikam::ItemInfo, [2306](#)
- setOriginal
  - Digikam::Imageface, [1971](#)
- setOriginalIccProfile
  - Digikam::Imageface, [1971](#)
- setOriginalPos
  - Digikam::DImgChildItem, [1096](#)
- setOutputProfile
  - Digikam::IccTransform, [1944](#)
  - Digikam::RawProcessingFilter, [2940](#)
- setOutputStream
  - Digikam::ExifToolParser, [1523](#)
- setOutputUrl
  - Digikam::BatchTool, [470](#)
- setOutputUrlFromInputUrl
  - Digikam::BatchTool, [470](#)
- setPageWidget
  - Digikam::DConfigDlg, [917](#)
- setParameter
  - Digikam::FaceDetector, [1554](#)
  - Digikam::FacialRecognitionWrapper, [1654](#)
- setParameters
  - Digikam::CoreDbAccess, [723](#)
  - Digikam::CoreDbUrl, [742](#)
  - Digikam::SimilarityDbAccess, [3235](#)
- setParentTag
  - Digikam::AddTagsComboBox, [232](#)
  - Digikam::AddTagsLineEdit, [234](#)
  - Digikam::AssignNameWidget, [408](#)
- setParser
  - Digikam::AdvancedRenameWidget, [248](#)
- setParseString
  - Digikam::AdvancedRenameWidget, [248](#)
- setPassiveMetadataUsage
  - Digikam::LensFunCameraSelector, [2508](#)
- setPerformFastScan
  - Digikam::CollectionScanner, [627](#)
- setPersistent
  - Digikam::PersistentWidgetDelegateOverlay, [2837](#)
- setPickLabel
  - Digikam::ItemInfo, [2306](#)
- setPickLabels
  - Digikam::PickLabelWidget, [2846](#)
- setPixel
  - Digikam::DColor, [899](#)
- setPixmap
  - Digikam::DConfigDlgTitle, [933](#), [934](#)
- setPixmapForNormalSearches
  - Digikam::SearchModel, [3113](#)
- setPixmapRequested
  - Digikam::ThumbnailLoadThread, [3484](#)
- setPixmapsets
  - Digikam::DateAlbumModel, [807](#)
- setPopupMenuEnabled
  - Digikam::DDateTable, [983](#)
- setPopupStyle
  - Digikam::DNotificationPopup, [1256](#)
- setPos
  - Digikam::DImgChildItem, [1096](#)
- setPosition
  - Digikam::DMultiTabBar, [1201](#)
  - Digikam::DMultiTabBarTab, [1209](#)
- setPositiveFilterIsActive
  - Digikam::GPSTiler, [1824](#)
- setPreloadThumbnails
  - Digikam::ItemThumbnailModel, [2440](#)
  - ShowFoto::ShowfotoThumbnailModel, [3763](#)
- setPreloadThumbnailSize
  - ShowFoto::ShowfotoThumbnailModel, [3763](#)
- setPreprocessor
  - Digikam::ItemModel, [2353](#)
- setPreview
  - Digikam::Imageface, [1971](#)
- setPreviewIccProfile
  - Digikam::Imageface, [1971](#)
- setPreviewSize
  - Digikam::Imageface, [1972](#)
- setPreviewType
  - Digikam::Imageface, [1972](#)
- setPriority
  - Digikam::DynamicThread, [1454](#)
  - Digikam::FacePipeline, [1568](#)
  - Digikam::WorkerObject, [3618](#)
- setProgramId
  - Digikam::MetaEngine, [2698](#)
- setProgress
  - Digikam::ProgressItem, [2873](#)
- setProgressMessage
  - Digikam::EditorToolThreaded, [1481](#)
- setProofProfile
  - Digikam::IccTransform, [1944](#)
- setProperty
  - Digikam::DConfigDlgMgr, [923](#)
- setQueryOperationFlag
  - Digikam::BdEngineBackendPrivate, [492](#)
- setRange
  - Digikam::DIntRangeBox, [1132](#)
  - Digikam::DPointSelect, [1346](#)
- setRating
  - Digikam::ItemInfo, [2306](#)
- setRatingEdited
  - Digikam::ItemViewDelegate, [2454](#)
  - Digikam::ItemViewImportDelegate, [2464](#)
- setRawLoadingRules

- Digikam::BatchTool, [470](#)
- setReadOnly
  - Digikam::DDateEdit, [969](#)
- setReadOnlyDrop
  - Digikam::ItemDragDropHandler, [2228](#)
- setReadWithExifTool
  - Digikam::MetaEngine, [2698](#)
- setRectInSceneCoordinates
  - Digikam::DImgChildItem, [1096](#)
- setRecursive
  - Digikam::AbstractCheckableAlbumModel, [169](#)
  - Digikam::AlbumSelectComboBox, [333](#)
  - Digikam::ItemLister, [2323](#)
- setRelativePos
  - Digikam::DImgChildItem, [1096](#)
- setRemoveAlphaChannel
  - Digikam::ThumbnailCreator, [3464](#)
- setRenderXGrid
  - Digikam::HistogramPainter, [1889](#)
- setReplaceNames
  - Digikam::SearchModel, [3113](#)
- setResetExifOrientationAllowed
  - Digikam::BatchTool, [470](#)
- setRestoreCheckState
  - Digikam::AbstractCheckableAlbumTreeView, [176](#)
- setResult
  - Digikam::SharedLoadingTask, [3195](#)
- setRootCheckable
  - Digikam::AbstractCheckableAlbumModel, [169](#)
- setSampleBoxVisible
  - Digikam::DFontProperties, [1032](#)
- setSampleText
  - Digikam::DFontProperties, [1032](#)
- setSaturation
  - Digikam::DColorValueSelector, [907](#)
  - Digikam::DHueSaturationSelector, [1042](#)
- setSaveAsNewVersion
  - Digikam::BatchTool, [470](#)
- setScale
  - Digikam::HistogramPainter, [1889](#)
- setScaleFitToWindow
  - Digikam::SinglePhotoPreviewLayout, [3244](#)
- setScrollBarPolicy
  - Digikam::ImportThumbnailBar, [2119](#)
  - Digikam::ItemThumbnailBar, [2425](#)
  - ShowFoto::ShowfotoThumbnailBar, [3751](#)
- setScrollCurrentToCenter
  - Digikam::ItemViewCategorized, [2447](#)
- setScrollStepGranularity
  - Digikam::ItemViewCategorized, [2447](#)
- setSearchModel
  - Digikam::ChoiceSearchComboBox, [595](#)
- setSearchTextSettings
  - Digikam::AlbumFilterModel, [272](#)
- selectAlbumOnClick
  - Digikam::AbstractAlbumTreeView, [155](#)
- setSelectedCamItemInfos
  - Digikam::ImportCategorizedView, [2018](#)
- setSelectedItemInfos
  - Digikam::ItemCategorizedView, [2189](#)
- setSelectedShowfotoItemInfos
  - ShowFoto::ShowfotoCategorizedView, [3656](#)
- setSelectedUrls
  - Digikam::ImportCategorizedView, [2018](#)
  - Digikam::ItemCategorizedView, [2189](#)
  - ShowFoto::ShowfotoCategorizedView, [3656](#)
- setSelection
  - Digikam::HistogramPainter, [1889](#)
  - Digikam::Imagelface, [1972](#)
- setSelectionArea
  - Digikam::DPreviewImage, [1353](#)
  - Digikam::DPreviewManager, [1357](#)
- setSelectOnContextMenu
  - Digikam::AbstractAlbumTreeView, [155](#)
- setSemanticInfo
  - Digikam::BalooWrap, [460](#)
- setSendItemInfoSignals
  - Digikam::ItemFilterModel, [2247](#)
- setSendRemovalSignals
  - Digikam::ImportItemModel, [2071](#)
  - Digikam::ItemModel, [2353](#)
  - ShowFoto::ShowfotoItemModel, [3700](#)
- setSendSurrogatePixmap
  - Digikam::ThumbnailLoadThread, [3484](#)
- setSetting
  - Digikam::CoreDB, [719](#)
  - Digikam::SimilarityDb, [3233](#)
- setSettings
  - Digikam::BatchTool, [470](#)
  - Digikam::DImgLoaderSettings, [1107](#)
  - Digikam::DMetadataSettings, [1190](#)
  - Digikam::FFmpegLauncher, [1662](#)
  - Digikam::GeolocationSettings, [1778](#)
  - Digikam::IccSettings, [1939](#)
  - Digikam::LocalizeSettings, [2583](#)
  - Digikam::MetaEngineSettings, [2713](#)
  - Digikam::RawProcessingFilter, [2940](#)
- setShallBeShown
  - Digikam::AnimatedClearButton, [375](#)
  - Digikam::ItemVisibilityController, [2475](#)
- setShouldLoaded
  - Digikam::DPlugin, [1301](#)
- setShowAtStart
  - Digikam::ProgressItem, [2873](#)
- setShowCheckStateSummary
  - Digikam::AlbumSelectComboBox, [333](#)
- setShowDeleteFaceTagsAction
  - Digikam::TagFolderView, [3357](#)
- setShowFindDuplicateAction
  - Digikam::TagFolderView, [3357](#)
- setShowfotoItemInfos
  - ShowFoto::ShowfotoItemModel, [3700](#)
- setShowOnHover
  - Digikam::FaceGroup, [1558](#)
- setSignalsEnabled
  - Digikam::CollectionScanner, [627](#)

- setSingleSteps
  - Digikam::CustomStepsDoubleSpinBox, [761](#)
  - Digikam::CustomStepsIntSpinBox, [763](#)
- setSize
  - Digikam::SidebarSplitter, [3223](#)
- setSizeRelative
  - Digikam::DFontProperties, [1032](#)
- setSketchImageFromXML
  - Digikam::SketchWidget, [3246](#)
- setSlave
  - Digikam::DImgThreadedFilter, [1120](#)
- setSoftProofingEnabled
  - Digikam::Canvas, [567](#)
- setSortCategoriesByNaturalComparison
  - Digikam::DCategorizedSortFilterProxyModel, [880](#)
- setSortKey
  - Digikam::MapWidget, [2624](#)
- setSourceAlbumModel
  - Digikam::AlbumFilterModel, [272](#)
- setSourceFilterModel
  - Digikam::AlbumFilterModel, [272](#)
- setSourceModel
  - Digikam::AlbumFilterModel, [272](#)
- setSourceTranscriptionEnabled
  - Digikam::DOnlineTranslator, [1277](#)
- setSourceTranslitEnabled
  - Digikam::DOnlineTranslator, [1277](#)
- setSpacing
  - Digikam::ImportDelegate, [2038](#)
  - Digikam::ItemDelegate, [2214](#)
  - Digikam::ItemViewCategorized, [2447](#)
  - Digikam::ItemViewDelegate, [2454](#)
  - Digikam::ItemViewImportDelegate, [2464](#)
  - ShowFoto::ShowfotoItemViewDelegate, [3709](#)
- setStandardButtons
  - Digikam::DConfigDlg, [918](#)
- setStartingGeographicPoint
  - Digikam::GeodeticCalculator, [1767](#)
- setState
  - Digikam::DMultiTabBarTab, [1209](#)
- setStateSavingDepth
  - Digikam::StateSavingObject, [3263](#)
- setStatus
  - Digikam::ProgressItem, [2873](#)
  - Digikam::TransactionItem, [3537](#)
- setStringComparisonType
  - Digikam::ApplicationSettings, [391](#)
- setStyle
  - Digikam::DMultiTabBar, [1201](#)
  - Digikam::DMultiTabBarTab, [1209](#)
- setSuffix
  - Digikam::DIntRangeBox, [1132](#)
- setSuggestedInitialValue
  - Digikam::CustomStepsDoubleSpinBox, [761](#)
  - Digikam::CustomStepsIntSpinBox, [763](#)
- setSuggestedValues
  - Digikam::CustomStepsDoubleSpinBox, [761](#)
  - Digikam::CustomStepsIntSpinBox, [764](#)
- setSupportingTagModel
  - Digikam::AddTagsLineEdit, [235](#)
  - Digikam::TagCompleter, [3339](#)
- setTab
  - Digikam::DMultiTabBar, [1201](#)
- setTag
  - Digikam::ItemInfo, [2306](#)
- setTagIcon
  - Digikam::CoreDB, [719](#)
- setTagList
  - Digikam::GPSItemContainer, [1805](#)
- setTagName
  - Digikam::CoreDB, [719](#)
- setTagParentID
  - Digikam::CoreDB, [720](#)
- setTagSelected
  - Digikam::AlbumSelectors, [343](#)
- setTagTreeView
  - Digikam::AddTagsLineEdit, [235](#)
- setText
  - Digikam::DConfigDlgTitle, [934](#)
  - Digikam::DCursorTracker, [966](#)
  - Digikam::DNotificationWidget, [1265](#)
- setThreshold
  - Digikam::OpenCVDNNFaceRecognizer, [2801](#)
- setThumbnail
  - Digikam::ProgressItem, [2874](#)
- setThumbnailCacheSize
  - Digikam::LoadingCache, [2548](#)
- setThumbnailInfoProvider
  - Digikam::ThumbnailCreator, [3464](#)
- setThumbnailLoadThread
  - Digikam::ItemThumbnailModel, [2440](#)
  - ShowFoto::ShowfotoThumbnailModel, [3763](#)
- setThumbnailSize
  - Digikam::AlbumThumbnailLoader, [358](#)
  - Digikam::DigikamItemView, [1064](#)
  - Digikam::DItemDelegate, [1153](#)
  - Digikam::ImportIconView, [2063](#)
  - Digikam::ItemViewDelegate, [2454](#)
  - Digikam::ItemViewImportDelegate, [2464](#)
  - Digikam::ThumbnailCreator, [3464](#)
  - Digikam::ThumbnailLoadThread, [3484](#)
  - Digikam::TrashView, [3543](#)
  - ShowFoto::ShowfotoItemViewDelegate, [3709](#)
  - ShowFoto::ShowfotoThumbnailModel, [3763](#)
- setThumbnailSize
  - Digikam::MapWidget, [2624](#)
- setTiffThumbnail
  - Digikam::MetaEngine, [2698](#)
- setTimeout
  - Digikam::DNotificationPopup, [1256](#)
- settings
  - Digikam::DImgLoaderSettings, [1107](#)
  - Digikam::DMetadataSettings, [1191](#)
  - Digikam::GeolocationSettings, [1778](#)
  - Digikam::IccSettings, [1939](#)
  - Digikam::LocalizeSettings, [2583](#)

- Digikam::MetaEngineSettings, [2713](#)
- Digikam::NREstimate, [2785](#)
- settingsChanged
  - Digikam::DConfigDlgMngr, [923](#)
- SettingsType
  - Digikam::ImageQualityConfSelector, [1979](#)
- settingsWidget
  - Digikam::BatchTool, [470](#)
- setTitle
  - Digikam::AltLangStrEdit, [371](#)
- setTitleWidget
  - Digikam::AltLangStrEdit, [372](#)
- setToolDescription
  - Digikam::BatchTool, [471](#)
- setToolIconName
  - Digikam::BatchTool, [471](#)
- setToolTitle
  - Digikam::BatchTool, [471](#)
- setTranslationOptionsEnabled
  - Digikam::DOnlineTranslator, [1277](#)
- setTranslationTranslitEnabled
  - Digikam::DOnlineTranslator, [1277](#)
- setTreeView
  - Digikam::AbstractAlbumTreeViewSelectComboBox, [162](#)
- setTristate
  - Digikam::AbstractCheckableAlbumModel, [169](#)
- setType
  - Digikam::FocusPoint, [1730](#)
- setTypeSelection
  - Digikam::AlbumSelectors, [343](#)
- setUIEnabled
  - Digikam::RGWidget, [2990](#)
- setUndoImg
  - Digikam::EditorCore, [1468](#)
- setUniqueBehavior
  - Digikam::ItemComments, [2197](#)
- setUnpairedImages
  - Digikam::EmptyImageListProvider, [1505](#)
  - Digikam::QListImageListProvider, [2891](#)
- setup
  - Digikam::AbstractCountingAlbumModel, [182](#)
  - Digikam::DPlugin, [1301](#)
  - Digikam::ItemViewHoverButton, [2457](#)
  - Digikam::MetadataWidget, [2666](#)
- setupAndStartDirectly
  - Digikam::DImgThreadedFilter, [1120](#)
- SetupCollectionDataRole
  - Digikam::SetupCollectionModel, [3171](#)
- SetupCollectionModel
  - Digikam::SetupCollectionModel, [3172](#)
- setUpdateFileTimeStamp
  - Digikam::MetaEngine, [2699](#)
- setupFilter
  - Digikam::DImgThreadedFilter, [1120](#)
- SetupICC
  - Digikam::SetupICC, [3179](#)
- setupIconTheme
  - Digikam::DXmlGuiWindow, [1448](#)
- setupKeywords
  - Digikam::DDateEdit, [969](#)
- setupValueWidgets
  - Digikam::SearchFieldAlbum, [3041](#)
  - Digikam::SearchFieldCheckBox, [3045](#)
  - Digikam::SearchFieldChoice, [3049](#)
  - Digikam::SearchFieldColorDepth, [3053](#)
  - Digikam::SearchFieldComboBox, [3056](#)
  - Digikam::SearchFieldLabels, [3065](#)
  - Digikam::SearchFieldMonthDay, [3069](#)
  - Digikam::SearchFieldPageOrientation, [3073](#)
  - Digikam::SearchFieldRangeDate, [3076](#)
  - Digikam::SearchFieldRangeDouble, [3080](#)
  - Digikam::SearchFieldRangeInt, [3084](#)
  - Digikam::SearchFieldRangeTime, [3088](#)
  - Digikam::SearchFieldRating, [3092](#)
  - Digikam::SearchFieldText, [3096](#)
- setupWidget
  - Digikam::DConfigDlgMngr, [924](#)
- setUseCompatibleFileName
  - Digikam::MetaEngine, [2699](#)
- setUsedByLabelsTree
  - Digikam::Album, [261](#)
- setUseManagedView
  - Digikam::lccSettings, [1939](#)
- setUseMultiCoreCPU
  - Digikam::AutotagsAssignment, [431](#)
  - Digikam::DbCleaner, [833](#)
  - Digikam::FingerPrintsGenerator, [1726](#)
  - Digikam::ImageQualitySorter, [1986](#)
  - Digikam::MaintenanceTool, [2599](#)
  - Digikam::MetadataRemover, [2653](#)
  - Digikam::MetadataSynchronizer, [2662](#)
  - Digikam::ThumbsGenerator, [3498](#)
- setUsePointingHandCursor
  - Digikam::ItemViewCategorized, [2447](#)
- setUserData
  - Digikam::AssignNameWidget, [409](#)
- setUserFilterSettings
  - Digikam::CoreDB, [720](#)
- setUsesBusyIndicator
  - Digikam::ProgressItem, [2874](#)
- setUseTokenMenu
  - Digikam::Rule, [2999](#)
- setUseXMPSidecar4Reading
  - Digikam::MetaEngine, [2699](#)
- setValue
  - Digikam::DbEngineActionType, [835](#)
- setValues
  - Digikam::DPointSelect, [1347](#)
- setValueWidgetsVisible
  - Digikam::SearchFieldAlbum, [3041](#)
  - Digikam::SearchFieldCheckBox, [3045](#)
  - Digikam::SearchFieldChoice, [3049](#)
  - Digikam::SearchFieldComboBox, [3056](#)
  - Digikam::SearchFieldLabels, [3065](#)
  - Digikam::SearchFieldMonthDay, [3069](#)

- Digikam::SearchFieldRangeDate, [3076](#)
- Digikam::SearchFieldRangeDouble, [3080](#)
- Digikam::SearchFieldRangeInt, [3084](#)
- Digikam::SearchFieldRangeTime, [3088](#)
- Digikam::SearchFieldRating, [3092](#)
- Digikam::SearchFieldText, [3096](#)
- setVersionItemFilterSettings
  - Digikam::ItemFilterModel, [2247](#)
- setView
  - Digikam::DNotificationPopup, [1256](#)
- setViewportRect
  - Digikam::RegionFrameItem, [2963](#)
- setVisible
  - Digikam::DPlugin, [1301](#)
  - Digikam::DPluginBqm, [1309](#)
  - Digikam::DPluginDImg, [1326](#)
  - Digikam::DPluginEditor, [1330](#)
  - Digikam::DPluginGeneric, [1334](#)
  - Digikam::DPluginRawImport, [1343](#)
  - Digikam::FaceGroup, [1558](#)
  - Digikam::FocusPointGroup, [1732](#)
  - Digikam::ItemInfo, [2306](#)
  - Digikam::SearchField, [3038](#)
- setWaitingDataProgress
  - Digikam::DRawDecoder, [1373](#)
- setWatchDisabled
  - Digikam::CollectionManager, [617](#)
- setWatchFlags
  - Digikam::ItemModel, [2353](#)
- setWidget
  - Digikam::DConfigDlgTitle, [935](#)
  - Digikam::ProxyLineEdit, [2888](#)
- setWindowsEnvironment
  - Digikam, [134](#)
- setWordWrap
  - Digikam::DNotificationWidget, [1265](#)
- setWorkingUrl
  - Digikam::BatchTool, [471](#)
- setWriteDngFiles
  - Digikam::MetaEngine, [2699](#)
- setWriteRawFiles
  - Digikam::MetaEngine, [2699](#)
- setWriteWithExifTool
  - Digikam::MetaEngine, [2699](#)
- setXmp
  - Digikam::MetaEngine, [2699](#)
- setXmpKeywords
  - Digikam::DMetadata, [1189](#)
  - Digikam::MetaEngine, [2700](#)
- setXmpSubCategories
  - Digikam::DMetadata, [1189](#)
  - Digikam::MetaEngine, [2700](#)
- setXmpSubjects
  - Digikam::DMetadata, [1189](#)
  - Digikam::MetaEngine, [2700](#)
- setXmpTagString
  - Digikam::MetaEngine, [2700](#)
- setXmpTagStringBag
  - Digikam::MetaEngine, [2700](#)
- setXmpTagStringLangAlt
  - Digikam::MetaEngine, [2701](#)
- setXmpTagStringListLangAlt
  - Digikam::MetaEngine, [2701](#)
- setXmpTagStringSeq
  - Digikam::MetaEngine, [2701](#)
- setXValue
  - Digikam::DPointSelect, [1347](#)
- setYCbCr
  - Digikam::DColor, [900](#)
- setYValue
  - Digikam::DPointSelect, [1347](#)
- setZoom
  - Digikam::BackendGoogleMaps, [445](#)
  - Digikam::BackendMarble, [453](#)
- setZoomFactor
  - Digikam::ImageZoomSettings, [2009](#)
- SFace
  - Digikam::FaceScanSettings, [1624](#)
- shade
  - Digikam::SchemeManager, [3027](#), [3028](#)
- ShadeRole
  - Digikam::SchemeManager, [3024](#)
- ShadowShade
  - Digikam::SchemeManager, [3024](#)
- SharpenFilter
  - Digikam::SharpenFilter, [3204](#)
- shortenedMakeInfo
  - Digikam::ItemPropertiesTab, [2380](#)
- shortenedTagPaths
  - Digikam::ItemPropertiesTab, [2380](#)
  - Digikam::TagsCache, [3403](#)
- shortestDistancesFrom
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1843](#)
- shortestPath
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1843](#)
  - Digikam::Graph< VertexProperties, EdgeProperties >::Path, [1850](#)
- shouldBeVisible
  - Digikam::ThumbBarDock, [3460](#)
- shouldLoaded
  - Digikam::DPlugin, [1302](#)
- show
  - Digikam::DNotificationPopup, [1257](#)
  - Digikam::ItemViewToolTip, [2468](#)
  - Digikam::ItemVisibilityController, [2475](#)
- showAnimationFinished
  - Digikam::DNotificationWidget, [1265](#)
- showAtStart
  - Digikam::ProgressItem, [2874](#)
- showContextMenu
  - Digikam::DigikamItemView, [1064](#)
  - Digikam::ImportIconView, [2063](#)
- showContextMenuAt
  - Digikam::AbstractAlbumTreeView, [156](#)

- showContextMenuOnIndex
  - Digikam::ImportCategorizedView, 2019
  - Digikam::ItemCategorizedView, 2189
  - Digikam::ItemViewCategorized, 2448
  - ShowFoto::ShowfotoCategorizedView, 3656
- showContextMenuOnInfo
  - Digikam::DigikamItemView, 1065
  - Digikam::ImportIconView, 2063
- showContinueCancel
  - Digikam::DMessageBox, 1170
- showContinueCancelList
  - Digikam::DMessageBox, 1171
- showContinueCancelWidget
  - Digikam::DMessageBox, 1171
- ShowCountAccordingToSettings
  - Digikam::AbstractAlbumTreeView, 150
- ShowFoto::NoDuplicatesShowfotoFilterModel, 3640
- ShowFoto::Showfoto, 3643
  - infoface, 3648
- ShowFoto::Showfoto::Private, 3648
- ShowFoto::ShowfotoCategorizedView, 3649
  - addOverlay, 3654
  - deselected, 3654
  - dragDropHandler, 3654
  - filterModel, 3654
  - hintAt, 3654
  - indexActivated, 3654
  - modelChanged, 3654
  - nextIndexHint, 3655
  - nextInOrder, 3655
  - selected, 3655
  - setCurrentInfo, 3655
  - setCurrentUrl, 3655
  - setCurrentWhenAvailable, 3655
  - setSelectedShowfotoItemInfos, 3656
  - setSelectedUrls, 3656
  - showContextMenuOnIndex, 3656
  - showfotoFilterModel, 3656
  - showfotoItemInfoActivated, 3656
  - showfotoThumbnailModel, 3656
  - toIndex, 3656
- ShowFoto::ShowfotoCoordinatesOverlay, 3657
  - checkIndex, 3659
  - createWidget, 3659
  - setActive, 3659
  - slotEntered, 3660
  - visualChange, 3660
- ShowFoto::ShowfotoCoordinatesOverlayWidget, 3660
- ShowFoto::ShowfotoDelegate, 3662
  - acceptsActivation, 3665
  - acceptsToolTip, 3665
  - clearCaches, 3666
  - clearModelDataCaches, 3666
  - imageInformationRect, 3666
  - pixmapForDrag, 3666
  - pixmapRect, 3666
  - retrieveThumbnailPixmap, 3666
  - setDefaultViewOptions, 3667
  - updateContentWidth, 3667
  - updateRects, 3667
  - updateSizeRectsAndPixmap, 3667
- ShowFoto::ShowfotoDelegate::ShowfotoDelegatePrivate, 3668
  - clearRects, 3669
- ShowFoto::ShowfotoDragDropHandler, 3670
  - accepts, 3671
  - createMimeData, 3671
  - dropEvent, 3671
  - mimeTypes, 3671
- ShowFoto::ShowfotoFilterModel, 3673
  - CategorizationModeRole, 3676
  - CategoryFormatRole, 3676
  - categoryIdentifier, 3676
  - compareCategories, 3676
  - compareInfosCategories, 3676
  - infosLessThan, 3677
  - setDirectSourceShowfotoModel, 3677
  - showfotoFilterModel, 3677
  - ShowfotoFilterModelPointerRole, 3676
  - ShowfotoFilterModelRoles, 3675
  - showfotoItemInfosAdded, 3677
  - SortOrderRole, 3676
  - subSortLessThan, 3677
- ShowFoto::ShowfotoFolderViewBar, 3678
- ShowFoto::ShowfotoFolderViewBookmarkDlg, 3680
- ShowFoto::ShowfotoFolderViewBookmarkItem, 3681
- ShowFoto::ShowfotoFolderViewBookmarkList, 3682
- ShowFoto::ShowfotoFolderViewBookmarks, 3683
- ShowFoto::ShowfotoFolderViewList, 3684
  - FileDate, 3685
  - FolderViewRole, 3684
- ShowFoto::ShowfotoFolderViewModel, 3685
  - currentFilePath, 3685
- ShowFoto::ShowfotoFolderViewSideBar, 3686
  - doLoadState, 3688
  - doSaveState, 3688
- ShowFoto::ShowfotoFolderViewToolTip, 3689
- ShowFoto::ShowfotoFolderViewUndo, 3690
- ShowFoto::ShowfotoInfoface, 3691
  - openSetupPage, 3693
- ShowFoto::ShowfotoItemInfo, 3693
  - isNull, 3694
  - operator==, 3694
  - size, 3694
- ShowFoto::ShowfotoItemModel, 3695
  - addShowfotoItemInfoSynchronously, 3698
  - allRefreshingFinished, 3698
  - clearShowfotoItemInfos, 3698
  - ExtraDataDuplicateCount, 3698
  - ExtraDataRole, 3698
  - indexForShowfotoItemInfo, 3698
  - indexForUrl, 3698
  - itemInfosAboutToBeAdded, 3698
  - itemInfosAboutToBeRemoved, 3699
  - itemInfosAdded, 3699
  - itemInfosRemoved, 3699



- preprocess, [3699](#)
- readyForIncrementalRefresh, [3699](#)
- removeIndex, [3699](#)
- requestIncrementalRefresh, [3700](#)
- retrieveShowfotoItemInfo, [3700](#)
- rowCount, [3700](#)
- setKeepsFileUrlCache, [3700](#)
- setSendRemovalSignals, [3700](#)
- setShowfotoItemInfos, [3700](#)
- showfotoItemInfo, [3700](#), [3701](#)
- showfotoItemInfosAboutToBeRemoved, [3701](#)
- showfotoItemInfosCleared, [3701](#)
- ShowfotoItemModelPointerRole, [3698](#)
- ShowfotoItemModelRoles, [3697](#)
- startIncrementalRefresh, [3701](#)
- ThumbnailRole, [3698](#)
- ShowFoto::ShowfotoItemSortSettings, [3701](#)
  - compare, [3703](#)
  - compareByOrder, [3703](#)
  - compareCategories, [3703](#)
  - compareValue, [3703](#)
  - currentCategorizationSortOrder, [3704](#)
  - DefaultOrder, [3702](#)
  - lessThan, [3703](#)
  - lessThanByOrder, [3704](#)
  - naturalCompare, [3704](#)
  - SortOrder, [3702](#)
- ShowFoto::ShowfotoItemViewDelegate, [3705](#)
  - acceptsActivation, [3708](#)
  - acceptsToolTip, [3708](#)
  - asDelegate, [3708](#)
  - gridSize, [3708](#)
  - imageInformationRect, [3708](#)
  - mouseMoved, [3708](#)
  - pixmapRect, [3709](#)
  - setDefaultViewOptions, [3709](#)
  - setSpacing, [3709](#)
  - setThumbnailSize, [3709](#)
- ShowFoto::ShowfotoItemViewDelegatePrivate, [3710](#)
  - clearRects, [3711](#)
- ShowFoto::ShowfotoKineticScroller, [3712](#)
  - enableKineticScrollFor, [3712](#)
- ShowFoto::ShowfotoNormalDelegate, [3713](#)
  - updateRects, [3717](#)
- ShowFoto::ShowfotoNormalDelegatePrivate, [3718](#)
- ShowFoto::ShowfotoSettings, [3720](#)
- ShowFoto::ShowfotoSetup, [3722](#)
  - execSinglePage, [3724](#)
- ShowFoto::ShowfotoSetupMetadata, [3725](#)
- ShowFoto::ShowfotoSetupMisc, [3726](#)
- ShowFoto::ShowfotoSetupPlugins, [3727](#)
- ShowFoto::ShowfotoSetupRaw, [3728](#)
- ShowFoto::ShowfotoSetupToolTip, [3729](#)
- ShowFoto::ShowfotoSortFilterModel, [3730](#)
  - mapToSourceShowfotoModel, [3732](#)
  - setDirectSourceShowfotoModel, [3732](#)
  - showfotoFilterModel, [3732](#)
  - showfotoItemInfosSorted, [3732](#)
- ShowFoto::ShowfotoStackViewFavoriteItem, [3733](#)
  - FavoriteFolder, [3734](#)
  - FavoriteItem, [3734](#)
  - FavoriteRoot, [3734](#)
  - FavoriteType, [3734](#)
  - hierarchyFromParent, [3734](#)
  - urlsToPaths, [3734](#)
- ShowFoto::ShowfotoStackViewFavoriteItemDlg, [3735](#)
- ShowFoto::ShowfotoStackViewFavoriteList, [3736](#)
  - filter, [3737](#)
  - setFilter, [3737](#)
  - signalSearchResult, [3737](#)
- ShowFoto::ShowfotoStackViewFavorites, [3738](#)
- ShowFoto::ShowfotoStackViewItem, [3739](#)
- ShowFoto::ShowfotoStackViewList, [3740](#)
  - FileDate, [3741](#)
  - StackViewRole, [3741](#)
- ShowFoto::ShowfotoStackViewSideBar, [3742](#)
  - doLoadState, [3743](#)
  - doSaveState, [3743](#)
- ShowFoto::ShowfotoStackViewToolTip, [3744](#)
- ShowFoto::ShowfotoThumbnailBar, [3746](#)
  - setModelsFiltered, [3751](#)
  - setScrollBarPolicy, [3751](#)
- ShowFoto::ShowfotoThumbnailDelegate, [3752](#)
  - acceptsActivation, [3756](#)
  - maximumSize, [3756](#)
  - setDefaultViewOptions, [3756](#)
  - updateContentWidth, [3756](#)
  - updateRects, [3756](#)
- ShowFoto::ShowfotoThumbnailDelegatePrivate, [3757](#)
- ShowFoto::ShowfotoThumbnailModel, [3759](#)
  - data, [3763](#)
  - setData, [3763](#)
  - setEmitDataChanged, [3763](#)
  - setPreloadThumbnails, [3763](#)
  - setPreloadThumbnailSize, [3763](#)
  - setThumbnailLoadThread, [3763](#)
  - setThumbnailSize, [3763](#)
  - showfotoItemInfosCleared, [3764](#)
  - ShowfotoThumbnailModel, [3762](#)
- showfotoFilterModel
  - ShowFoto::ShowfotoCategorizedView, [3656](#)
  - ShowFoto::ShowfotoFilterModel, [3677](#)
  - ShowFoto::ShowfotoSortFilterModel, [3732](#)
- ShowfotoFilterModelPointerRole
  - ShowFoto::ShowfotoFilterModel, [3676](#)
- ShowfotoFilterModelRoles
  - ShowFoto::ShowfotoFilterModel, [3675](#)
- showfotoItemInfo
  - ShowFoto::ShowfotoItemModel, [3700](#), [3701](#)
- showfotoItemInfoActivated
  - ShowFoto::ShowfotoCategorizedView, [3656](#)
- showfotoItemInfosAboutToBeRemoved
  - ShowFoto::ShowfotoItemModel, [3701](#)
- showfotoItemInfosAdded
  - ShowFoto::ShowfotoFilterModel, [3677](#)
- showfotoItemInfosCleared

- ShowFoto::ShowfotoItemModel, [3701](#)
- ShowFoto::ShowfotoThumbnailModel, [3764](#)
- showfotoItemInfosSorted
  - ShowFoto::ShowfotoSortFilterModel, [3732](#)
- ShowfotoItemModelPointerRole
  - ShowFoto::ShowfotoItemModel, [3698](#)
- ShowfotoItemModelRoles
  - ShowFoto::ShowfotoItemModel, [3697](#)
- ShowfotoThumbnailModel
  - ShowFoto::ShowfotoThumbnailModel, [3762](#)
- showfotoThumbnailModel
  - ShowFoto::ShowfotoCategorizedView, [3656](#)
- showInformationList
  - Digikam::DMessageBox, [1171](#)
- showInformationWidget
  - Digikam::DMessageBox, [1171](#)
- showItem
  - Digikam::ItemVisibilityController, [2475](#)
- showOnIndex
  - Digikam::AssignNameOverlay, [403](#)
  - Digikam::PersistentWidgetDelegateOverlay, [2837](#)
- showPageHeader
  - Digikam::DConfigDlgView, [941](#)
- showRawCameraList
  - Digikam, [134](#)
- showSideBars
  - Digikam::DXmlGuiWindow, [1448](#)
- showThumbBar
  - Digikam::DXmlGuiWindow, [1448](#)
- showToolTip
  - Digikam::ItemViewCategorized, [2448](#)
- showYesNo
  - Digikam::DMessageBox, [1171](#)
- showYesNoList
  - Digikam::DMessageBox, [1172](#)
- showYesNoWidget
  - Digikam::DMessageBox, [1172](#)
- shrink
  - Digikam::Sidebar, [3219](#)
- shutDown
  - Digikam::DynamicThread, [1454](#)
  - Digikam::FacePipeline, [1569](#)
  - Digikam::ScanController, [3014](#)
  - Digikam::WorkerObject, [3619](#)
- shutDownExifTool
  - Digikam::ExifToolProcess, [1529](#)
- Sidebar
  - Digikam::Sidebar, [3216](#)
- SidebarSplitter
  - Digikam::SidebarSplitter, [3222](#)
- SidebarWidget
  - Digikam::SidebarWidget, [3225](#)
- sidecarFilePathForFile
  - Digikam::MetaEngine, [2701](#)
- sidecarFiles
  - Digikam::DFileOperations, [1022](#)
- sidecarPath
  - Digikam::MetaEngine, [2701](#)
- sidecarUrl
  - Digikam::MetaEngine, [2701](#), [2702](#)
- sig
  - Digikam::Haar::SignatureData, [1873](#)
- signalAddItemsToProcess
  - Digikam::DatabaseTask, [793](#)
  - Digikam::MaintenanceThread, [2596](#)
- signalAdvance
  - Digikam::MaintenanceThread, [2596](#)
- signalAlbumAboutToBeAdded
  - Digikam::AlbumManager, [301](#)
- signalAlbumAboutToBeDeleted
  - Digikam::AlbumManager, [301](#)
- signalAlbumAboutToBeMoved
  - Digikam::AlbumManager, [301](#)
- signalAlbumAdded
  - Digikam::AlbumManager, [301](#)
- signalAlbumDeleted
  - Digikam::AlbumManager, [301](#)
- signalAlbumHasBeenDeleted
  - Digikam::AlbumManager, [301](#)
- signalAlbumMoved
  - Digikam::AlbumManager, [302](#)
- signalAssignSettings2Widget
  - Digikam::BatchTool, [471](#)
- signalCancelActionTask
  - Digikam::ActionThread, [218](#)
- signalCanceled
  - Digikam::MaintenanceThread, [2596](#)
  - Digikam::MaintenanceTool, [2599](#)
- signalChangedTab
  - Digikam::Sidebar, [3219](#)
- signalClicked
  - Digikam::DMultiTabBarButton, [1204](#)
- signalComplete
  - Digikam::MaintenanceTool, [2600](#)
- signalCompleted
  - Digikam::MaintenanceThread, [2596](#)
- signalData
  - Digikam::MaintenanceThread, [2596](#)
- signalDecodedImage
  - Digikam::DPluginRawImport, [1343](#)
- signalDispatchThumbnailInternal
  - Digikam::AlbumThumbnailLoader, [358](#)
- signalDone
  - Digikam::ActionJob, [212](#)
- signalFailed
  - Digikam::AlbumThumbnailLoader, [358](#)
- signalFileMetadataChanged
  - Digikam::ItemAttributesWatch, [2180](#)
- signalFilterChanged
  - Digikam::AlbumFilterModel, [273](#)
- signalFindDuplicates
  - Digikam::AlbumSelectionTreeView, [339](#)
- signalFinished
  - Digikam::ActionThread, [218](#)
  - Digikam::DOnlineTranslator, [1277](#)
  - Digikam::MaintenanceThread, [2596](#)

- signalImageLoaded
  - Digikam::LoadSaveThread, [2572](#)
- signalImageRatingChanged
  - Digikam::ItemAttributesWatch, [2180](#)
- signalImagesChanged
  - Digikam::ItemAttributesWatch, [2181](#)
- signalImageStartedLoading
  - Digikam::LoadSaveThread, [2572](#)
- signalImageTagsChanged
  - Digikam::ItemAttributesWatch, [2181](#)
- signalLoadingProgress
  - Digikam::LoadSaveThread, [2572](#)
- signalLoadRaw
  - Digikam::DPluginRawImport, [1343](#)
- signalModified
  - Digikam::AltLangStrEdit, [372](#)
- signalMoreCompleteLoadingAvailable
  - Digikam::LoadSaveThread, [2572](#)
- signalNotificationError
  - Digikam::SidebarWidget, [3227](#)
- signalProgress
  - Digikam::ActionJob, [212](#)
- signalProgressCanceled
  - Digikam::DProgressWdg, [1360](#)
- signalProgressChanged
  - Digikam::RGWidget, [2990](#)
- signalProgressSetup
  - Digikam::RGWidget, [2991](#)
- signalQueueProcessed
  - Digikam::ActionThread, [218](#)
- signalReloadThumbnails
  - Digikam::AlbumThumbnailLoader, [358](#)
- signalRemovePending
  - Digikam::MaintenanceThread, [2597](#)
- signalSearchResult
  - Digikam::DPluginConfView, [1312](#)
  - ShowFoto::ShowfotoStackViewFavoriteList, [3737](#)
- signalSelectionChanged
  - Digikam::AltLangStrEdit, [372](#)
- signalSelectionMoved
  - Digikam::PanIconWidget, [2818](#)
- signalSettingsChanged
  - Digikam::DImgLoaderSettings, [1107](#)
- signalSetUIEnabled
  - Digikam::RGWidget, [2991](#)
- signalShowOnlyAvailableAlbumsChanged
  - Digikam::AlbumManager, [302](#)
- signalStarted
  - Digikam::ActionJob, [212](#)
  - Digikam::MaintenanceThread, [2597](#)
- signalStarting
  - Digikam::ActionThread, [218](#)
- signalTagFilterChanged
  - Digikam::FilterSideBarWidget, [1717](#)
- signalThumbnail
  - Digikam::AlbumThumbnailLoader, [358](#)
- signalTokenTriggered
  - Digikam::Token, [3511](#)
- signalUndoCommand
  - Digikam::RGWidget, [2991](#)
- signalValueAdded
  - Digikam::AltLangStrEdit, [372](#)
- signalValueDeleted
  - Digikam::AltLangStrEdit, [372](#)
- signalViewChanged
  - Digikam::Sidebar, [3219](#)
- signatureAsText
  - Digikam::Haarface, [1879](#)
- SimilarityDbAccess
  - Digikam::SimilarityDbAccess, [3234](#)
- similarityParameters
  - Digikam::DbEngineParameters, [846](#)
- similarityTo
  - Digikam::ItemExtendedProperties, [2230](#)
- SimpleFiltering
  - Digikam::AlbumFilterModel, [269](#)
- SimpleResize
  - Digikam::GreycstorationFilter, [1861](#)
- singleGroupMainItem
  - Digikam::ItemInfoList, [2314](#)
- singleItem
  - Digikam::ProgressManager, [2881](#)
- sixteenBitsImage
  - Digikam::DRawDecoderSettings, [1381](#)
- Size
  - Digikam::ThumbnailSize, [3486](#)
- size
  - Digikam::CamItemInfo, [557](#)
  - Digikam::DownloadInfo, [1291](#)
  - Digikam::EmptyImageListProvider, [1505](#)
  - Digikam::QListImageListProvider, [2891](#)
  - Digikam::SidebarSplitter, [3223](#)
  - ShowFoto::ShowfotoItemInfo, [3694](#)
- sizeHint
  - Digikam::DDatePicker, [974](#)
  - Digikam::DDateTable, [983](#)
  - Digikam::DFontProperties, [1032](#)
  - Digikam::DNotificationWidget, [1266](#)
  - Digikam::FaceRejectionOverlayButton, [1621](#)
  - Digikam::ImportRotateOverlayButton, [2105](#)
  - Digikam::ItemFullScreenOverlayButton, [2267](#)
  - Digikam::ItemRotateOverlayButton, [2395](#)
  - Digikam::ItemSelectionOverlayButton, [2408](#)
  - Digikam::ItemViewHoverButton, [2457](#)
  - Digikam::SqueezedComboBox, [3256](#)
  - Digikam::TableViewColumn, [3287](#)
  - Digikam::TableViewColumns::ColumnThumbnail, [3320](#)
  - Digikam::TableViewItemDelegate, [3322](#)
- sizelsRelative
  - Digikam::DFontProperties, [1032](#)
- sketchImageToXML
  - Digikam::SketchWidget, [3246](#)
- Skip
  - Digikam::FaceScanSettings, [1624](#)
- SkipAlreadyScanned

- Digikam::FacePipeline, 1566
- slotAlbumDelete
  - Digikam::AlbumModificationHelper, 317
- slotAlbumEdit
  - Digikam::AlbumModificationHelper, 317
- slotAlbumNew
  - Digikam::AlbumModificationHelper, 317
- slotAlbumRename
  - Digikam::AlbumModificationHelper, 317
- slotAnalyserStarted
  - Digikam::EditorToolThreaded, 1481
- slotAppendPressed
  - Digikam::SetupCollectionModel, 3172
- slotApplicationSettingsChanged
  - Digikam::LightTableWindow, 2537
- slotAssignSettings2Widget
  - Digikam::BatchTool, 471
- slotAwayFromSelection
  - Digikam::TableView, 3285
- slotCategoryButtonPressed
  - Digikam::SetupCollectionModel, 3172
- slotClearHighlight
  - Digikam::DPreviewImage, 1353
- slotClustersClicked
  - Digikam::MapWidget, 2624
- slotClustersMoved
  - Digikam::MapWidget, 2624
- slotCreateFuzzySearchFromDropped
  - Digikam::SearchModificationHelper, 3116
- slotCreateFuzzySearchFromImage
  - Digikam::SearchModificationHelper, 3117
- slotCreateFuzzySearchFromSketch
  - Digikam::SearchModificationHelper, 3117
- slotCreateTimeLineSearch
  - Digikam::SearchModificationHelper, 3118
- slotDateTimeForUrl
  - Digikam::DInfoInterface, 1128
  - Digikam::DMetaInfoface, 1195
- slotDeleteSelected
  - Digikam::TableView, 3285
- slotEnabledInternalWidgets
  - Digikam::AltLangStrEdit, 372
- slotEntered
  - Digikam::AbstractWidgetDelegateOverlay, 203
  - Digikam::GroupIndicatorOverlay, 1868
  - Digikam::ImportCoordinatesOverlay, 2031
  - Digikam::ImportDownloadOverlay, 2045
  - Digikam::ImportLockOverlay, 2082
  - Digikam::ImportRatingOverlay, 2097
  - Digikam::ItemCoordinatesOverlay, 2202
  - Digikam::ItemRatingOverlay, 2388
  - Digikam::PersistentWidgetDelegateOverlay, 2837
  - Digikam::TagsLineEditOverlay, 3415
  - ShowFoto::ShowfotoCoordinatesOverlay, 3660
- slotFaceTagDelete
  - Digikam::TagModificationHelper, 3381, 3382
- slotFilterStarted
  - Digikam::EditorToolThreaded, 1481
- slotFitToWindow
  - Digikam::ItemIconView, 2290
- slotImageChange
  - Digikam::ItemAlbumModel, 2179
- slotImageQualitySorter
  - Digikam::ItemIconView, 2291
- slotItemDisplaySettingsChanged
  - Digikam::MapWidget, 2624
- slotLayoutChanged
  - Digikam::PersistentWidgetDelegateOverlay, 2837
- slotMetadataChangedForUrl
  - Digikam::DInfoInterface, 1129
  - Digikam::DMetaInfoface, 1195
- slotMouseModeChanged
  - Digikam::MapWidget, 2625
- slotMultipleFaceTagDel
  - Digikam::TagModificationHelper, 3382
- slotMultipleTagDel
  - Digikam::TagModificationHelper, 3382
- slotMultipleTagsToFaceTags
  - Digikam::TagModificationHelper, 3383
- slotNewAlbum
  - Digikam::AlbumSelectTreeView, 350
- slotNewModelData
  - Digikam::GPSMarkerTiler, 1824
- slotNewSelectionFromMap
  - Digikam::MapWidget, 2625
- slotProgress
  - Digikam::EditorToolThreaded, 1481
- slotRemoveTag
  - Digikam::ItemIconView, 2291
- slotReset
  - Digikam::AbstractWidgetDelegateOverlay, 203
  - Digikam::PersistentWidgetDelegateOverlay, 2837
- slotResetFilters
  - Digikam::FilterSideBarWidget, 1718
- slotReturnPressed
  - Digikam::AddTagsLineEdit, 235
- slotRootAlbumAvailable
  - Digikam::AbstractAlbumTreeView, 156
- slotRowsRemoved
  - Digikam::PersistentWidgetDelegateOverlay, 2838
- slotScheduleUpdate
  - Digikam::BackendMarble, 454
- slotSearchDelete
  - Digikam::SearchModificationHelper, 3118
- slotSearchRename
  - Digikam::SearchModificationHelper, 3118
- slotSetCurrentWhenAvailable
  - Digikam::TableView, 3285
- slotSetHighlightArea
  - Digikam::DPreviewImage, 1354
- slotSetHighlightShown
  - Digikam::DPreviewImage, 1354
- slotSetSelection
  - Digikam::DPreviewImage, 1354
- slotSetupChanged
  - Digikam::DigikamItemView, 1065

- Digikam::ImportIconView, [2063](#)
- Digikam::ImportThumbnailBar, [2119](#)
- Digikam::ItemThumbnailBar, [2425](#)
- slotStandardCancelHandler
  - Digikam::ProgressManager, [2881](#)
- slotTagDelete
  - Digikam::TagModificationHelper, [3383](#)
- slotTagEdit
  - Digikam::TagModificationHelper, [3383](#), [3384](#)
- slotTagNew
  - Digikam::TagModificationHelper, [3384](#)
- slotTagToFaceTag
  - Digikam::TagModificationHelper, [3385](#)
- slotUpdateActionsEnabled
  - Digikam::MapWidget, [2625](#)
- slotViewportEntered
  - Digikam::PersistentWidgetDelegateOverlay, [2838](#)
- Small
  - Digikam::ThumbnailSize, [3487](#)
- smoothScale
  - Digikam::DImg, [1088](#)
- smoothScaleClipped
  - Digikam::DImg, [1088](#)
- smoothScaleSection
  - Digikam::DImg, [1088](#)
- snapItemsTo
  - Digikam::GPSBookmarkModelHelper, [1792](#)
- snappedZoomFactor
  - Digikam::ImageZoomSettings, [2009](#)
- snappedZoomStep
  - Digikam::ImageZoomSettings, [2009](#)
- Socks5Proxy
  - Digikam::SystemSettings, [3280](#)
- software
  - Digikam::DRawInfo, [1395](#)
- sort
  - Digikam::DCategorizedSortFilterProxyModel, [880](#)
  - Digikam::TableViewModel, [3325](#)
- SortByAspectRatio
  - Digikam::ItemSortSettings, [2415](#)
- SortByFaces
  - Digikam::ItemSortSettings, [2415](#)
- SortByImageSize
  - Digikam::ItemSortSettings, [2415](#)
- sortByProximity
  - Digikam::ItemScanner, [2401](#)
- SortCategoriesAlphabetically
  - Digikam::ActionItemModel, [210](#)
- SortCategoriesByInsertionOrder
  - Digikam::ActionItemModel, [210](#)
- sortCategoriesByNaturalComparison
  - Digikam::DCategorizedSortFilterProxyModel, [881](#)
- sortColumn
  - Digikam::DCategorizedSortFilterProxyModel, [881](#)
- sortForInfo
  - Digikam::ItemHistoryGraph, [2278](#)
- SortOrder
  - Digikam::CamItemSortSettings, [559](#)
  - Digikam::ItemSortSettings, [2414](#)
  - ShowFoto::ShowfotoItemSortSettings, [3702](#)
- sortOrder
  - Digikam::DCategorizedSortFilterProxyModel, [881](#)
- SortOrderRole
  - Digikam::ImportFilterModel, [2054](#)
  - Digikam::ItemFilterModel, [2243](#)
  - ShowFoto::ShowfotoFilterModel, [3676](#)
- SortRole
  - Digikam::ItemSortSettings, [2414](#)
- sortRoleData
  - Digikam::AbstractAlbumModel, [146](#)
  - Digikam::DateAlbumModel, [808](#)
- soundTrackLength
  - Digikam::FFmpegLauncher, [1662](#)
- Source
  - Digikam::HistoryImageId, [1895](#)
- source
  - Digikam::DOnlineTranslator, [1277](#)
  - Digikam::ItemCopyright, [2207](#)
- sourceLanguage
  - Digikam::DOnlineTranslator, [1278](#)
- sourceLanguageName
  - Digikam::DOnlineTranslator, [1278](#)
- sourceRect
  - Digikam::ImageZoomSettings, [2009](#)
- sourceTranscription
  - Digikam::DOnlineTranslator, [1278](#)
- sourceTranslit
  - Digikam::DOnlineTranslator, [1278](#)
- SpecialMatch
  - Digikam::AlbumFilterModel, [269](#)
- spellCheckSettings
  - Digikam::DPlainTextEdit, [1296](#)
  - Digikam::DTextEdit, [1414](#)
- SPHERE
  - Digikam::Ellipsoid, [1496](#)
- SQLException
  - Digikam::BdEngineBackend, [482](#)
- SQLiteDatabaseType
  - Digikam::DbEngineParameters, [846](#)
- SqueezedComboBox
  - Digikam::SqueezedComboBox, [3253](#)
- sRGB
  - Digikam::IccProfile, [1925](#)
- SSDMOBILENET
  - Digikam::FaceScanSettings, [1624](#)
- StackedViewMode
  - Digikam::ImportStackedView, [2113](#)
- StackViewRole
  - ShowFoto::ShowfotoStackViewList, [3741](#)
- standardView
  - Digikam::DNotificationPopup, [1257](#)
- starPolygon
  - Digikam::RatingWidget, [2931](#)
- start
  - Digikam::DynamicThread, [1454](#)
  - Digikam::FacePipelineDetect, [1579](#)

- Digikam::FacePipelineDetectRecognize, [1583](#)
- Digikam::FacePipelineEdit, [1588](#)
- Digikam::FacePipelineRecognize, [1601](#)
- Digikam::FacePipelineReset, [1605](#)
- Digikam::FacePipelineRetrain, [1609](#)
- startAlbumsJobThread
  - Digikam::DBJobsManager, [860](#)
- startAnalyse
  - Digikam::AutoCrop, [418](#)
  - Digikam::DImgThreadedAnalyser, [1114](#)
  - Digikam::ImageQualityParser, [1981](#)
  - Digikam::NREstimate, [2785](#)
- startDatabaseProcess
  - Digikam::DatabaseServer, [788](#)
- startDate
  - Digikam::CoreDbUrl, [743](#)
- startDatesJobThread
  - Digikam::DBJobsManager, [860](#)
- startDrag
  - Digikam::NamespaceListView, [2757](#)
  - Digikam::TagMngrListView, [3366](#)
- startDTrashItemsListingForCollection
  - Digikam::IOJobsManager, [2158](#)
- started
  - Digikam::ClickDragReleaseItem, [602](#)
  - Digikam::DImgThreadedFilter, [1121](#)
- startEditing
  - Digikam::ComboBoxDelegate, [650](#)
- StartError
  - Digikam::DatabaseServerError, [789](#)
- startFilter
  - Digikam::DImgThreadedFilter, [1121](#)
- startFilterDirectly
  - Digikam::DImgThreadedFilter, [1121](#)
- startGPSJobThread
  - Digikam::DBJobsManager, [860](#)
- startIncrementalRefresh
  - Digikam::ImportItemModel, [2071](#)
  - Digikam::ItemModel, [2353](#)
  - ShowFoto::ShowfotoItemModel, [3701](#)
- starting
  - Digikam::DynamicThread, [1454](#)
- startIOJobs
  - Digikam::IOJobsManager, [2158](#)
- startLookup
  - Digikam::LookupAltitudeGeonames, [2588](#)
- startOfDay
  - Digikam, [134](#)
- startProcess
  - Digikam::ProcessLauncher, [2867](#)
- startRefresh
  - Digikam::ImportItemModel, [2072](#)
  - Digikam::ItemModel, [2353](#)
- startScan
  - Digikam::AlbumManager, [302](#)
- startScanningAlbumRoot
  - Digikam::CollectionScanner, [628](#)
- startSearchesJobThread
  - Digikam::DBJobsManager, [861](#)
- startTagsJobThread
  - Digikam::DBJobsManager, [861](#)
- State
  - Digikam::ItemVisibilityController, [2473](#)
- stateChanged
  - Digikam::HidingStateChanger, [1884](#)
- StateSavingDepth
  - Digikam::StateSavingObject, [3260](#)
- StateSavingObject
  - Digikam::StateSavingObject, [3261](#)
- staticMetacallPointer
  - Digikam::ParallelAdapter< A >, [2821](#)
- Status
  - Digikam::BdEngineBackend, [482](#)
  - Digikam::CollectionLocation, [608](#)
  - Digikam::DisjointMetadataDataFields, [1146](#)
  - Digikam::MetadataHub, [2634](#)
- status
  - Digikam::BdEngineBackend, [488](#)
  - Digikam::CollectionLocation, [610](#)
  - Digikam::CoreDbDownloadHistory, [733](#)
  - Digikam::ProgressItem, [2874](#)
- statusBarText
  - Digikam::TrashView, [3543](#)
- statusbarVisibility
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- StayPoppedUpComboBox
  - Digikam::StayPoppedUpComboBox, [3268](#)
- stayVisibleWhenAnimatedOut
  - Digikam::AnimatedClearButton, [375](#)
- stop
  - Digikam::DynamicThread, [1454](#)
- stopCalculation
  - Digikam::ImageHistogram, [1967](#)
- stopDatabaseProcess
  - Digikam::DatabaseServer, [788](#)
- stopHistogramComputation
  - Digikam::CurvesWidget, [759](#)
  - Digikam::HistogramWidget, [1893](#)
- stopLoading
  - Digikam::ManagedLoadSaveThread, [2609](#)
- stopSaving
  - Digikam::ManagedLoadSaveThread, [2609](#)
- store
  - Digikam::ThumbnailCreator, [3464](#)
- storeDetailThumbnail
  - Digikam::ThumbnailLoadThread, [3484](#)
- storedSize
  - Digikam::ThumbnailCreator, [3464](#)
- StoreIntermediates
  - Digikam::VersionFileOperation, [3574](#)
- storeThumbnails
  - Digikam::FaceUtils, [1651](#)
- StrictFiltering
  - Digikam::AlbumFilterModel, [269](#)
- StringComparisonType
  - Digikam::ApplicationSettings, [388](#)

- stripImageData
  - Digikam::DImg, [1089](#)
- StyleSheetDebugger
  - Digikam::StyleSheetDebugger, [3274](#)
- SubclassRoles
  - Digikam::ItemModel, [2348](#)
- subjectCode
  - Digikam::ItemExtendedProperties, [2230](#)
- subSortLessThan
  - Digikam::DCategorizedSortFilterProxyModel, [881](#)
  - Digikam::ImportFilterModel, [2055](#)
  - Digikam::ItemFilterModel, [2247](#)
  - ShowFoto::ShowfotoFilterModel, [3677](#)
- subTags
  - Digikam::AlbumManager, [302](#)
- success
  - Digikam::ProcessLauncher, [2867](#)
- suggestedWatchFlags
  - Digikam::ItemFilterModel, [2248](#)
- supportAlbums
  - Digikam::DBInfolface, [856](#)
  - Digikam::DMetalInfolface, [1195](#)
- supportBmff
  - Digikam::MetaEngine, [2702](#)
- supportedCamera
  - Digikam::DRawDecoder, [1373](#)
- supportedCodecs
  - Digikam::FFmpegLauncher, [1662](#)
- supportedDropActions
  - Digikam::DragDropModelImplementation, [1363](#)
  - Digikam::TagMngrListModel, [3365](#)
- supportedFilters
  - Digikam::BasicDImgFilterGenerator< T >, [462](#)
  - Digikam::DImgFilterGenerator, [1098](#)
  - Digikam::DImgFilterManager, [1101](#)
- supportedFormats
  - Digikam::FFmpegLauncher, [1663](#)
- supportedImageMimeTypeTypes
  - Digikam, [134](#)
- supportedRFC3066
  - Digikam::DOnlineTranslator, [1278](#)
- supportedVersions
  - Digikam::BasicDImgFilterGenerator< T >, [462](#)
  - Digikam::DImgBuiltinFilter, [1092](#)
  - Digikam::DImgFilterGenerator, [1098](#)
  - Digikam::DImgFilterManager, [1101](#)
- supportJpegXL
  - Digikam::MetaEngine, [2702](#)
- supportMetadataWriting
  - Digikam::MetaEngine, [2702](#)
- supportOlderVersionIf
  - Digikam::DImgThreadedFilter::DefaultFilterAction< Filter >, [1125](#)
- supportXmp
  - Digikam::MetaEngine, [2702](#)
- suspendCollectionScan
  - Digikam::ScanController, [3014](#)
- SVCD1
  - Digikam::VidSlideSettings, [3595](#)
- SVCD2
  - Digikam::VidSlideSettings, [3596](#)
- SVGA
  - Digikam::VidSlideSettings, [3596](#)
- SVM
  - Digikam::OpenCVDNNFaceRecognizer, [2800](#)
- SXGA
  - Digikam::VidSlideSettings, [3596](#)
- SXGAPLUS
  - Digikam::VidSlideSettings, [3596](#)
- T1
  - Digikam::GeodeticCalculator, [1769](#)
- tab
  - Digikam::DMultiTabBar, [1202](#)
- tableClicked
  - Digikam::DDatePicker, [974](#)
  - Digikam::DDateTable, [983](#)
- tables
  - Digikam::BdEngineBackend, [489](#)
- TableViewSelectionModeSyncer
  - Digikam::TableViewSelectionModeSyncer, [3327](#)
- tabStyle
  - Digikam::DMultiTabBar, [1202](#)
- TAG
  - Digikam::Album, [255](#)
- tagAdded
  - Digikam::TagsCache, [3403](#)
- TagCompleter
  - Digikam::TagCompleter, [3338](#)
- tagFilterModel
  - Digikam::DBInfolface, [856](#)
  - Digikam::DInfoInterface, [1129](#)
- TagFilterView
  - Digikam::TagFilterView, [3349](#)
- TagFolderView
  - Digikam::TagFolderView, [3355](#)
- tagForColorLabel
  - Digikam::TagsCache, [3404](#)
- tagForName
  - Digikam::TagsCache, [3404](#)
- tagForPath
  - Digikam::TagsCache, [3404](#)
- tagForPerson
  - Digikam::FaceTags, [1637](#)
- tagForPickLabel
  - Digikam::TagsCache, [3404](#)
- TaggingAction
  - Digikam::TaggingAction, [3359](#)
- taggingActionActivated
  - Digikam::AddTagsComboBox, [232](#)
  - Digikam::AddTagsLineEdit, [235](#)
- taggingActionSelected
  - Digikam::AddTagsComboBox, [232](#)
  - Digikam::AddTagsLineEdit, [235](#)
- tagId
  - Digikam::CoreDbUrl, [743](#)
- tagIds

- Digikam::CoreDbUrl, [743](#)
- Digikam::ItemInfo, [2307](#)
- tagItemHistoryGraph
  - Digikam::ItemScanner, [2401](#)
- TagModel
  - Digikam::TagModel, [3378](#)
- TagModificationHelper
  - Digikam::TagModificationHelper, [3380](#)
- tagName
  - Digikam::TagsCache, [3404](#)
- tagNames
  - Digikam::AlbumManager, [302](#), [303](#)
- tagPath
  - Digikam::TagsCache, [3404](#)
  - Digikam::TAlbum, [3438](#)
- tagPaths
  - Digikam::AlbumManager, [303](#)
- TagProperties
  - Digikam::TagProperties, [3386](#)
- tagPropsEdit
  - Digikam::TagFolderView, [3358](#)
- TagRegion
  - Digikam::TagRegion, [3394](#)
- tags
  - Digikam::DisjointMetadata, [1141](#)
- TAGS\_DATABASE
  - Digikam::ExifToolProcess, [1527](#)
- tagsContaining
  - Digikam::TagsCache, [3405](#)
- tagsDatabase
  - Digikam::ExifToolParser, [1523](#)
- tagsDbToOrderedMap
  - Digikam::ExifToolParser, [1523](#)
- tagsForName
  - Digikam::TagsCache, [3405](#)
- tagsListing
  - Digikam::TagsDBJobsThread, [3408](#)
- TagsMap
  - Digikam::MetaEngine, [2672](#)
- tagsWithProperty
  - Digikam::TagsCache, [3405](#)
- tagsWithPropertyCached
  - Digikam::TagsCache, [3405](#)
- Task
  - Digikam::VersionFileOperation, [3574](#)
- TerminationPolicy
  - Digikam::ManagedLoadSaveThread, [2608](#)
- TerminationPolicyTerminateAll
  - Digikam::ManagedLoadSaveThread, [2608](#)
- TerminationPolicyTerminateLoading
  - Digikam::ManagedLoadSaveThread, [2608](#)
- TerminationPolicyTerminatePreloading
  - Digikam::ManagedLoadSaveThread, [2608](#)
- TerminationPolicyWait
  - Digikam::ManagedLoadSaveThread, [2608](#)
- text
  - Digikam::DConfigDlgTitle, [935](#)
  - Digikam::DNotificationWidget, [1266](#)
- Digikam::DPlainTextEdit, [1297](#)
- Digikam::DTextEdit, [1414](#)
- TextStyle
  - Digikam::DMultiTabBar, [1199](#)
- THEORA
  - Digikam::VidSlideSettings, [3594](#)
- threadMutex
  - Digikam::DynamicThread, [1454](#)
- thresholds
  - Digikam::NRContainer, [2781](#)
- thumbbarVisibility
  - Digikam::DXmlGuiWindow, [1448](#)
  - Digikam::DXmlGuiWindow::Private, [1450](#)
- thumbnail
  - Digikam::DRawInfo, [1395](#)
- ThumbnailCreator
  - Digikam::ThumbnailCreator, [3462](#)
- thumbnailIdentifier
  - Digikam::ItemInfo, [2307](#)
  - Digikam::LoadingDescription, [2555](#)
- ThumbnailImageCatcher
  - Digikam::ThumbnailImageCatcher, [3468](#)
- thumbnailInfo
  - Digikam::ThumbsDbInfoProvider, [3494](#)
- thumbnailLoaded
  - Digikam::LoadSaveNotifier, [2564](#)
  - Digikam::LoadSaveThread, [2573](#)
  - Digikam::ThumbnailLoadThread, [3484](#)
- thumbnailParameters
  - Digikam::DbEngineParameters, [846](#)
- thumbnailPixmap
  - Digikam::ItemFaceDelegate, [2235](#)
- ThumbnailRole
  - Digikam::ImportItemModel, [2068](#)
  - Digikam::ItemModel, [2348](#)
  - ShowFoto::ShowfotoItemModel, [3698](#)
- thumbnailSize
  - Digikam::AlbumThumbnailLoader, [359](#)
  - Digikam::ThumbnailCreator, [3465](#)
- thumbnailToPixmapSize
  - Digikam::ThumbnailLoadThread, [3485](#)
- ThumbsDbAccess
  - Digikam::ThumbsDbAccess, [3489](#)
- ThumbsGenerator
  - Digikam::ThumbsGenerator, [3497](#)
- thumbSize
  - Digikam::DRawInfo, [1395](#)
- ThumbsSizeCtrl
  - Digikam::DZoomBar, [1458](#)
- tileNew
  - Digikam::GPSMarkerTiler, [1824](#)
  - Digikam::ItemMarkerTiler, [2343](#)
- tilerFlags
  - Digikam::AbstractMarkerTiler, [193](#)
  - Digikam::ItemMarkerTiler, [2343](#)
- timeout
  - Digikam::DNotificationPopup, [1257](#)
- timeSeed



- Digikam::RandomNumberGenerator, [2910](#)
- tipContents
  - Digikam::BlackFrameToolTip, [502](#)
  - Digikam::DItemToolTip, [1163](#)
  - Digikam::FreeSpaceToolTip, [1749](#)
  - Digikam::ItemViewToolTip, [2468](#)
- title
  - Digikam::Album, [261](#)
  - Digikam::ItemInfo, [2307](#)
- titles
  - Digikam::DisjointMetadata, [1142](#)
- titleLabel
  - Digikam::AltLangStrEdit, [373](#)
- toCaptionsMap
  - Digikam::ItemComments, [2197](#)
- toFaceTagsFaces
  - Digikam::FaceUtils, [1651](#)
- toFirstIndex
  - Digikam::ItemViewCategorized, [2448](#)
- toggled
  - Digikam::DConfigDlgWdgItem, [955](#)
  - Digikam::DConfigDlgWdgModel, [962](#)
- toggleZoomActions
  - Digikam::EditorWindow, [1487](#)
- toImage
  - Digikam::UndoMetadataContainer, [3565](#)
- toIndex
  - Digikam::ImportCategorizedView, [2019](#)
  - Digikam::ItemCategorizedView, [2189](#)
  - ShowFoto::ShowfotoCategorizedView, [3656](#)
- toJson
  - Digikam::DOnlineTranslator, [1278](#)
  - Digikam::DOnlineTranslatorOption, [1284](#)
- tokens
  - Digikam::Rule, [3000](#)
- toList
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1843](#)
- toolbarsVisibility
  - Digikam::DXmlGuiWindow::Private, [1451](#)
- toolButtonStyleSheet
  - Digikam, [134](#)
- toolGroup
  - Digikam::BatchTool, [471](#)
- toolGroupToString
  - Digikam::BatchTool, [472](#)
- toolOperations
  - Digikam::BatchTool, [472](#)
- tools
  - Digikam::DPluginBqm, [1309](#)
- Tooltip
  - Digikam::SchemeManager, [3023](#)
- toolVersion
  - Digikam::BatchTool, [472](#)
- oplevelDirectory
  - Digikam::DatabaseVersionManager, [795](#)
- ToplevelMenuCategory
  - Digikam::ActionItemModel, [210](#)
- topMargin
  - Digikam::DRawInfo, [1395](#)
- topologicalSort
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1843](#)
- toRect
  - Digikam::TagRegion, [3395](#)
- toSourceIndex
  - Digikam::RGTagModel, [2988](#)
- toString
  - Digikam::DPluginAction, [1305](#)
  - Digikam::DPluginAuthor, [1305](#)
- totalFilesToScan
  - Digikam::CollectionScanner, [628](#)
- toTransform
  - Digikam::MetaEngineRotation, [2711](#)
- toVariant
  - Digikam::TagRegion, [3396](#)
- toXml
  - Digikam::DImageHistory, [1069](#)
  - Digikam::TagRegion, [3396](#)
- train
  - Digikam::FacePipeline, [1569](#)
  - Digikam::FacialRecognitionWrapper, [1654](#), [1655](#)
  - Digikam::OpenCVDNNFaceRecognizer, [2801](#)
- TrainAll
  - Digikam::FacePipelineBase, [1574](#)
- trainData
  - Digikam::FaceDb, [1546](#)
- Trainer
  - Digikam::MLPipelineFoundation, [2729](#)
- trainer
  - Digikam::FacePipelineDetect, [1579](#)
  - Digikam::FacePipelineDetectRecognize, [1583](#)
  - Digikam::FacePipelineEdit, [1588](#)
  - Digikam::FacePipelineRecognize, [1601](#)
  - Digikam::FacePipelineReset, [1605](#)
  - Digikam::FacePipelineRetrain, [1609](#)
- TrainNew
  - Digikam::FacePipelineBase, [1574](#)
- TrainRemove
  - Digikam::FacePipelineBase, [1574](#)
- TrainReset
  - Digikam::FacePipelineBase, [1574](#)
- TRANS\_ALL\_EXIF
  - Digikam::ExifToolProcess, [1528](#)
- TRANS\_ALL\_IPTC
  - Digikam::ExifToolProcess, [1528](#)
- TRANS\_ALL\_XMP
  - Digikam::ExifToolProcess, [1528](#)
- TRANS\_TAGS
  - Digikam::ExifToolProcess, [1527](#)
- transactionFinished
  - Digikam::CoreDbBackendPrivate, [731](#)
- transform
  - Digikam::DImg, [1089](#)
  - Digikam::FileActionMngr, [1667](#)
  - Digikam::FileActionMngrFileWorker, [1676](#)

- Digikam::Haar::Calculator, [1872](#)
- Digikam::IccManager, [1917](#)
- TransformationAction
  - Digikam::MetaEngineRotation, [2710](#)
- transformations
  - Digikam::MetaEngineRotation, [2712](#)
- transformDefault
  - Digikam::IccManager, [1917](#)
- transformForDisplay
  - Digikam::IccManager, [1917](#), [1918](#)
- transformForOutput
  - Digikam::IccManager, [1918](#)
- TransformTool
  - Digikam::BatchTool, [465](#)
- transformToSRGB
  - Digikam::IccManager, [1918](#)
- transitiveClosure
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1843](#)
- transitiveReduction
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1844](#)
- translate
  - Digikam::DOnlineTranslator, [1279](#)
- translateTags
  - Digikam::ExifToolParser, [1523](#)
- TranslateTagsOps
  - Digikam::ExifToolProcess, [1527](#)
- translation
  - Digikam::DOnlineTranslator, [1279](#)
- TranslationError
  - Digikam::DOnlineTranslator, [1272](#)
- translationLanguage
  - Digikam::DOnlineTranslator, [1279](#)
- translationLanguageName
  - Digikam::DOnlineTranslator, [1279](#)
- translationOptions
  - Digikam::DOnlineTranslator, [1279](#)
- TRANSLATIONS\_LIST
  - Digikam::ExifToolProcess, [1527](#)
- translationsList
  - Digikam::ExifToolParser, [1524](#)
- translationTranslit
  - Digikam::DOnlineTranslator, [1280](#)
- Tree
  - Digikam::OpenCVDNNFaceRecognizer, [2800](#)
- TreeViewComboBox
  - Digikam::TreeViewComboBox, [3546](#)
- TreeViewLineEditComboBox
  - Digikam::TreeViewLineEditComboBox, [3550](#)
- TtsError
  - Digikam::DOnlineTts, [1285](#)
- TXGA
  - Digikam::VidSlideSettings, [3596](#)
- Type
  - Digikam::Album, [254](#)
  - Digikam::CollectionLocation, [609](#)
  - Digikam::DImgBuiltinFilter, [1090](#)
  - Digikam::HistoryImageId, [1895](#)
  - Digikam::TaggingAction, [3358](#)
- type
  - Digikam::Album, [261](#)
  - Digikam::CollectionLocation, [610](#)
  - Digikam::IccProfile, [1925](#)
  - Digikam::ItemComments, [2198](#)
  - Digikam::LoadingTask, [2561](#)
  - Digikam::SavingTask, [3008](#)
- typeForAttribute
  - Digikam::FaceTagsIface, [1646](#)
- typeForId
  - Digikam::FaceTagsIface, [1647](#)
- TypeMimeFilter
  - Digikam::MimeFilter, [2718](#)
- typeMimes
  - Digikam::DPluginDImg, [1326](#)
- TypePoint
  - Digikam::FocusPoint, [1729](#)
- typeSelection
  - Digikam::AlbumSelectors, [343](#)
- UHD4K
  - Digikam::VidSlideSettings, [3596](#)
- UHD5K
  - Digikam::VidSlideSettings, [3596](#)
- UHD6K
  - Digikam::VidSlideSettings, [3596](#)
- UHD8K
  - Digikam::VidSlideSettings, [3596](#)
- uiConfidenceThreshold
  - Digikam::DNNBaseDetectorModel, [1218](#)
- unAssignTag
  - Digikam::ItemTagPair, [2418](#)
- Unavailable
  - Digikam::BdEngineBackend, [482](#)
- unclipColors
  - Digikam::DRawDecoderSettings, [1381](#)
- unconfirmedEntry
  - Digikam::FaceTagsEditor, [1642](#)
- unconfirmedFaceCount
  - Digikam::ItemInfo, [2307](#)
- unconfirmedFaceTagsIfaces
  - Digikam::FaceTagsEditor, [1642](#)
- unconfirmedNameFaceTagsIfaces
  - Digikam::FaceTagsEditor, [1643](#)
- Undefined
  - Digikam::CollectionLocation, [609](#)
- UniqueBehavior
  - Digikam::ItemComments, [2194](#)
- uniqueCameraModel
  - Digikam::DRawInfo, [1395](#)
- uniqueHash
  - Digikam::ItemInfo, [2307](#)
  - Digikam::ThumbnailInfo, [3472](#)
- UniquePerLanguage
  - Digikam::ItemComments, [2194](#)
- UniquePerLanguageAndAuthor
  - Digikam::ItemComments, [2194](#)

- UnknownCaseSensitivity
  - Digikam::CollectionLocation, [608](#)
- unknownFaces
  - Digikam::DigikamItemView, [1065](#)
- unloadQtTranslationFiles
  - Digikam, [135](#)
- unregisterWidget
  - Digikam::GeolocationSettings, [1778](#)
- unregisterXmpNameSpace
  - Digikam::MetaEngine, [2702](#)
- Unselected
  - Digikam::TimeLineWidget, [3509](#)
- unsetCustomDatePainting
  - Digikam::DDateTable, [983](#)
- UnspecifiedOps
  - Digikam, [127](#)
- UnsupportedEmotion
  - Digikam::DOnlineTts, [1287](#)
- UnsupportedEngine
  - Digikam::DOnlineTts, [1287](#)
- UnsupportedLanguage
  - Digikam::DOnlineTts, [1287](#)
- UnsupportedVoice
  - Digikam::DOnlineTts, [1287](#)
- update
  - Digikam::TagCompleter, [3339](#)
- updateActionAvailability
  - Digikam::BackendGoogleMaps, [446](#)
  - Digikam::BackendMarble, [454](#)
- updateButton
  - Digikam::ActionVersionsOverlay, [225](#)
  - Digikam::FaceRejectionOverlay, [1619](#)
  - Digikam::HoverButtonDelegateOverlay, [1906](#)
  - Digikam::ImportRotateOverlay, [2102](#)
  - Digikam::ItemFullScreenOverlay, [2265](#)
  - Digikam::ItemRotateOverlay, [2392](#)
  - Digikam::ItemSelectionOverlay, [2406](#)
  - Digikam::ShowHideVersionsOverlay, [3213](#)
- updateClusters
  - Digikam::BackendGoogleMaps, [446](#)
  - Digikam::BackendMarble, [454](#)
  - Digikam::MapWidget, [2625](#)
  - Digikam::TileGrouper, [3500](#)
- updateContents
  - Digikam::DColorValueSelector, [907](#)
  - Digikam::DHueSaturationSelector, [1042](#)
- updateContentWidth
  - Digikam::ImportDelegate, [2038](#)
  - Digikam::ImportThumbnailDelegate, [2124](#)
  - Digikam::ItemDelegate, [2214](#)
  - Digikam::ItemThumbnailDelegate, [2430](#)
  - ShowFoto::ShowfotoDelegate, [3667](#)
  - ShowFoto::ShowfotoThumbnailDelegate, [3756](#)
- updateData
  - Digikam::CurvesWidget, [759](#)
  - Digikam::HistogramWidget, [1893](#)
- UpdateDecorationRole
  - Digikam::SetupCollectionModel, [3171](#)
- updateFace
  - Digikam::AssignNameOverlay, [403](#)
- updateFileTimeStamp
  - Digikam::MetaEngine, [2702](#)
- updateFilter
  - Digikam::AlbumFilterModel, [273](#)
- updateItem
  - Digikam::CoreDB, [720](#)
- updateItemWidgets
  - Digikam::DItemsListViewItem, [1162](#)
  - Digikam::DWItemDelegate, [1437](#)
  - Digikam::SetupCollectionDelegate, [3167](#)
- updateMarkers
  - Digikam::BackendGoogleMaps, [446](#)
  - Digikam::BackendMarble, [454](#)
- updatePALbumIcon
  - Digikam::AlbumManager, [304](#)
- updateRects
  - Digikam::DigikamItemDelegate, [1053](#)
  - Digikam::ImportDelegate, [2038](#)
  - Digikam::ImportNormalDelegate, [2087](#)
  - Digikam::ImportThumbnailDelegate, [2125](#)
  - Digikam::ItemDelegate, [2214](#)
  - Digikam::ItemFaceDelegate, [2235](#)
  - Digikam::ItemThumbnailDelegate, [2431](#)
  - ShowFoto::ShowfotoDelegate, [3667](#)
  - ShowFoto::ShowfotoNormalDelegate, [3717](#)
  - ShowFoto::ShowfotoThumbnailDelegate, [3756](#)
- updateRejectButton
  - Digikam::AssignNameWidget::Private, [410](#)
- updateSAlbum
  - Digikam::AlbumManager, [304](#)
- updateScrollbars
  - Digikam::DCategorizedView::Private, [889](#)
- updateSearch
  - Digikam::CoreDB, [720](#)
- updateSelectionData
  - Digikam::HistogramWidget, [1893](#)
- updateSettings
  - Digikam::DConfigDlgMngr, [924](#)
- updateSizeRectsAndPixmaps
  - Digikam::ImportDelegate, [2038](#)
  - Digikam::ItemDelegate, [2214](#)
  - ShowFoto::ShowfotoDelegate, [3667](#)
- updateTagShortcut
  - Digikam::TagsActionMngr, [3397](#)
- updateTAlbumIcon
  - Digikam::AlbumManager, [304](#)
- updateText
  - Digikam::AlbumSelectComboBox, [334](#)
- updateThumbInfoFromCache
  - Digikam::CameraThumbsCtrl, [555](#)
- updateThumbnailSize
  - Digikam::TableViewColumn, [3287](#)
  - Digikam::TableViewColumns::ColumnThumbnail, [3320](#)
- updateToolTip
  - Digikam::FaceRejectionOverlayButton, [1622](#)

- Digikam::ImportRotateOverlayButton, [2105](#)
- Digikam::ItemFullScreenOverlayButton, [2268](#)
- Digikam::ItemRotateOverlayButton, [2395](#)
- Digikam::ItemSelectionOverlayButton, [2409](#)
- Digikam::ItemViewHoverButton, [2458](#)
- updateUniqueHash
  - Digikam::ScanController, [3014](#)
- updateWidgets
  - Digikam::DConfigDlgMgr, [924](#)
- updateWidgetsDefault
  - Digikam::DConfigDlgMgr, [924](#)
- uploadItem
  - Digikam::GPCamera, [1789](#)
  - Digikam::UMSCamera, [3560](#)
- uploadUrl
  - Digikam::DBInfolface, [856](#)
  - Digikam::DMetalInfolface, [1195](#)
- uploadWidget
  - Digikam::DBInfolface, [856](#)
  - Digikam::DInfoInterface, [1129](#)
  - Digikam::DMetalInfolface, [1196](#)
- url
  - Digikam::CamItemInfo, [557](#)
- urlsToPaths
  - ShowFoto::ShowfotoStackViewFavoriteItem, [3734](#)
- useCompatibleFileName
  - Digikam::MetaEngine, [2703](#)
- UseEmbeddedProfile
  - Digikam::ICCSettingsContainer, [1941](#)
- useManagedPreviews
  - Digikam::lccSettings, [1940](#)
- userLoadingHint
  - Digikam::DImgPreviewItem, [1110](#)
- usesBusyIndicator
  - Digikam::ProgressItem, [2874](#)
- useTokenMenu
  - Digikam::Rule, [3000](#)
- useXMPSidecar4Reading
  - Digikam::MetaEngine, [2703](#)
- uuid
  - Digikam::ItemInfo, [2307](#)
- UW10K
  - Digikam::VidSlideSettings, [3596](#)
- UW16K
  - Digikam::VidSlideSettings, [3597](#)
- UWFHD
  - Digikam::VidSlideSettings, [3596](#)
- UXGA
  - Digikam::VidSlideSettings, [3596](#)
- vacuum
  - Digikam::CoreDB, [721](#)
  - Digikam::FaceDb, [1546](#)
  - Digikam::FacialRecognitionWrapper, [1655](#)
  - Digikam::IdentityProvider, [1953](#)
  - Digikam::SimilarityDb, [3233](#)
  - Digikam::ThumbsDb, [3488](#)
- value
  - Digikam::SearchXmlReader, [3150](#)
  - Digikam::TagProperties, [3387](#)
- valueChanged
  - Digikam::DPointSelect, [1347](#)
- valuesFromPosition
  - Digikam::DPointSelect, [1348](#)
- valueString
  - Digikam::DDoubleSliderSpinBox, [993](#)
  - Digikam::DSliderSpinBox, [1406](#)
- valueToString
  - Digikam::DMetadata, [1189](#)
- valueWidgetRects
  - Digikam::SearchFieldAlbum, [3041](#)
  - Digikam::SearchFieldCheckBox, [3045](#)
  - Digikam::SearchFieldChoice, [3050](#)
  - Digikam::SearchFieldComboBox, [3056](#)
  - Digikam::SearchFieldLabels, [3065](#)
  - Digikam::SearchFieldMonthDay, [3069](#)
  - Digikam::SearchFieldRangeDate, [3077](#)
  - Digikam::SearchFieldRangeDouble, [3081](#)
  - Digikam::SearchFieldRangeInt, [3085](#)
  - Digikam::SearchFieldRangeTime, [3088](#)
  - Digikam::SearchFieldRating, [3092](#)
  - Digikam::SearchFieldText, [3096](#)
- VBR04
  - Digikam::VidSlideSettings, [3594](#)
- VBR05
  - Digikam::VidSlideSettings, [3594](#)
- VBR10
  - Digikam::VidSlideSettings, [3594](#)
- VBR12
  - Digikam::VidSlideSettings, [3594](#)
- VBR15
  - Digikam::VidSlideSettings, [3594](#)
- VBR20
  - Digikam::VidSlideSettings, [3594](#)
- VBR25
  - Digikam::VidSlideSettings, [3594](#)
- VBR30
  - Digikam::VidSlideSettings, [3594](#)
- VBR40
  - Digikam::VidSlideSettings, [3594](#)
- VBR45
  - Digikam::VidSlideSettings, [3594](#)
- VBR50
  - Digikam::VidSlideSettings, [3594](#)
- VBR60
  - Digikam::VidSlideSettings, [3594](#)
- VBR80
  - Digikam::VidSlideSettings, [3594](#)
- VCD1
  - Digikam::VidSlideSettings, [3595](#)
- VCD2
  - Digikam::VidSlideSettings, [3595](#)
- vectortomat
  - Digikam::DNNFaceExtractorBase, [1228](#)
- version
  - Digikam::DPlugin, [1302](#)
  - Digikam::ExifToolParser, [1524](#)

- Digikam::FilterAction, [1707](#)
- VERSION\_STRING
  - Digikam::ExifToolProcess, [1527](#)
- versionFileName
  - Digikam::DefaultVersionNamingScheme, [1001](#)
  - Digikam::VersionNamingScheme, [3581](#)
- VersionFileOperation
  - Digikam::VersionFileOperation, [3574](#)
- versionManager
  - Digikam::ImageWindow, [2005](#)
- VersionNamingScheme
  - Digikam::VersionNamingScheme, [3580](#)
- verticesBreadthFirst
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1844](#)
- verticesDepthFirstSorted
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1844](#)
- verticesDominatedBy
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1844](#)
- verticesDominatedByDepthFirstSorted
  - Digikam::Graph< VertexProperties, EdgeProperties >, [1845](#)
- VGA
  - Digikam::VidSlideSettings, [3596](#)
- VidBitRate
  - Digikam::VidSlideSettings, [3594](#)
- VidCodec
  - Digikam::VidSlideSettings, [3594](#)
- VIDEOCOLORMODEL
  - Digikam::DMetadata, [1183](#)
- videoColorModelToString
  - Digikam::DMetadata, [1189](#)
- VideoMergeBackend
  - Digikam::MetaEngine, [2673](#)
- videoTypeNames
  - Digikam::VidSlideSettings, [3597](#)
- VidFormat
  - Digikam::VidSlideSettings, [3594](#)
- VidPlayer
  - Digikam::VidSlideSettings, [3595](#)
- VidStd
  - Digikam::VidSlideSettings, [3595](#)
- VidType
  - Digikam::VidSlideSettings, [3595](#)
- View
  - Digikam::SchemeManager, [3022](#)
- view
  - Digikam::DCategoryDrawer, [895](#)
  - Digikam::DNotificationPopup, [1257](#)
  - Digikam::ListViewComboBox, [2543](#)
  - Digikam::TreeViewComboBox, [3547](#)
- viewHasMultiSelection
  - Digikam::ItemDelegateOverlay, [2218](#)
- viewportClicked
  - Digikam::ItemViewCategorized, [2448](#)
- viewportEvent
  - Digikam::AbstractAlbumTreeView, [156](#)
- viewportLeaveEvent
  - Digikam::AbstractWidgetDelegateOverlay, [203](#)
  - Digikam::AssignNameOverlay, [403](#)
  - Digikam::PersistentWidgetDelegateOverlay, [2838](#)
- viewPosition
  - Digikam::DConfigDlgView, [941](#)
- visibleItems
  - Digikam::ItemVisibilityController, [2475](#)
- VisitedBackground
  - Digikam::SchemeManager, [3022](#)
- VisitedText
  - Digikam::SchemeManager, [3024](#)
- visualCategoryRectInViewport
  - Digikam::DCategorizedView::Private, [889](#)
- visualChange
  - Digikam::AssignNameOverlay, [404](#)
  - Digikam::GroupIndicatorOverlay, [1868](#)
  - Digikam::HoverButtonDelegateOverlay, [1906](#)
  - Digikam::ImportCoordinatesOverlay, [2031](#)
  - Digikam::ImportDownloadOverlay, [2045](#)
  - Digikam::ImportLockOverlay, [2082](#)
  - Digikam::ImportRatingOverlay, [2097](#)
  - Digikam::ItemCoordinatesOverlay, [2202](#)
  - Digikam::ItemDelegateOverlay, [2219](#)
  - Digikam::ItemRatingOverlay, [2388](#)
  - Digikam::TagsLineEditOverlay, [3415](#)
  - ShowFoto::ShowfotoCoordinatesOverlay, [3660](#)
- visualRect
  - Digikam::DCategorizedView::Private, [889](#)
- visualRectInViewport
  - Digikam::DCategorizedView::Private, [889](#)
- Voice
  - Digikam::DOnlineTts, [1287](#)
- voice
  - Digikam::DOnlineTts, [1289](#)
- voiceCode
  - Digikam::DOnlineTts, [1289](#)
- VolumeHardWired
  - Digikam::CollectionLocation, [609](#)
- VolumeRemovable
  - Digikam::CollectionLocation, [609](#)
- wait
  - Digikam::BdEngineBackendPrivate::ErrorLocker, [498](#)
  - Digikam::DynamicThread, [1455](#)
- waitForExifToolResult
  - Digikam::ExifToolProcess, [1530](#)
- warningContinueCancelList
  - Digikam::CameraMessageBox, [550](#)
- WarningMessage
  - Digikam::DConfigDlgTitle, [931](#)
- wasExifRotated
  - Digikam::DImg, [1089](#)
- watchFlags
  - Digikam::ItemFilterSettings, [2258](#)
  - Digikam::ItemSortSettings, [2416](#)
- wb

- Digikam::DRawDecoding, [1387](#)
- WEBMVP8
  - Digikam::VidSlideSettings, [3594](#)
- weekDayFirstOfMonth
  - Digikam::DDateTable::Private, [986](#)
- WeightBin
  - Digikam::Haar::WeightBin, [1874](#)
- WGS84
  - Digikam::Ellipsoid, [1496](#)
- WhiteBalance
  - Digikam::DRawDecoderSettings, [1376](#)
- whiteBalance
  - Digikam::DRawDecoderSettings, [1381](#)
- whiteBalanceArea
  - Digikam::DRawDecoderSettings, [1381](#)
- whitePoint
  - Digikam::DRawDecoderSettings, [1381](#)
  - Digikam::DRawInfo, [1396](#)
- wholeAlbumsChecked
  - Digikam::AlbumSelectors, [343](#)
- wholeTagsChecked
  - Digikam::AlbumSelectors, [343](#)
- WHSXGA
  - Digikam::VidSlideSettings, [3596](#)
- WHUXGA
  - Digikam::VidSlideSettings, [3596](#)
- WHXGA
  - Digikam::VidSlideSettings, [3596](#)
- widget
  - Digikam::DConfigDlgWdgItem, [955](#)
- widgetEnterEvent
  - Digikam::AbstractWidgetDelegateOverlay, [203](#)
  - Digikam::AssignNameOverlay, [404](#)
  - Digikam::FaceRejectionOverlay, [1619](#)
  - Digikam::ImportRatingOverlay, [2097](#)
  - Digikam::ImportRotateOverlay, [2102](#)
  - Digikam::ItemFullScreenOverlay, [2265](#)
  - Digikam::ItemRatingOverlay, [2388](#)
  - Digikam::ItemRotateOverlay, [2392](#)
- widgetEnterNotifyMultiple
  - Digikam::AbstractWidgetDelegateOverlay, [204](#)
- widgetLeaveEvent
  - Digikam::AssignNameOverlay, [404](#)
  - Digikam::FaceRejectionOverlay, [1619](#)
  - Digikam::ImportRatingOverlay, [2097](#)
  - Digikam::ImportRotateOverlay, [2102](#)
  - Digikam::ItemFullScreenOverlay, [2265](#)
  - Digikam::ItemRatingOverlay, [2388](#)
  - Digikam::ItemRotateOverlay, [2392](#)
- widgetModified
  - Digikam::DConfigDlgMgr, [924](#)
- WidgetRole
  - Digikam::DConfigDlgModel, [926](#)
- willHaveEffect
  - Digikam::lccTransform, [1944](#)
- willWriteMetadata
  - Digikam::DisjointMetadata, [1142](#)
  - Digikam::MetadataHub, [2635](#)
- Window
  - Digikam::SchemeManager, [3022](#)
- WMV7
  - Digikam::VidSlideSettings, [3594](#)
- WMV8
  - Digikam::VidSlideSettings, [3594](#)
- WMV9
  - Digikam::VidSlideSettings, [3594](#)
- wordWrap
  - Digikam::DNotificationWidget, [1266](#)
- WorkerObject
  - Digikam::WorkerObject, [3617](#)
- WorkerObjectQtMetacall
  - Digikam::ParallelAdapter< A >, [2821](#)
  - Digikam::ParallelWorkers, [2826](#)
- workspaceProfiles
  - Digikam::lccSettings, [1940](#)
- WQHD
  - Digikam::VidSlideSettings, [3596](#)
- WQSXGA
  - Digikam::VidSlideSettings, [3596](#)
- WQUXGA
  - Digikam::VidSlideSettings, [3596](#)
- WQXGA
  - Digikam::VidSlideSettings, [3596](#)
- WQXGAPLUS
  - Digikam::VidSlideSettings, [3596](#)
- writableFormats
  - Digikam::ExifToolParser, [1524](#)
- write
  - Digikam::DisjointMetadata, [1142](#)
  - Digikam::MetadataHub, [2636](#), [2637](#)
  - Digikam::SearchField, [3038](#)
  - Digikam::SearchFieldAlbum, [3042](#)
  - Digikam::SearchFieldCheckBox, [3046](#)
  - Digikam::SearchFieldChoice, [3050](#)
  - Digikam::SearchFieldComboBox, [3056](#)
  - Digikam::SearchFieldKeyword, [3062](#)
  - Digikam::SearchFieldLabels, [3066](#)
  - Digikam::SearchFieldMonthDay, [3070](#)
  - Digikam::SearchFieldRangeDate, [3077](#)
  - Digikam::SearchFieldRangeDouble, [3081](#)
  - Digikam::SearchFieldRangeInt, [3085](#)
  - Digikam::SearchFieldRangeTime, [3089](#)
  - Digikam::SearchFieldRating, [3093](#)
  - Digikam::SearchFieldText, [3097](#)
- WRITE\_EXISTING\_TAGS
  - Digikam::ExifToolProcess, [1528](#)
- WRITE\_FORMATS
  - Digikam::ExifToolProcess, [1527](#)
- WRITE\_TO\_FILE\_ONLY
  - Digikam::MetaEngine, [2673](#)
- WRITE\_TO\_SIDECAR\_AND\_FILE
  - Digikam::MetaEngine, [2673](#)
- WRITE\_TO\_SIDECAR\_ONLY
  - Digikam::MetaEngine, [2673](#)
- WRITE\_TO\_SIDECAR\_ONLY\_FOR\_READ\_ONLY\_FILES
  - Digikam::MetaEngine, [2673](#)

- writeDngFiles
  - Digikam::MetaEngine, [2703](#)
- writeField
  - Digikam::SearchXmlWriter, [3153](#)
- writeGroup
  - Digikam::SearchXmlWriter, [3154](#)
- writeLocations
  - Digikam::GPSItemContainer, [1806](#)
- writeMetadata
  - Digikam::FileActionMngrFileWorker, [1676](#)
- writeMetadataToFiles
  - Digikam::FileActionMngrFileWorker, [1676](#)
- WriteMode
  - Digikam::DisjointMetadata, [1139](#)
  - Digikam::FacePipeline, [1566](#)
  - Digikam::FacePipelineBase, [1575](#)
  - Digikam::MetadataHub, [2634](#)
- writeOrientationToFiles
  - Digikam::FileActionMngrFileWorker, [1676](#)
- Writer
  - Digikam::MLPipelineFoundation, [2729](#)
- writer
  - Digikam::FacePipelineDetect, [1579](#)
  - Digikam::FacePipelineDetectRecognize, [1583](#)
  - Digikam::FacePipelineEdit, [1588](#)
  - Digikam::FacePipelineRecognize, [1601](#)
  - Digikam::FacePipelineReset, [1605](#)
  - Digikam::FacePipelineRetrain, [1609](#)
- writeRawFiles
  - Digikam::MetaEngine, [2703](#)
- writeSettings
  - Digikam::DRawDecoderWidget, [1384](#)
- writeTags
  - Digikam::MetadataHub, [2637](#), [2638](#)
- writeTagsToXmp
  - Digikam::GPSItemContainer, [1806](#)
- writeToBaloo
  - Digikam::MetadataHub, [2638](#)
- writeToFile
  - Digikam::lccProfile, [1925](#)
- writeToMetadata
  - Digikam::MetadataHub, [2638](#)
- writeUnconfirmedResults
  - Digikam::FaceUtils, [1651](#)
- writeValue
  - Digikam::SearchXmlWriter, [3154](#)
- writeWithExifTool
  - Digikam::MetaEngine, [2703](#)
- WritingTagsMode
  - Digikam::ExifToolProcess, [1528](#)
- WSComboBoxIntermediate
  - Digikam::WSComboBoxIntermediate, [3626](#)
- WSXGA
  - Digikam::VidSlideSettings, [3596](#)
- WSXGAPLUS
  - Digikam::VidSlideSettings, [3596](#)
- WUXGA
  - Digikam::VidSlideSettings, [3596](#)
- WVGA
  - Digikam::VidSlideSettings, [3596](#)
- WXGA1
  - Digikam::VidSlideSettings, [3596](#)
- WXGA2
  - Digikam::VidSlideSettings, [3596](#)
- X264
  - Digikam::VidSlideSettings, [3594](#)
- xml
  - Digikam::SearchXmlWriter, [3154](#)
- xmlSection
  - Digikam::DPluginAction, [1305](#)
- xmpData
  - Digikam::DRawInfo, [1396](#)
- XmpTagType
  - Digikam::MetaEngine, [2674](#)
- xValue
  - Digikam::DPointSelect, [1348](#)
- XVGA
  - Digikam::VidSlideSettings, [3596](#)
- yesOrNo
  - Digikam::RandomNumberGenerator, [2910](#)
- YOLOv3
  - Digikam::FaceScanSettings, [1624](#)
- YOLOV5NANO
  - Digikam, [125](#), [127](#)
- YOLOV5XLARGE
  - Digikam, [125](#), [127](#)
- YoloVersions
  - Digikam, [127](#)
- YuNet
  - Digikam::FaceScanSettings, [1624](#)
- yValue
  - Digikam::DPointSelect, [1348](#)
- zoomedSize
  - Digikam::ImageZoomSettings, [2009](#)
- zoomFactor
  - Digikam::ImageZoomSettings, [2010](#)
- zoomIn
  - Digikam::BackendGoogleMaps, [446](#)
  - Digikam::BackendMarble, [454](#)
- zoomOut
  - Digikam::BackendGoogleMaps, [446](#)
  - Digikam::BackendMarble, [454](#)