

Package ‘manifesto’

May 8, 2026

Title Create Project Manifest Files

Version 0.0.2

Description Provides 'TOML' representations of packages needed that must be installed to run a project. Includes functions to specify detailed installation functions, validate files, and to use a given file as the requirements for a project. Handles package installations, when necessary, via 'pak'.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports cli, jsonlite, pak, tomledit

URL <https://github.com/christopherkenny/manifesto>,
<http://christophertkenny.com/manifesto/>

BugReports <https://github.com/christopherkenny/manifesto/issues>

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website christopherkenny/ctktemplate

VignetteBuilder knitr

Depends R (>= 4.2.0)

NeedsCompilation no

Author Christopher T. Kenny [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-9386-6860>>)

Maintainer Christopher T. Kenny <ctkenny@proton.me>

Repository CRAN

Date/Publication 2026-05-05 18:10:02 UTC

Contents

current_r_version	2
manifest_all_groups	3
manifest_check	3
manifest_check_system	4
manifest_create	4
manifest_from_description	5
manifest_from_installed	6
manifest_from_loaded	7
manifest_from_pak	8
manifest_from_renv	8
manifest_install	9
manifest_parse	10
manifest_peek	10
manifest_to_description	11
manifest_validate	11
manifest_version	12

Index	13
--------------	-----------

current_r_version	<i>Get the current R version</i>
-------------------	----------------------------------

Description

The version is returned as a string in the format "major.minor.patch" (e.g., "4.4.0"), using components from `R.version`.

Usage

```
current_r_version()
```

Value

A character string representing the full R version.

Examples

```
current_r_version()
```

manifest_all_groups *Return all defined optional dependency groups in a manifest file*

Description

Return all defined optional dependency groups in a manifest file

Usage

```
manifest_all_groups(path = "rproject.toml")
```

Arguments

path Path to the rproject.toml file.

Value

A character vector of group names (without "-dependencies")

Examples

```
manifest_all_groups(path = system.file(package = 'manifesto', 'minimal.toml'))
```

manifest_check *Check if the installed packages match the manifest requirements*

Description

Check if the installed packages match the manifest requirements

Usage

```
manifest_check(path = "rproject.toml", groups = NULL)
```

Arguments

path Path to the TOML manifest file. Defaults to "rproject.toml".
groups Optional dependency groups to include. Defaults to NULL (core only).

Value

A data.frame reporting installed version, required version, and status.

Examples

```
manifest_check(system.file(package = 'manifesto', 'minimal.toml'))
```

manifest_check_system *Check for system dependencies*

Description

This function checks for the presence of system dependencies listed in the [system-dependencies] section of the manifest file.

Usage

```
manifest_check_system(path = "rproject.toml")
```

Arguments

path Path to the rproject.toml file.

Value

A data.frame reporting the system dependency, and its status.

Examples

```
path <- manifest_create(`system-dependencies` = list(git = '*'))
manifest_check_system(path)
```

manifest_create *Create a minimal TOML manifest file*

Description

This creates a minimal TOML file with a [manifesto], [project], [environment], and empty [dependencies] section.

Usage

```
manifest_create(
  path,
  project_name = "Project",
  project_version = "0.0.1",
  manifesto_version = manifest_version(),
  r_version = "*",
  ...
)
```

Arguments

path File path to write to. If missing, a temporary file will be used.
 project_name Optional project name. Defaults to Project.
 project_version Optional project version. Defaults to 0.0.1.
 manifesto_version Optional manifesto version. Defaults to the current package version.
 r_version Optional R version settings. Defaults to '*'.
 ... Additional named arguments to add to the manifest. These will be added to the top-level of the TOML file.

Value

Invisibly returns the written path.

Examples

```

path <- manifest_create(
  dependencies = list(dplyr = '>= 1.0.0'),
  'suggests-dependencies' = list(testthat = '>= 3.0.0')
)

```

manifest_from_description

Generate a TOML manifest from a DESCRIPTION file

Description

Parses fields from a DESCRIPTION file and generates a corresponding TOML manifest with [project], [environment], and dependency groups.

Usage

```

manifest_from_description(
  description = "DESCRIPTION",
  path,
  include_empty_groups = FALSE
)

```

Arguments

description Path to the DESCRIPTION file.
 path Optional output file path. Defaults to a temporary .toml file.
 include_empty_groups Whether to include empty dependency sections.

Details

By default, both Depends and Imports are mapped to [dependencies]. Suggests, LinkingTo, and Enhances are mapped to their own optional groups.

Value

Path to the generated TOML file (invisibly).

Examples

```
path <- manifest_from_description(system.file(package = 'cli', 'DESCRIPTION'))
```

```
manifest_from_installed
```

Generate a TOML manifest from installed packages

Description

Captures all packages installed on `.libPaths()` and generates a manifesto-style TOML manifest. Package versions can either reflect their installed versions, or use a wildcard (*) to accept any.

Usage

```
manifest_from_installed(
  path,
  include_base = FALSE,
  min_version = c("installed", "*"),
  r_version = current_r_version()
)
```

Arguments

<code>path</code>	Optional path to write the manifest. If missing, a temporary <code>.toml</code> file is created.
<code>include_base</code>	Logical. Whether to include base packages. Defaults to <code>FALSE</code> .
<code>min_version</code>	Whether to require exact installed versions (<code>'installed'</code> , default) or allow any version via a wildcard (<code>'*'</code>).
<code>r_version</code>	Optional R version settings. Defaults to <code>current_r_version()</code> .

Details

Base packages are excluded by default to minimize boilerplate.

Value

Path to the generated TOML file (invisibly).

Examples

```
path <- manifest_from_installed()
```

manifest_from_loaded *Generate a TOML manifest from currently loaded packages*

Description

Captures all currently loaded packages in the R session and generates a manifesto-style TOML manifest. Package versions can either reflect their current loaded versions, or use a wildcard (*) to accept any.

Usage

```
manifest_from_loaded(  
  path,  
  include_base = FALSE,  
  min_version = c("loaded", "*"),  
  r_version = current_r_version()  
)
```

Arguments

path	Optional path to write the manifest. If missing, a temporary .toml file is created.
include_base	Logical. Whether to include base packages. Defaults to FALSE.
min_version	Whether to require exact loaded versions ('loaded', default) or allow any version via a wildcard (*).
r_version	Optional R version settings. Defaults to current_r_version()

Details

Base packages are excluded by default to minimize boilerplate.

Value

Path to the generated TOML file (invisibly).

Examples

```
path <- manifest_from_loaded()
```

manifest_from_pak *Generate a TOML manifest from a pak lockfile*

Description

Reads a {pak} lockfile (JSON format) and converts it to a manifesto-style TOML manifest. Extracts package versions and the locked R version.

Usage

```
manifest_from_pak(lockfile = "pkg.lock", path, r_version)
```

Arguments

lockfile Path to a pkg.lock JSON file created by {pak}.

path Optional path to write the manifest. Defaults to a temporary .toml file.

r_version Optional R version settings. Defaults to version found in lockfile.

Value

Path to the generated TOML file (invisibly).

Examples

```
path <- manifest_from_pak(system.file(package = 'manifesto', 'pkg.lock'))
```

manifest_from_renv *Generate a TOML manifest from a renv lockfile*

Description

Reads a {renv} lockfile (JSON format) and converts it to a manifesto-style TOML manifest. Extracts package versions and the R environment version.

Usage

```
manifest_from_renv(lockfile = "renv.lock", path, r_version)
```

Arguments

lockfile Path to a renv.lock JSON file. Defaults to 'renv.lock'.

path Optional path to write the TOML manifest. Defaults to a temporary .toml file.

r_version Optional R version settings. Defaults to version found in lockfile.

Value

Path to the generated TOML file (invisibly).

Examples

```
path <- manifest_from_renv(system.file(package = 'manifesto', 'renv.lock'))
```

manifest_install	<i>Install packages from a manifesto manifest</i>
------------------	---

Description

Install packages from a manifesto manifest

Usage

```
manifest_install(path = "rproject.toml", groups = NULL, dry_run = FALSE)
```

Arguments

path	Path to the rproject.toml file. Defaults to "rproject.toml" in the current directory.
groups	Optional character vector of dependency groups to include (e.g., "dev", "workshop").
dry_run	If TRUE, show what would be installed but do not install anything.

Value

Invisibly returns a character vector of package references that were installed.

Examples

```
manifest_install(  
  path = system.file(package = 'manifesto', 'minimal.toml'),  
  dry_run = TRUE  
)
```

manifest_parse	<i>Parse a manifesto manifest file</i>
----------------	--

Description

By default, `groups = NULL`, which will only install core dependencies. If you want to include all optional groups, set `groups = 'all'`.

Usage

```
manifest_parse(path = "rproject.toml", groups = NULL)
```

Arguments

path	Path to the <code>rproject.toml</code> file.
groups	Optional character vector of dependency groups to include.

Value

A character vector of resolved package references.

Examples

```
manifest_parse(path = system.file(package = 'manifesto', 'minimal.toml'))
```

manifest_peek	<i>Peek into a manifest file</i>
---------------	----------------------------------

Description

Peek into a manifest file

Usage

```
manifest_peek(path = "rproject.toml")
```

Arguments

path	Path to the <code>rproject.toml</code> file. Defaults to "rproject.toml" in the current directory.
------	--

Value

Invisibly returns a list with parsed contents.

Examples

```
manifest_peek(system.file('complex.toml', package = 'manifesto'))
```

manifest_to_description *Convert a TOML manifest to a DESCRIPTION file*

Description

Generates a valid DESCRIPTION file from a manifest. Required fields like Title, Description, License, and Authors@R are inserted as TODOs if not present.

Usage

```
manifest_to_description(path = "rproject.toml", out = "DESCRIPTION")
```

Arguments

path	Path to the TOML manifest file.
out	Path to the DESCRIPTION file to write. Defaults to 'DESCRIPTION'.

Value

Invisibly returns the path to the written DESCRIPTION file.

Examples

```
out <- tempfile(pattern = 'DESCRIPTION')
manifest_to_description(
  path = system.file('minimal.toml', package = 'manifesto'),
  out = out
)
```

manifest_validate *Validate a manifesto manifest file*

Description

Validate a manifesto manifest file

Usage

```
manifest_validate(path = "rproject.toml", groups = NULL)
```

Arguments

path	Path to the rproject.toml file.
groups	Optional character vector of dependency groups to include.

Value

Invisibly returns TRUE if the manifest is valid; otherwise, stops with an error.

Examples

```
manifest_validate(path = system.file(package = 'manifesto', 'minimal.toml'))
```

manifest_version	<i>Return the current version of the manifesto package</i>
------------------	--

Description

Return the current version of the manifesto package

Usage

```
manifest_version()
```

Value

A character version string.

Examples

```
manifest_version()
```

Index

[current_r_version](#), 2

[manifest_all_groups](#), 3

[manifest_check](#), 3

[manifest_check_system](#), 4

[manifest_create](#), 4

[manifest_from_description](#), 5

[manifest_from_installed](#), 6

[manifest_from_loaded](#), 7

[manifest_from_pak](#), 8

[manifest_from_renv](#), 8

[manifest_install](#), 9

[manifest_parse](#), 10

[manifest_peek](#), 10

[manifest_to_description](#), 11

[manifest_validate](#), 11

[manifest_version](#), 12