

Package ‘releaser’

June 8, 2026

Type Package

Title Help with Preparing a New Version of an R Package

Version 1.1.1

Description Helps to prepare a release. Before releasing an R package it is important to update the DESCRIPTION file and the changelog. This package prepares these files and also updates the versions according to the branches. It relies heavily on the 'desc' packages.

License MIT + file LICENSE

URL <https://github.com/TanguyBarthelemy/releaser>,
<https://tanguybarthelemy.github.io/releaser/>

BugReports <https://github.com/TanguyBarthelemy/releaser/issues>

Encoding UTF-8

Depends R (>= 4.1)

Imports gh, desc, base64enc, roxygen2

Suggests rlang, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-GB

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Tanguy Barthelemy [aut, cre, art]

Maintainer Tanguy Barthelemy <tanguy.barthelemy@insee.fr>

Repository CRAN

Date/Publication 2026-06-08 15:40:02 UTC

Contents

change_remotes_field	2
check	3
check_docs	4
get_changes	4

get_different_future_version	5
get_github_branches	6
get_latest_version	6
set_latest_deps_version	7
update_news_md	8

Index	9
--------------	----------

change_remotes_field *Change the Remotes field in DESCRIPTION*

Description

Update the Remotes field of a package DESCRIPTION file so that dependencies point to specific development targets (develop, snapshot, or main).

Usage

```
change_remotes_field(
  path,
  target = c("develop", "snapshot", "main"),
  verbose = TRUE
)
```

Arguments

path **[character]** Path to the package root directory.

target **[character]** Target branch or type of remote: must be one of "develop", "snapshot", or "main".

verbose **[logical]** Whether to print current and new remote fields (default: TRUE).

Value

Invisibly returns the new vector of remote specifications (character).

Examples

```
path_rjd3workspace <- file.path(tempdir(), "rjd3workspace")
file.copy(
  from = system.file("rjd3workspace", package = "releaser"),
  to = dirname(path_rjd3workspace),
  recursive = TRUE
)

change_remotes_field(path = path_rjd3workspace, target = "develop")
```

check

Check for missing or deprecated documentation tag

Description

`check_missing_examples_tag()` Identifies exported functions that do not declare any `@examples` (or `@examplesIf`) tag in their roxygen2 documentation. `check_missing_examples_content()` detects exported functions that declare an `@examples` tag but whose examples are not actually present in the combined example blocks. `check_missing_title()` identifies exported functions that do not define a `@title` tag in their documentation. `check_describeIn_usage()` lists all documented topics using the `@describeIn` tag. `check_return_tag_usage()` identifies functions still using `@return`. `check_missing_returns()` identifies exported functions that do not declare a `@returns` tag.

Usage

```
check_missing_examples_tag(pkg_dir)
```

```
check_missing_examples_content(pkg_dir)
```

```
check_missing_title(pkg_dir)
```

```
check_describeIn_usage(pkg_dir)
```

```
check_return_tag_usage(pkg_dir)
```

```
check_missing_returns(pkg_dir)
```

Arguments

`pkg_dir` Path to the R package.

Details

The `@describeIn` tag is generally discouraged in modern roxygen2 workflows. The `@return` tag is superseded by `@returns`.

Value

A character vector of function names

check_docs	<i>Check R package documentation</i>
------------	--------------------------------------

Description

Performs a series of documentation consistency checks on a package using roxygen2 parsed blocks. The following checks are performed:

- Missing @examples tag in exported functions
- Missing example content
- Missing @title
- Usage of discouraged @describeIn
- Usage of deprecated @return
- Missing @returns

Usage

```
check_docs(pkg_dir = ".", verbose = interactive(), error_on_fail = FALSE)
```

Arguments

pkg_dir	Path to the R package.
verbose	Logical. If TRUE, prints a formatted summary.
error_on_fail	Logical. If TRUE, stops execution if any issue is found.

Value

An object of class "releaser_doc_check" containing a named list of detected issues.

get_changes	<i>Extract changelog entries for a given version</i>
-------------	--

Description

Extracts the section of NEWS.md corresponding to a given version.

Usage

```
get_changes(path, version_number, verbose = TRUE)
```

Arguments

path	[character] Path to the package root directory.
version_number	[character] Current version number string (e.g. "1.2.3").
verbose	[logical] Whether to print current and new remote fields (default: TRUE).

Value

A character string containing the formatted changelog for the given version.

Examples

```
path_rjd3workspace <- system.file("rjd3workspace", package = "releaser")

get_changes(path = path_rjd3workspace, version_number = "Unreleased")
get_changes(path = path_rjd3workspace, version_number = "3.2.4")
get_changes(path = path_rjd3workspace, version_number = "3.5.1")
```

get_different_future_version

Compute possible future semantic versions

Description

Given a current package version, compute the potential next patch, minor, and major versions following semantic versioning rules.

Usage

```
get_different_future_version(version_number, verbose = TRUE)
```

Arguments

version_number [character] Current version number string (e.g. "1.2.3").

verbose [logical] Whether to print current and new remote fields (default: TRUE).

Value

A named character vector with:

- current_version – the input version,
- future_patch_version – next patch version,
- future_minor_version – next minor version,
- future_major_version – next major version.

Examples

```
get_different_future_version("1.2.3")
```

get_github_branches *List GitHub repository branches*

Description

Retrieve all branch names from a GitHub repository.

Usage

```
get_github_branches(  
  gh_repo = file.path("rjdverse", "rjd3toolkit"),  
  verbose = TRUE  
)
```

Arguments

gh_repo [character] GitHub repository in the format "owner/repo".
verbose [logical] Whether to print current and new remote fields (default: TRUE).

Value

A character vector with branch names.

Examples

```
get_github_branches("r-lib/usethis")
```

get_latest_version *Get latest GitHub release version*

Description

Retrieve the version number of the latest GitHub release for a repository and optionally print versions found across all branches.

Usage

```
get_latest_version(  
  gh_repo = file.path("rjdverse", "rjd3toolkit"),  
  verbose = TRUE  
)
```

Arguments

gh_repo [character] GitHub repository in the format "owner/repo".
verbose [logical] Whether to print current and new remote fields (default: TRUE).

Value

A character string with the version of the latest release.

Examples

```
get_latest_version("r-lib/usethis")
```

```
set_latest_deps_version  
  Set latest versions for rjd3* dependencies
```

Description

Update the DESCRIPTION file of a package so that all dependencies beginning with "rjd3" require the latest released version from GitHub.

Usage

```
set_latest_deps_version(path, verbose = TRUE)
```

Arguments

path [character] Path to the package root directory.
verbose [logical] Whether to print current and new remote fields (default: TRUE).

Value

Invisibly updates the DESCRIPTION file in place.

Examples

```
path_rjd3workspace <- file.path(tempdir(), "rjd3workspace")  
file.copy(  
  from = system.file("rjd3workspace", package = "releaser"),  
  to = dirname(path_rjd3workspace),  
  recursive = TRUE  
)  
  
set_latest_deps_version(path = path_rjd3workspace)
```

update_news_md	<i>Update NEWS.md for a new release</i>
----------------	---

Description

Modify the NEWS.md file of a package to replace the "Unreleased" section with a new version heading and update GitHub comparison links.

Usage

```
update_news_md(path, version_number, verbose = TRUE)
```

Arguments

path [character] Path to the package root directory.
version_number [character] Current version number string (e.g. "1.2.3").
verbose [logical] Whether to print current and new remote fields (default: TRUE).

Details

The argument version_number is the new version number to update the changelog.

Value

Invisibly returns TRUE if the file was successfully updated.

Examples

```
path_rjd3workspace <- file.path(tempdir(), "rjd3workspace")
file.copy(
  from = system.file("rjd3workspace", package = "releaser"),
  to = dirname(path_rjd3workspace),
  recursive = TRUE
)

update_news_md(path = path_rjd3workspace, version_number = "1.2.3")
```

Index

change_remotes_field, 2
character, 2, 4-8
check, 3
check_describeIn_usage (check), 3
check_describeIn_usage(), 3
check_docs, 4
check_missing_examples_content (check),
3
check_missing_examples_content(), 3
check_missing_examples_tag (check), 3
check_missing_examples_tag(), 3
check_missing_returns (check), 3
check_missing_returns(), 3
check_missing_title (check), 3
check_missing_title(), 3
check_return_tag_usage (check), 3
check_return_tag_usage(), 3

get_changes, 4
get_different_future_version, 5
get_github_branches, 6
get_latest_version, 6

logical, 2, 4-8

set_latest_deps_version, 7

update_news_md, 8