

Package: pkgdev (via r-universe)

October 17, 2026

Type Package

Title Helpers to Develop Packages with GitHub Actions

Version 0.1.0.9150

Description Helpers for package maintenance, documentation rendering and GitHub Actions workflows. The package is primarily intended for personal use, but you are welcome to use it at your own risk.

License MIT + file LICENSE

URL <https://dieghernan.github.io/pkgdev/>,
<https://github.com/dieghernan/pkgdev>

BugReports <https://github.com/dieghernan/pkgdev/issues>

Depends R (>= 4.1.0)

Imports cffr (>= 0.2.0), cli (>= 3.4.0), codemeter (>= 0.3.3),
codetools, desc, devtools (>= 2.5.0), jsonlite, knitr (>= 1.33),
lintr (>= 2.0.1), pkgdown, quarto, resmush, roxygen2 (>= 8.0.0),
roxygen2md, stringr, styler (>= 1.5.1), tools, urlchecker (>= 1.0.0),
usethis (>= 2.2.2), utils, withr, xfun

Suggests ggplot2, jsonld, rmarkdown, testthat (>= 3.0.0), yaml

VignetteBuilder knitr, quarto

Remotes quarto=github::quarto-dev/quarto-r

Config/Needs/website dieghernan/gitdevr, cranlogs, tidyverse,
lubridate, reactable, Polychrome, pkgsearch, htmltools

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

X-schema.org-keywords developer-tools, experimental, github-actions,
r, r-package, rstats

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev git make libharfbuzz-dev libgit2-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libnode-dev libx11-dev zlib1g-dev

Repository https://dieghernan.r-universe.dev

Date/Publication 2026-10-17 09:51:49 UTC

RemoteUrl https://github.com/dieghernan/pkgdev

RemoteRef HEAD

RemoteSha 01975b1243e4586b5d51d6ab3ae6c3bf89823102

Contents

add_global_gitignore	2
build_qmd	3
check_rd_titles	4
gha_check_full	4
gha_pkgdown_branch	5
gha_update_docs	6
precompute_vignette	7
update_docs	9

Index 11

add_global_gitignore	Add a global .gitignore file to a package
----------------------	---

Description

Adds a global .gitignore file to the package. The file is based on the default .gitignore provided by GitHub.

Usage

add_global_gitignore(pkg = ".")

Arguments

pkg Path to the package root directory.

Value

NULL, invisibly, after updating .gitignore files.

See Also

- `usethis::use_git_ignore()` adds entries to .gitignore.
- `usethis::git_vaccinate()` adds global Git ignore rules.

Package maintenance helpers: `update_docs()`

Examples

```
## Not run:  
add_global_gitignore()  
  
## End(Not run)
```

build_qmd

Build Quarto files for a package

Description

[build_qmd\(\)](#) is a wrapper around [quarto::quarto_render\(\)](#) that first installs a temporary copy of the package, then renders each Quarto file in a clean R session. [build_readme_qmd\(\)](#) locates your README.qmd and builds it into README.md.

Usage

```
build_qmd(files, path = ".", ..., quiet = TRUE)
```

```
build_readme_qmd(path = ".", quiet = TRUE, ...)
```

Arguments

files	Quarto files to be rendered.
path	Path to the top-level directory of the source package.
...	Additional arguments passed to quarto::quarto_render() .
quiet	If TRUE, suppresses most output. Set to FALSE if you need to debug.

Value

A [logical](#) value, TRUE, invisibly, after rendering the Quarto files.

See Also

- [update_docs\(\)](#) runs the broader package maintenance workflow.
- [devtools::build_readme\(\)](#) builds README files from R Markdown.

Documentation rendering helpers: [precompute_vignette\(\)](#)

check_rd_titles	<i>Inspect Rd file titles</i>
-----------------	-------------------------------

Description

Checks generated Rd file titles for sentence case and trailing periods.

Usage

```
check_rd_titles(pkg = ".")
```

Arguments

pkg Path to the package root directory.

Value

A [data frame](#) with one row per Rd file and columns for the source path, title, sentence-case title, final character and sentence-case check. Returns [NULL](#) if no Rd files are found.

See Also

[update_docs\(\)](#) runs this check after roxygenizing the package.

Examples

```
## Not run:
check_rd_titles()

## End(Not run)
```

gha_check_full	<i>Create a GitHub Actions workflow that checks your package regularly</i>
----------------	--

Description

The GitHub Actions workflow runs R CMD check on your package. It uses a wide range of platforms, which can be reduced by commenting out or deleting platforms in the matrix configuration.

Usage

```
gha_check_full(pkg = ".", overwrite = TRUE, cron_expr = "30 08 1 * *")
```

Arguments

pkg	Path to the package root directory.
overwrite	Whether to overwrite an existing workflow file.
cron_expr	A valid cron expression. Defaults to 08:30 UTC on the first day of the month. See Details .

Details

Use crontab.guru to check and create your own cron expression.

Value

`NULL`, invisibly, after writing a GitHub Actions workflow to `<pkg>/.github/workflows`.

Source

Examples from [r-lib/actions](https://github.com/r-lib/actions).

See Also

`usethis::use_github_action()` creates GitHub Actions workflows.

GitHub Actions workflow helpers: `gha_pkgdown_branch()`, `gha_update_docs()`

Examples

```
## Not run:
gha_check_full(cron_expr = "57 16 12 * *")

## End(Not run)
```

gha_pkgdown_branch	Create a GitHub Actions workflow that builds a R <code>pkgdown</code> site
--------------------	---

Description

The GitHub Actions workflow deploys a **pkgdown** site for your package to the `gh-pages` branch.

Usage

```
gha_pkgdown_branch(
  pkg = ".",
  overwrite = TRUE,
  platform = "macOS",
  version = "latest"
)
```

Arguments

pkg	Path to the package root directory.
overwrite	Whether to overwrite an existing workflow file.
platform	Runner operating system to use for deploying the site. See Details .
version	Runner image version. See Details .

Details

Check <https://github.com/actions/runner-images> to see the available options.

Value

`NULL`, invisibly, after writing a GitHub Actions workflow to `<pkg>/.github/workflows`.

Source

Examples from [r-lib/actions](#).

See Also

`pkgdown::build_site()` builds the package website locally.

GitHub Actions workflow helpers: `gha_check_full()`, `gha_update_docs()`

Examples

```
## Not run:
# With Ubuntu 20.04.
gha_pkgdown_branch(platform = "ubuntu", version = "20.04")

## End(Not run)
```

gha_update_docs	<i>Create a GitHub Actions workflow that documents and checks your package</i>
-----------------	--

Description

The GitHub Actions workflow documents your package (see `update_docs()`), checks it and deploys its **pkgdown** site to the gh-pages branch.

Usage

```
gha_update_docs(
  pkg = ".",
  overwrite = TRUE,
  platform = "macOS",
  version = "latest"
)
```

Arguments

pkg	Path to the package root directory.
overwrite	Whether to overwrite an existing workflow file.
platform	Runner operating system to use for deploying the site. See Details .
version	Runner image version. See Details .

Details

Check <https://github.com/actions/runner-images> to see the available options.

Value

`NULL`, invisibly, after writing a GitHub Actions workflow to `<pkg>/.github/workflows`.

Source

Examples from [r-lib/actions](#).

See Also

[update_docs\(\)](#) documents and checks your package.

GitHub Actions workflow helpers: [gha_check_full\(\)](#), [gha_pkgdown_branch\(\)](#)

Examples

```
## Not run:
# With Ubuntu 20.04.
gha_update_docs(platform = "ubuntu", version = "20.04")

## End(Not run)
```

precompute_vignette	<i>Precompute vignettes</i>
---------------------	-----------------------------

Description

Precompute vignettes following the CRAN approach described at <https://ropensci.org/blog/2019/12/08/precompute-vignettes/>.

Usage

```
precompute_vignette(
  source,
  pkg = ".",
  figure_ext = ".png",
  create_r_file = FALSE
)

precompute_vignette_all(dir = "vignettes", pkg = ".", ...)
```

Arguments

source	Name of the .Rmd.orig or .qmd.orig file, without the path (e.g. "some_name.Rmd.orig" or "some_name.qmd.orig").
pkg	Path to the package root directory.
figure_ext	File extension for figures plotted in the vignette. See Details .
create_r_file	Whether to create an additional R script with the code from the vignette.
dir	Path to the directory where the .Rmd.orig and .qmd.orig files are stored.
...	Additional arguments passed to <code>precompute_vignette()</code> .

Details

This function reads vignette source files from the package's vignettes directory and moves plots from the package root to that directory.

`precompute_vignette()` processes the files named in `source`. `precompute_vignette_all()` finds and processes every .Rmd.orig and .qmd.orig file in `dir`.

Important:

In your .Rmd.orig or .qmd.orig file, set the following chunk option when producing plots:

```
knitr::opts_chunk$set(  
  ...,  
  fig.path = "./",  
  ...,  
)
```

Value

`NULL`, invisibly, after precomputing the vignettes.

Source

Based on <https://ropensci.org/blog/2019/12/08/precompute-vignettes/>.

See Also

`update_docs()` runs the broader package maintenance workflow.

Documentation rendering helpers: `build_qmd()`

Examples

```
## Not run:  
precompute_vignette(source = "precompute.Rmd.orig")  
  
## End(Not run)
```

Description

Performs routine package maintenance:

- Clean DESCRIPTION with `usethis::use_tidy_description()`.
- Compress data in the `"./data"` and `"./R"` paths using `tools::resaveRdaFiles()`.
- Style code with `styler::style_pkg()`.
- Check URLs with `urlchecker::url_check()`.
- Roxygenize with `roxygen2::roxygenise()`.
- Precompute vignettes with `precompute_vignette_all()` if present.
- Rebuild README.qmd, README.Rmd or **pkgdown** index files, if present.
- Optimize images with `resmush::resmush_dir()`.
- Write codemeta.json with `codemeta::write_codemeta()`.
- Write CITATION.cff with `cffr::cff_write()`.
- Write a project-level Codex configuration file.

Usage

```
update_docs(
  pkg = ".",
  url_update = TRUE,
  create_codemeta = TRUE,
  create_cff = TRUE,
  build_readme = TRUE,
  verbose = TRUE,
  precompute = TRUE,
  opt_imgs = TRUE,
  opt_dir = c("man/figures", "vignettes"),
  opt_ext = "png$|jpg$",
  opt_overwrite = TRUE,
  add_contributors = TRUE,
  ...
)
```

Arguments

<code>pkg</code>	Path to the package root directory.
<code>url_update</code>	A logical value. Should URLs be updated with <code>urlchecker::url_update()</code> ?
<code>create_codemeta</code>	A logical value. Should <code>codemeta.json</code> be created with <code>codemeta::write_codemeta()</code> ?

create_cff	A logical value. Should CITATION.cff be created with <code>cffr::cff_write()</code> ?
build_readme	A logical value. Should README.qmd, README.Rmd or pkgdown index files be built?
verbose	A logical value. Should progress messages be displayed in the console?
precompute	A logical value. Should vignettes be detected and precomputed? See also <code>precompute_vignette_all()</code> .
opt_imgs	A logical value. Should images be optimized with <code>resmush::resmush_dir()</code> ?
opt_dir, opt_ext, opt_overwrite	Options passed to the dir, ext and overwrite arguments of <code>resmush::resmush_dir()</code> .
add_contributors	Deprecated.
...	Additional arguments passed to downstream functions.

Details

This function checks URLs, regenerates roxygen2 documentation and rebuilds the README. It also applies the **tidyverse** style to DESCRIPTION and package code and writes package metadata to codemeta.json.

Value

`NULL`, invisibly, after updating the package. Emits progress messages when `verbose = TRUE`.

See Also

- `build_qmd()` builds Quarto files.
- `build_readme_qmd()` builds README.md from README.qmd.
- `check_rd_titles()` checks generated Rd titles.
- `gha_update_docs()` automates this workflow with GitHub Actions.
- `precompute_vignette_all()` precomputes all vignette source files.
- `devtools::build_readme()` builds README.Rmd files.

Package maintenance helpers: `add_global_gitignore()`

Examples

```
## Not run:
update_docs()

## End(Not run)
```

Index

- * **actions**
 - gha_check_full, 4
 - gha_pkgdown_branch, 5
 - gha_update_docs, 6
- * **checks**
 - check_rd_titles, 4
- * **maintenance**
 - add_global_gitignore, 2
 - update_docs, 9
- * **renderers**
 - build_qmd, 3
 - precompute_vignette, 7

add_global_gitignore, 2

add_global_gitignore(), 10

build_qmd, 3

build_qmd(), 3, 8, 10

build_readme_qmd (build_qmd), 3

build_readme_qmd(), 3, 10

cffr::cff_write(), 9, 10

check_rd_titles, 4

check_rd_titles(), 10

codemeta::write_codemeta(), 9

data frame, 4

devtools::build_readme(), 3, 10

gha_check_full, 4

gha_check_full(), 6, 7

gha_pkgdown_branch, 5

gha_pkgdown_branch(), 5, 7

gha_update_docs, 6

gha_update_docs(), 5, 6, 10

logical, 3

NULL, 2, 4–8, 10

pkgdown::build_site(), 6

precompute_vignette, 7

precompute_vignette(), 3, 8

precompute_vignette_all
(precompute_vignette), 7

precompute_vignette_all(), 8–10

quarto::quarto_render(), 3

resmush::resmush_dir(), 9, 10

roxygen2::roxygenise(), 9

styler::style_pkg(), 9

tools::resaveRdaFiles(), 9

update_docs, 9

update_docs(), 2–4, 6–8

urlchecker::url_check(), 9

urlchecker::url_update(), 9

usethis::git_vaccinate(), 2

usethis::use_git_ignore(), 2

usethis::use_github_action(), 5

usethis::use_tidy_description(), 9