

# Package: kwb.pkgbuild (via r-universe)

September 15, 2024

**Title** R package for standardised development at KWB

**Version** 0.2.3

**Description** Helper functions for automating R package development at KWB to a predefined style.

**License** MIT + file LICENSE

**URL** <https://kwb-r.github.io/kwb.pkgbuild>,  
<https://github.com/KWB-R/kwb.pkgbuild>

**BugReports** <https://github.com/KWB-R/kwb.pkgbuild/issues>

**Imports** callr, cli, crayon, desc, fs, httr, kwb.utils, magrittr,  
openssl, pkgdown, processx, rematch2, sessioninfo, stringr,  
usethis, withr, yaml

**Suggests** covr, knitr, kwb.orcid, rmarkdown, testthat

**VignetteBuilder** knitr

**Remotes** github::kwb-r/kwb.orcid, github::kwb-r/kwb.utils

**ByteCompile** true

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**Repository** <https://kwb-r.r-universe.dev>

**RemoteUrl** <https://github.com/KWB-R/kwb.pkgbuild>

**RemoteRef** HEAD

**RemoteSha** 074a26b9ea68c42054846bcc2904b18da3b6c3d5

## Contents

copy_files_from_vignettes_dir_to_deploy_dir . . . . .	2
create_empty_branch . . . . .	3
create_empty_branch_ghpages . . . . .	4
create_pkg_dir . . . . .	5

deploy_site_github_with_extra_files . . . . .	5
deploy_to_branch_with_extra_files . . . . .	6
get_pkgname . . . . .	7
git_check_config . . . . .	8
git_setup_user . . . . .	8
read_description . . . . .	9
set_github_user . . . . .	10
use_autopkgdown . . . . .	11
use_badge_appveyor . . . . .	12
use_badge_codecov . . . . .	12
use_badge_cran . . . . .	13
use_badge_ghactions . . . . .	13
use_badge_ghactions_pkgdown . . . . .	14
use_badge_ghactions_rcmdcheck . . . . .	14
use_badge_lifecycle . . . . .	15
use_badge_runiverse . . . . .	15
use_badge_travis . . . . .	16
use_codecov . . . . .	16
use_description . . . . .	17
use_ghactions . . . . .	17
use_gitlab_ci_blogdown . . . . .	18
use_gitlab_ci_docs . . . . .	18
use_gitlab_ci_ghpages . . . . .	19
use_gitlab_ci_pkgdown . . . . .	19
use_index_md . . . . .	20
use_installation . . . . .	20
use_mit_license . . . . .	21
use_news_md . . . . .	21
use_pkg . . . . .	22
use_pkgdown . . . . .	23
use_pkg_skeleton . . . . .	24
use_readme_md . . . . .	24
write_to_gitignore . . . . .	25
write_to_rbuildignore . . . . .	25

**Index****26**


---

copy\_files\_from\_vignettes\_dir\_to\_deploy\_dir

*Copy files from Vignettes Dir to Deploy idir*

---

**Description**

Copy files from Vignettes Dir to Deploy idir

**Usage**

```
copy_files_from_vignettes_dir_to_deploy_dir(
  source_dir = ".",
  deploy_dir = "docs",
  pattern = "\\\\.json$",
  overwrite = TRUE
)
```

**Arguments**

source_dir	default: "."
deploy_dir	default: "docs"
pattern	file pattern to export (default: "\\.json\$")
overwrite	should existing files be overwritten (default: TRUE)

**Value**

files matching pattern copied to deploy\_dir

---

create\_empty\_branch    *Create Empty Branch From Github*

---

**Description**

Create Empty Branch From Github

**Usage**

```
create_empty_branch(
  repo = NULL,
  branch = NULL,
  org = "KWB-R",
  set_githubuser = TRUE,
  git_exe = path_to_git(),
  dest_dir = tempdir(),
  execute = TRUE,
  dbg = TRUE,
  ...
)
```

**Arguments**

repo	name of existing Github repository (default: NULL)
branch	name of branch to be created (default: NULL)
org	repo owner (default: "KWB-R")

set_githubuser	should the Github user be set before executing the branch creation (default: TRUE). In this case set_github_user() is run
git_exe	path to GIT executable (default: kwb.pkgbuild:::path_to_git()), only required on "windows". On all other platforms it is assumed that "git" is sufficient!
dest_dir	directory where the the repo should be checked out (default: tempdir())
execute	should a batch file be run?
dbg	print debug messages (default: TRUE)
...	additional arguments passed to set_github_user

**Value**

create empty branch (defined in 'branch') and push to org/repo on Github

---

create\_empty\_branch\_ghpages  
*Create Empty gh-pages branch*

---

**Description**

Create Empty gh-pages branch

**Usage**

```
create_empty_branch_ghpages(
  repo = NULL,
  org = "KWB-R",
  set_githubuser = TRUE,
  git_exe = path_to_git(),
  dest_dir = tempdir(),
  execute = TRUE,
  dbg = TRUE,
  ...
)
```

**Arguments**

repo	name of existing Github repository (default: NULL)
org	repo owner (default: "KWB-R")
set_githubuser	should the Github user be set before executing the branch creation (default: TRUE). In this case set_github_user() is run
git_exe	path to GIT executable (default: kwb.pkgbuild:::path_to_git()), only required on "windows". On all other platforms it is assumed that "git" is sufficient!
dest_dir	directory where the the repo should be checked out (default: tempdir())
execute	should a batch file be run?
dbg	print debug messages (default: TRUE)
...	additional arguments passed to set_github_user

**Value**

create and push an empty gh-pages branch to org/repo

---

create_pkg_dir	<i>Create Package Directory</i>
----------------	---------------------------------

---

**Description**

Create Package Directory

**Usage**

```
create_pkg_dir(pkg_dir)
```

**Arguments**

pkg\_dir            path to package directory to be created, including the pkgname as last folder

**Value**

creates directory for package if not existing

**Examples**

```
## not a valid pkg folder
## Not run:
pkg_dir <- file.path(tempdir(), "pkgname/pkgname")
create_pkg_dir(pkg_dir)

## End(Not run)
## valid pkg folder
pkg_dir <- file.path(tempdir(), "pkgname")
create_pkg_dir(pkg_dir)
```

---

deploy_site_github_with_extra_files	<i>deploy_site_github_with_extra_files</i>
-------------------------------------	--

---

**Description**

for details see pkgdown::deploy\_site\_github(), only parameter vignettes\_file\_pattern\_to\_copy added

**Usage**

```

deploy_site_github_with_extra_files(
  pkg = ".",
  vignettes_file_pattern_to_copy = "\\\\.json$",
  install = TRUE,
  tarball = Sys.getenv("PKG_TARBALL", ""),
  ssh_id = Sys.getenv("id_rsa", ""),
  commit_message = construct_commit_message(pkg),
  clean = FALSE,
  verbose = TRUE,
  host = "github.com",
  repo_slug = Sys.getenv("TRAVIS_REPO_SLUG", ""),
  ...
)

```

**Arguments**

pkg	."
vignettes_file_pattern_to_copy	file pattern for copying files from vignettes directory into deploy directory (default: "\\.json\$")
install	Optionally, opt-out of automatic installation. This is necessary if the package you're documenting is a dependency of pkgdown
tarball	tarball Sys.getenv("PKG_TARBALL", "")
ssh_id	ssh_id Sys.getenv("id_rsa", "")
commit_message	commit_message (default: construct_commit_message(pkg))
clean	Clean all files from old site (default: FALSE)
verbose	Print verbose output (default: TRUE)
host	The GitHub host url (default: "github.com")
repo_slug	The 'user/repo' slug for the repository.
...	additional arguments passed to [deploy_to_branch_with_extra_files]

**Value**

deploy pkgdown site including additional files in "vignettes" source folder to "gh-pages"

---

deploy\_to\_branch\_with\_extra\_files

*Build and deploy a site locally with extra files*

---

**Description**

Assumes that you're in a git clone of the project, and the package is already installed.

**Usage**

```

deploy_to_branch_with_extra_files(
  pkg = ".",
  vignettes_file_pattern_to_copy = "\\\\.json",
  commit_message = construct_commit_message(pkg),
  clean = FALSE,
  branch = "gh-pages",
  remote = "origin",
  github_pages = (branch == "gh-pages"),
  ...
)

```

**Arguments**

pkg	"."
vignettes_file_pattern_to_copy	file pattern for copying files from vignettes directory into deploy directory (default: "\.json\$")
commit_message	commit_message (default: construct_commit_message(pkg))
clean	Clean all files from old site (default: FALSE)
branch	The git branch to deploy to
remote	The git remote to deploy to
github_pages	Is this a GitHub pages deploy. If 'TRUE', adds a 'CNAME' file for custom domain name support, and a '.nojekyll' file to suppress jekyll rendering.
...	Additional arguments passed to pkgdown::build_site()

---

get\_pkgname

*Helper Function: Get Package Name*


---

**Description**

Helper Function: Get Package Name

**Usage**

```
get_pkgname(pkgname = NULL)
```

**Arguments**

pkgname	either package name or NULL. In this case the DESCRIPTION file in the current working directory is read and its package name used (default: NULL)
---------	---

**Value**

package name

---

git\_check\_config      *Git Check Configuration*

---

**Description**

Git Check Configuration

**Usage**

```
git_check_config git_exe = path_to_git()
```

**Arguments**

git\_exe      path to git executable on Windows (default: path\_to\_git())

**Value**

prints 'global' and 'local' Git configuration

**Examples**

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

---

git\_setup\_user      *Git Setup User (Name and Email)*

---

**Description**

Git Setup User (Name and Email)

**Usage**

```
git_setup_user(  
  github_username,  
  github_email,  
  scope = c("local", "global"),  
  git_exe = path_to_git()  
)
```



**Arguments**

github_username	username should be your GitHub username
github_email	email should be identical to email you registered with at GitHub (e.g. your.name(at)kompetenz-wasser.de)
scope	scope (default: c("local", "global"))
git_exe	path to git executable on Windows (default: path_to_git())

**Value**

runs "git" to setup the Git(Hub) user

**See Also**

<<https://support.rstudio.com/hc/en-us/community/posts/115001143667-Author-Change-with-Git-in-RStudio>>

**Examples**

```
## Not run:  
git_setup_user("mrustl", "michael.rustler@kompetenz-wasser.de")  
  
## End(Not run)
```

---

read\_description      *Helper function: read\_description*

---

**Description**

Helper function: read\_description

**Usage**

```
read_description(file = "DESCRIPTION")
```

**Arguments**

file	path to DESCRIPTION file (default: DESCRIPTION)
------	---

**Value**

list with pkg "name", "title", "desc", "version"

---

set_github_user	<i>Set Github User For GIT</i>
-----------------	--------------------------------

---

## Description

Set Github User For GIT

## Usage

```
set_github_user(
  git_username = "kwb.pkgbuild::use_autopkgdown()",
  git_fullname = "kwb.pkgbuild::use_autopkgdown()",
  git_email = "kwbr-bot@kompetenz-wasser.de",
  git_exe = path_to_git(),
  execute = FALSE,
  execute_dir = tempdir(),
  dbg = TRUE
)
```

## Arguments

git_username	Github username (default: "kwb.pkgbuild::use_autopkgdown()")
git_fullname	Github fullname (default: "kwb.pkgbuild::use_autopkgdown()")
git_email	(default: 'kwbr-bot@kompetenz-wasser.de'), is then internally derived by calling whoami::fullname() and assuming that Github "kompetenz-wasser.de" used as email on Github
git_exe	path git git executable (only used on Windows as it is not installed in the local "environment"). On all other plattformen it is assumed that a call with "git" works
execute	should a batch file be run?
execute_dir	path to directory where batch script should be run in case that execute == TRUE (default: tempdir())
dbg	print debug messages (default: TRUE)

## Value

sets globally user.name and user.email in Git

---

use_autopkgdown	<i>Use Auto Pkgdown</i>
-----------------	-------------------------

---

### Description

automate pkgdown::build\_site by running it on Travis and deploying to gh-pages branch

### Usage

```
use_autopkgdown(
  repo = NULL,
  org = "KWB-R",
  set_githubuser = TRUE,
  git_exe = path_to_git(),
  dest_dir = tempdir(),
  execute = TRUE,
  dbg = TRUE,
  ...
)
```

### Arguments

repo	name of existing Github repository (default: NULL)
org	repo owner (default: "KWB-R")
set_githubuser	should the Github user be set before executing the branch creation (default: TRUE). In this case set_github_user() is run
git_exe	path to GIT executable (default: kwb.pkgbuild:::path_to_git()), only required on "windows". On all other platforms it is assumed that "git" is sufficient!
dest_dir	directory where the the repo should be checked out (default: tempdir())
execute	should a batch file be run?
dbg	print debug messages (default: TRUE)
...	additional arguments passed to set_github_user

### Value

workflow required for automating pkgdown::build\_site() on Travis

---

use\_badge\_appveyor      *Badge appveyor*

---

**Description**

Badge appveyor

**Usage**

```
use_badge_appveyor(repo = NULL, user = "KWB-R", domain = "github")
```

**Arguments**

repo	name of repository (default: NULL)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: KWB-R)
domain	under which repository is hosted (default: "github")

**Value**

generates appveyor badge link

---

use\_badge\_codecov      *Badge codecov*

---

**Description**

Badge codecov

**Usage**

```
use_badge_codecov(repo = NULL, user = "KWB-R", domain = "github")
```

**Arguments**

repo	name of repository (default: NULL)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: KWB-R)
domain	under which repository is hosted (default: "github")

**Value**

generates codecov badge link

---

use_badge_cran	<i>Badge CRAN</i>
----------------	-------------------

---

**Description**

Badge CRAN

**Usage**

```
use_badge_cran(pkgname = NULL)
```

**Arguments**

pkgname	name of R package (default: NULL)
---------	-----------------------------------

**Value**

generates CRAN badge link

---

use_badge_ghactions	<i>Badge Github Actions</i>
---------------------	-----------------------------

---

**Description**

Badge Github Actions

**Usage**

```
use_badge_ghactions(repo = NULL, user = "KWB-R", branch = NULL)
```

**Arguments**

repo	name of repository (default: NULL)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: KWB-R)
branch	default: NULL (i.e. use "default" branch) or user defined branch (e.g. "dev")

**Value**

generates Github Actions badges link

---

use\_badge\_ghactions\_pkgdown

*Badge Github Actions Pkgdown*

---

### Description

Badge Github Actions Pkgdown

### Usage

```
use_badge_ghactions_pkgdown(repo = NULL, user = "KWB-R", branch = NULL)
```

### Arguments

repo	name of repository (default: NULL)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: KWB-R)
branch	default: NULL (i.e. use "default" branch) or user defined branch (e.g. "dev")

### Value

generates Github Actions Pkgdown badge link

---

use\_badge\_ghactions\_rcmdcheck

*Badge Github Actions RCMD Check*

---

### Description

Badge Github Actions RCMD Check

### Usage

```
use_badge_ghactions_rcmdcheck(repo = NULL, user = "KWB-R", branch = NULL)
```

### Arguments

repo	name of repository (default: NULL)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: KWB-R)
branch	default: NULL (i.e. use "default" branch) or user defined branch (e.g. "dev")

### Value

generates travis badge link

---

use\_badge\_lifecycle *Badge lifecycle*

---

**Description**

Badge lifecycle

**Usage**

```
use_badge_lifecycle(stage = "experimental")
```

**Arguments**

stage	badge declares the developmental stage of a package, according to [ <a href="https://www.tidyverse.org/lifecycle/">https://www.tidyverse.org/lifecycle/</a> ](https://www.tidyverse.org/lifecycle/)(https://www.tidyverse.org/lifecycle/). Valid arguments are: "experimental", "maturing", "stable", "retired", "archived", "dormant", "questioning", (default: "experiment")
-------	--

**Value**

generates lifecycle badge link

---

use\_badge\_runiverse *Badge R-Universe*

---

**Description**

Badge R-Universe

**Usage**

```
use_badge_runiverse(repo = NULL, user = "KWB-R")
```

**Arguments**

repo	name of repository (default: NULL)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: KWB-R)

**Value**

generates codecov badge link

---

use_badge_travis	<i>Badge travis</i>
------------------	---------------------

---

**Description**

Badge travis

**Usage**

```
use_badge_travis(repo = NULL, user = "KWB-R")
```

**Arguments**

repo	name of repository (default: NULL)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: KWB-R)

**Value**

generates travis badge link

---

use_codecov	<i>Adds codecov.yml</i>
-------------	-------------------------

---

**Description**

Adds codecov.yml

**Usage**

```
use_codecov()
```

**Value**

writes codecov.yml and adds it .Rbuildignore



---

use\_description      *Use DESCRIPTION with KWB styling*

---

**Description**

Use DESCRIPTION with KWB styling

**Usage**

```
use_description(
  author = kwb_author("rustler"),
  pkg = kwb_package("kwb.umberto"),
  version = NULL,
  license = "MIT + file LICENSE",
  copyright_holder_name = kwb_string(),
  funder = NULL
)
```

**Arguments**

author	author information in list format (default: kwb.pkgbuild::kwb_author("rustler"))
pkg	package description in list format (default: kwb.pkgbuild::kwb_package("kwb.umberto"))
version	user defined version number (e.g. 0.1.0) or default version number 0.0.0.9000 in case (version = NULL) of first release
license	license (default: "MIT + file LICENSE")
copyright_holder_name	name of copyright holder (default: kwb.pkgbuild::kwb_string())
funder	funder/funding agency of R package (default: NULL), e.g. project name (e.g. AQUANES)

**Value**

writes DESCRIPTION file using usethis::use\_description() with KWB style

---

use\_ghactions      *Adds default .github/workflows/*

---

**Description**

Adds default .github/workflows/

**Usage**

```
use_ghactions()
```

**Value**

writes `.github/workflows/` and adds it `.Rbuildignore`

---

`use_gitlab_ci_blogdown`

*Adds `.gitlab-ci.yml` (if repo contains on root in a "gh-pages" branch)*

---

**Description**

Adds `.gitlab-ci.yml` (if repo contains on root in a "gh-pages" branch)

**Usage**

```
use_gitlab_ci_blogdown(
  dest_dir = getwd(),
  yml_vector = gitlab_ci_template_blogdown()
)
```

**Arguments**

`dest_dir`            directoy to write (default: `getwd()`)  
`yml_vector`        a yml imported as string vector (default: `gitlab_ci_template_blogdown()`)

**Value**

writes `.gitlab-ci.yml`

---

`use_gitlab_ci_docs`

*Adds `.gitlab-ci.yml` (if repo contains a "docs" subfolder)*

---

**Description**

Adds `.gitlab-ci.yml` (if repo contains a "docs" subfolder)

**Usage**

```
use_gitlab_ci_docs(dest_dir = getwd(), yml_vector = gitlab_ci_template_docs())
```

**Arguments**

`dest_dir`            directoy to write (default: `getwd()`)  
`yml_vector`        a yml imported as string vector (default: `gitlab_ci_template_docs()`)

**Value**

writes `.gitlab-ci.yml` and adds it `.Rbuildignore`

---

use\_gitlab\_ci\_ghpages *Adds .gitlab-ci.yml (which should be saved in root dir of "gh-pages" branch)*

---

### Description

Adds .gitlab-ci.yml (which should be saved in root dir of "gh-pages" branch)

### Usage

```
use_gitlab_ci_ghpages(
  dest_dir = getwd(),
  yml_vector = gitlab_ci_template_ghpages()
)
```

### Arguments

dest\_dir            directory to write (default: getwd())  
 yml\_vector        a yml imported as string vector (default: gitlab\_ci\_template\_ghpages())

### Value

writes .gitlab-ci.yml

---

use\_gitlab\_ci\_pkgdown *Adds .gitlab-ci.yml*

---

### Description

Adds .gitlab-ci.yml

### Usage

```
use_gitlab_ci_pkgdown(
  dest_dir = getwd(),
  yml_vector = gitlab_ci_template_pkgdown()
)
```

### Arguments

dest\_dir            directory to write (default: getwd())  
 yml\_vector        a yml imported as string vector (default: gitlab\_ci\_template\_pkgdown(), where "`<owner>/<repo>`" is replaced with value from DESCRIPTION specified in field URL, e.g. `https://github.com/KWB-R/kwb.pkgbuild` sets "`KWB-R/kwb.pkgbuild`" as "`<owner>/<repo>`")

**Value**

writes .gitlab-ci.yml

---

use_index_md	<i>Use index.md (used for pkgdown::build_home())</i>
--------------	--

---

**Description**

Use index.md (used for pkgdown::build\_home())

**Usage**

```
use_index_md(user = "KWB-R", domain = "github", stage = "experimental")
```

**Arguments**

user            user name or organisation under which repository defined in parameter "repo" is hosted (default: "KWB-R")

domain         under which repository is hosted (default: "github")

stage          badge declares the developmental stage of a package, according to [<https://www.tidyverse.org/lifecycle/>](https://www.tidyverse.org/lifecycle/) (valid arguments are: "experimental", "maturing", "stable", "retired", "archived", "dormant", "questioning"), (default: "experiment")

**Value**

generates travis badge link

---

use_installation	<i>Helper function: describe pkg installation in index.Rmd / README.md</i>
------------------	--

---

**Description**

Helper function: describe pkg installation in index.Rmd / README.md

**Usage**

```
use_installation(pkgname, user, domain)
```

**Arguments**

pkgname        package name

user            user name or organisation under which repository defined in parameter "repo" is hosted

domain         under which repository is hosted

---

use\_mit\_license      *Create MIT licence with KWB style*

---

**Description**

Create MIT licence with KWB style

**Usage**

```
use_mit_license(
  copyright_holder = list(name = kwb_string(), start_year = NULL)
)
```

**Arguments**

copyright\_holder

list with name of copyright holder and additional start\_year (e.g. "2017") of copyright protection. If not specified (start\_year = NULL), the current year is used as year only! (default: "list(name = kwb.pkgbuild::kwb\_string(), start\_year = NULL)")

**Value**

creates MIT licence file

---

use\_news\_md      *Create KWB-styled "NEWS.md"*

---

**Description**

Create KWB-styled "NEWS.md"

**Usage**

```
use_news_md(
  news_txt = "* Added a `NEWS.md` file to track changes to the package.",
  style_guide_url = "https://style.tidyverse.org/news.html"
)
```

**Arguments**

news\_txt

text added to news (default: "\* Added a 'NEWS.md' file to track changes to the package.")

style\_guide\_url

refer to tidyverse style website documenting how to write a good "NEWS.md" (default: "https://style.tidyverse.org/news.html")

**Value**

writes "NEWS.md" (in case it is not existing)

---

use_pkg	<i>Wrapper function for preparing R package with KWB styling</i>
---------	--

---

**Description**

Wrapper function for preparing R package with KWB styling

**Usage**

```
use_pkg(
  author = kwb_author("rustler"),
  pkg = kwb_package("kwb.umberto"),
  version = "0.0.0.9000",
  license = "MIT + file LICENSE",
  copyright_holder = list(name = kwb_string(), start_year = NULL),
  funder = NULL,
  user = "KWB-R",
  domain = "github",
  stage = "experimental",
  auto_build_pkgdown = FALSE,
  dbg = TRUE,
  ...
)
```

**Arguments**

author	author information in list format (default: <code>kwb.pkgbuild::kwb_author("rustler")</code> )
pkg	package description in list format (default: <code>kwb.pkgbuild::kwb_package("kwb.umberto")</code> )
version	user defined version number (e.g. 0.1.0) or default "0.0.0.9000" version number for initial development version (default: "0.0.0.9000")
license	license (default: "MIT + file LICENSE")
copyright_holder	list with name of copyright holder and additional start_year (e.g. "2017") of copyright protection. If not specified (start_year = NULL), the current year is used as year only! (default: " <code>list(name = kwb.pkgbuild::kwb_string(), start_year = NULL)</code> ")
funder	funder/funding agency of R package (default: NULL), e.g. project name (e.g. AQUANES)
user	user name or organisation under which repository defined in parameter "repo" is hosted (default: "KWB-R")
domain	under which repository is hosted (default: "github")

stage	badge declares the developmental stage of a package, according to [https://www.tidyverse.org/lifecycle/](https://www.tidyverse.org/lifecycle/)(H valid arguments are: "experimental", "maturing", "stable", "retired", "archived", "dormant", "questioning"), (default: "experiment")
auto_build_pkgdown	prepare Travis for pkgdown::build_site() (default: FALSE), only possible if GITHUB repo already existing
dbg	print debug messages (default: TRUE)
...	additional arguments passed to use_autopkgdown() (only relevant if "auto_build_pkgdown" == TRUE)

**Value**

writes DESCRIPTION file using usethis::use\_description() with KWB style

---

use_pkgdown	<i>pkgdown for KWB</i>
-------------	------------------------

---

**Description**

pkgdown for KWB

**Usage**

```
use_pkgdown(
  author = kwb_author("rustler"),
  copyright_holder_name = kwb_string(),
  pkg = get_pkgname(),
  user = "kwb-r",
  domain = "github"
)
```

**Arguments**

author	list of author attributes (default: kwb.pkgbuild::kwb_author("rustler"))
copyright_holder_name	name of copyright holder (default: kwb.pkgbuild::kwb_string())
pkg	name of KWB package (default: get_pkgname())
user	name of GitHub user/organisation (default: 'kwb-r')
domain	name of domain for webpage publishing (default: 'github')

**Value**

performs usethis::use\_pkgdown() and additionally writes \_pkgdown.yml based on KWB styling

`use_pkg_skeleton`      *Use Package Skeleton*

---

### Description

Use Package Skeleton

### Usage

```
use_pkg_skeleton(pkg_name)
```

### Arguments

`pkg_name`          name of R package

### Value

creates pkg skeleton in current working directory

### Examples

```
## valid pkg folder  
pkg_name <- "pkgname"  
pkg_dir <- file.path(tempdir(), pkg_name)  
pkg_dir <- create_pkg_dir(pkg_dir)  
withr::with_dir(pkg_dir, {use_pkg_skeleton(pkg_name)})
```

---

`use_readme_md`          *Use README*

---

### Description

Use README

### Usage

```
use_readme_md(user = "KWB-R", domain = "github", stage = "experimental")
```

### Arguments

`user`                user name or organisation under which repository defined in parameter "repo" is hosted (default: "KWB-R")

`domain`             under which repository is hosted (default: "github")

`stage`              badge declares the developmental stage of a package, according to [https://www.tidyverse.org/lifecycle/](https://www.tidyverse.org/lifecycle/)(h valid arguments are: "experimental", "maturing", "stable", "retired", "archived", "dormant", "questioning"), (default: "experiment")



**Value**

generates README.md

---

write\_to\_gitignore      *Write to .gitignore*

---

**Description**

Write to .gitignore

**Usage**

```
write_to_gitignore(ignore_pattern, comment = NULL)
```

**Arguments**

ignore\_pattern    pattern that should be added to .Rbuildignore  
comment            optional comment to write to file before the ignore pattern #' (default: NULL)

**Value**

.gitignore created/updated with pattern

---

write\_to\_rbuildignore    *Write to .Rbuildignore*

---

**Description**

Write to .Rbuildignore

**Usage**

```
write_to_rbuildignore(ignore_pattern, comment = NULL)
```

**Arguments**

ignore\_pattern    pattern that should be added to .Rbuildignore  
comment            optional comment to write to file before the ignore pattern (default: NULL)

**Value**

.Rbuildignore created/updated with pattern

# Index

copy\_files\_from\_vignettes\_dir\_to\_deploy\_dir, use\_readme\_md, [24](#)  
[2](#)  
create\_empty\_branch, [3](#)  
create\_empty\_branch\_ghpages, [4](#)  
create\_pkg\_dir, [5](#)  
  
deploy\_site\_github\_with\_extra\_files, [5](#)  
deploy\_to\_branch\_with\_extra\_files, [6](#)  
  
get\_pkgname, [7](#)  
git\_check\_config, [8](#)  
git\_setup\_user, [8](#)  
  
read\_description, [9](#)  
  
set\_github\_user, [10](#)  
  
use\_autopkgdown, [11](#)  
use\_badge\_appveyor, [12](#)  
use\_badge\_codecov, [12](#)  
use\_badge\_cran, [13](#)  
use\_badge\_ghactions, [13](#)  
use\_badge\_ghactions\_pkgdown, [14](#)  
use\_badge\_ghactions\_rcmdcheck, [14](#)  
use\_badge\_lifecycle, [15](#)  
use\_badge\_runiverse, [15](#)  
use\_badge\_travis, [16](#)  
use\_codecov, [16](#)  
use\_description, [17](#)  
use\_ghactions, [17](#)  
use\_gitlab\_ci\_blogdown, [18](#)  
use\_gitlab\_ci\_docs, [18](#)  
use\_gitlab\_ci\_ghpages, [19](#)  
use\_gitlab\_ci\_pkgdown, [19](#)  
use\_index\_md, [20](#)  
use\_installation, [20](#)  
use\_mit\_license, [21](#)  
use\_news\_md, [21](#)  
use\_pkg, [22](#)  
use\_pkg\_skeleton, [24](#)  
use\_pkgdown, [23](#)  
  
write\_to\_gitignore, [25](#)  
write\_to\_rbuildignore, [25](#)