

Package: pkgmkr (via r-universe)

June 9, 2026

Title Create an R Package with One Simple Function

Version 0.1.0

Description Simplifies the process of creating R packages by providing a single, straightforward function that handles package initialization, Git integration, license selection, README generation, and pkgdown site setup. Package creation can also be configured using YAML or JSON configuration files.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Imports available, configr, fs, pkgdown, usethis, withr

Suggests covr, knitr, rmarkdown, testthat

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 libx11-dev

Repository <https://sdhutchins.r-universe.dev>

Date/Publication 2025-12-17 00:28:31 UTC

RemoteUrl <https://github.com/sdhutchins/pkgmkr>

RemoteRef HEAD

RemoteSha 7c760b60d573b0e71fe941143c399d05932562a2

Contents

import_config	2
mk_pkg	2
mk_pkg_from_config	3
write_config	3

Index	4
--------------	----------

import_config	<i>Package Config</i>
---------------	-----------------------

Description

Import configuration data from a file.

Usage

```
import_config(path, file_type = "yaml")
```

Arguments

path	character. The path to the configuration file.
file_type	character. The file type of the configuration file (either "yaml" or "json").

Value

A list containing the configuration data.

mk_pkg	<i>Make Package</i>
--------	---------------------

Description

Create a new R package with the specified parameters

Usage

```
mk_pkg(  
  path,  
  author,  
  email,  
  git = TRUE,  
  git_username = NULL,  
  git_email = NULL,  
  readme_md = TRUE,  
  check_pkg_name = TRUE,  
  license = "MIT",  
  pkgdown = TRUE  
)
```

Arguments

path	character. The path to create the package at.
author	character. The name of the package author.
email	character. The email to use for Git and GitHub.
git	logical. Whether to initialize a Git repository for the package.
git_username	character. The username to use for Git and GitHub.
git_email	character. The email to use for Git and GitHub.
readme_md	logical. Whether to create a README file in markdown format.
check_pkg_name	logical. Whether to check that the package name is available on CRAN.
license	character. The license to use for the package (either "MIT" or "GPL-3").
pkgdown	logical. Whether to create a pkgdown site for the package.

mk_pkg_from_config *Make Package from Config*

Description

Create a package based on a configuration file.

Usage

```
mk_pkg_from_config(config_path, file_type = "yaml")
```

Arguments

config_path	character. The path to the configuration file.
file_type	character. The file type of the configuration file (either "yaml" or "json").

write_config *Write Config*

Description

Write configuration data to a file.

Usage

```
write_config(path, config_data)
```

Arguments

path	character. The path to write the configuration file to.
config_data	list. The configuration data to write to the file.

Index

`import_config`, [2](#)

`mk_pkg`, [2](#)

`mk_pkg_from_config`, [3](#)

`write_config`, [3](#)