Package ‘pax’

March 14, 2017

Type Package

Title A Package Templating Package

Version 0.2.0

Date 2017-03-14

Author Tyler Rinker <tyler.rinker@gmail.com>

Maintainer Tyler Rinker <tyler.rinker@gmail.com>

Depends R (>= 3.1.0)

Imports clipr, curl, rmarkdown, utils

Suggests testthat, knitr

LazyData TRUE

Description A templating system for R packages that assumes the use of GitHub, RStudio, 'testthat', Travis-CI and 'covr' (coveralls).

License GPL-2

URL http://trinker.github.com/pax/

BugReports http://github.com/trinker/pax/issues

Roxygen list(wrap = FALSE)

RoxygenNote 6.0.1

RemoteType local

RemoteUrl C:\{ }Users\{ }trinker\{ }GitHub\{ }pax

RemoteSha NA

RemoteBranch master

RemoteUsername NA

RemoteRepo NA

R topics documented:

is.global *Test If Environment is Global*

Description

A logical test to determine if the current environment is the global environment.

Usage

```
is.global(n = 1)
```

Arguments

n	The number of generations to go back. If used as a function argument n should be set to 2.

Value

A logical response.

Author(s)

Simon O'Hanlon and Tyler Rinker <tyler.rinker@gmail.com>

References

http://stackoverflow.com/questions/18637656/detect-if-environment-is-global-enviroment

See Also

globalenv, parent.frame

Examples

```
is.global()
lapply(1:3, function(i) is.global())
FUN <- function() is.global(); FUN()

FUN2 <- function(x = is.global(2)) x
FUN2()
FUN3 <- function() FUN2(); FUN3()
```

new_data — *Add and Document (**roxygen2** Style) Data*

Description

Add a '.rda' file to the 'data' directory of a package and document **roxygen2** style in the 'R/____package.R' file.

Usage

```
new_data(data, data.path = "data", stand.alone = FALSE,
  doc.path = sprintf("R/%s-package.R", basename(getwd())))
```

Arguments

data	A data set (environment, list, data.frame, vector).
data.path	The path to the data directory (where the data will be placed). Defaults to 'data'.
stand.alone	logical. If TRUE the documentation is added to its own file.
doc.path	The path to the package documentation '.R' file. Use NULL to skip adding documentation.

Value

Generates a '____.rda' file and accompanying **roxygen2** style documentation.

References

http://r-pkgs.had.co.nz/data.html#documenting-data

See Also

Other new functions: new_r_test, new_r, new_test, new_vignette

Examples

```
## Not run:
pax("temp_dir", open = FALSE)
dir.create("temp_dir/data")
## Note: If used in RStudio with the root directory set to the
##       package the user does not need to supply `data.path` or `doc.path`
new_data(mtcars, data.path = "temp_dir/data",
    doc.path = "temp_dir/R/temp_dir-package.R")
unlink("temp_dir", TRUE, TRUE)

## End(Not run)
```

new_r *Generate* **roxygen2** *Style R Files*

Description

new_r - Quickly produce a **roxygen2** style '.R' template file from a function (output file will include the function) or a character string.

new_r_min - A minimal version of new_r that does not output

Usage

```
new_r(fun, path = "R", file.name = NULL)

new_r_min(fun, path = "R", file.name = NULL)
```

Arguments

fun	A function or character string naming the function.
path	Path to directory to generate the function test in. Default is to use "R" for ease of use within RStudio. Setting to TRUE copies just the roxygen2 code to the clipboard. Setting to NULL just prints the output to the console.
file.name	By default the file is named the same as fun + ".R". This can be changed by supplying a file name to file.name.

Value

Generates a '____.R' file.

References

http://r-pkgs.had.co.nz/man.html

See Also

Other new functions: new_data, new_r_test, new_test, new_vignette

Examples

```
dir.create("temp_dir")
new_r(paste, "temp_dir")
new_r("myfun", "temp_dir")
unlink("temp_dir", TRUE, TRUE)
```

new_r_test *Wrapper:* new_r + new_test

Description

Quickly produce a **roxygen2** style '.R' and **testthat** style test '.R' template files from a function (output file will include the function) or a character string. Wraps new_r and new_test function capabilities into one function call.

Usage

```
new_r_test(fun, r.path = "R", r.file.name = NULL,
  test.path = "tests/testthat", test.file.name = NULL)
```

Arguments

fun	A function or character string naming the function.
r.path	Path to directory to generate the function test in. Default is to use "R" for ease of use within RStudio.
r.file.name	By default the file is named the same as fun + ".R". This can be changed by supplying a file name to file.name.
test.path	Path to directory to generate the function test in. Default is to use "tests/testthat" for ease of use within RStudio.
test.file.name	By default the file is named the same as: "text-" + fun + ".R". This can be changed by supplying a file name to file.name.

Value

Generates '____.R' and 'test-____.R' files.

References

http://r-pkgs.had.co.nz/man.html
http://r-pkgs.had.co.nz/tests.html

See Also

Other new functions: new_data, new_r, new_test, new_vignette

Examples

```
dir.create("temp_dir")
new_r_test(paste, r.path = "temp_dir", test.path = "temp_dir")
new_r_test("myfun", r.path = "temp_dir", test.path = "temp_dir")
unlink("temp_dir", TRUE, TRUE)
```

new_test *Generate* **testthat** *Style Test .R Files*

Description

Quickly produce a **testthat** style test '.R' template file from a function (output file will include the function) or a character string.

Usage

```
new_test(fun, path = "tests/testthat", file.name = NULL)
```

Arguments

fun	A function or character string naming the function.
path	Path to directory to generate the function test in. Default is to use "tests/testthat" for ease of use within RStudio.
file.name	By default the file is named the same as: "text-" + fun + ".R". This can be changed by supplying a file name to file.name.

Value

Generates a 'test-____.R' file.

References

http://r-pkgs.had.co.nz/tests.html

See Also

Other new functions: new_data, new_r_test, new_r, new_vignette

Examples

```
dir.create("temp_dir")
new_test(paste, "temp_dir")
new_test("myfun", "temp_dir")
unlink("temp_dir", TRUE, TRUE)
```

new_vignette *Add an* **rmarkdown** *Template*

Description

Generates an **rmarkdown** style template ‘.Rmd’ file and adds **knitr** to VignetteBuilder & Suggests fields if not already included.

Usage

```
new_vignette(title = paste("Introduction to", basename(getwd())),
  file.name = paste0(gsub("\\s+", "_", tolower(title)), ".Rmd"),
  vign.path = "vignettes", add.builder = TRUE, desc.path = "DESCRIPTION",
  builder = "knitr")
```

Arguments

title	The title of the **rmarkdown** vignette document. Defaults to *Introduction to x* where *x* is the name of the package.
file.name	The name of the vignette file. Defaults to lower case, underscore (for space) .Rmd file that utilizes the title argument.
vign.path	The path to the vignettes directory (where the vignette will be placed).
add.builder	logical. If TRUE the builder argument will be added as the VignetteBuilder fied in the ‘DESCRIPTION’ file. Additionally, if builder = "knitr" this will be added to Suggests: field if not already a dependency.
desc.path	The path to the ‘DESCRIPTION’ file. Defaults to desc.path = "DESCRIPTION".
builder	The name of the VignetteBuilder in the ‘DESCRIPTION’ file. Defaults to builder = "knitr".

Value

Generates a ‘____.Rmd’ vignette file template.

References

http://rmarkdown.rstudio.com/package_vignette_format.html
http://r-pkgs.had.co.nz/vignettes.html

See Also

Other new functions: new_data, new_r_test, new_r, new_test

Examples

```
## Not run:
pax("temp_dir", open = FALSE)
dir.create("temp_dir/vignettes")
## Note: If used in RStudio with the root directory set to the
##       package the user does not need to supply `vign.path` or `desc.path`
new_vignette("intro", vign.path = "temp_dir/vignettes",
    desc.path = "temp_dir/DESCRIPTION")
```

```
unlink("temp_dir", TRUE, TRUE)

## End(Not run)
```

pax **pax**: *A Gold Version Package Template*

Description

pax is a package template system that is NOT designed to be light weight. It is the deluxe, gold version of a package template. **pax** is not flexible, rather it maintains and enforces a narrow package management philosophy.

This function creates a package template. It utilizes a framework with defaults that expects the user to use **testthat**, Travis-CI and **covr** (coveralls) to maintain the package. This is not a light weight template, but a deluxe template.

Usage

```
pax(path, name = getOption("name"), email = getOption("email"),
  license = getOption("license"), open = is.global(2), news = TRUE,
  readme = TRUE, rstudio = TRUE, gitignore = TRUE, testthat = TRUE,
  travis = TRUE, coverage = TRUE, github.user = getOption("github.user"),
  samples = getOption("samples"), tweak = getOption("tweak"), ...)
```

Arguments

path — location to create new regular expression library package. The last component of the path will be used as the package name.

name — A named vector that minimally contains the user's first and last name (e.g., c(first="Tyler", last="Rinker"))). This can be set in the user's options in the '.Rprofile'; for example:
options(name = c(first="Tyler", middle = "W.", last="Rinker")).

email — An email address to use for **CRAN** maintainer. This can be set in the user's options in the '.Rprofile'; for example:
options(email = "tyler.rinker@gmail.com").

license — A license to use in the 'DESCRIPTION' file (e.g., "GPL-2", "MIT"). This can be set in the user's options in the '.Rprofile'; for example:
options(license = "GPL-2").

open — logical. If TRUE the project will be opened in RStudio. The default is to test if new_report is being used in the global environment, if it is then the project directory will be opened.

news — logical. If TRUE a 'NEWS' file is generated.

readme — logical. If TRUE a 'README.md' file is generated.

rstudio — logical. If TRUE it is assumed RStudio will be used and a 'xxx.proj' file is generated.

gitignore — logical. If TRUE a '.gitignore' file is generated.

testthat — logical. If TRUE it is assumed **testthat** will be used and a 'test' sub-folder with appropriate **testthat** subdirectories and files will be created.

travis logical. If TRUE it is assumed Travis-CI will be used and a ‘travis.yml’ file is generated. For more on managing a ‘travis.yml’ with **R** see: https://github.com/craigcitro/r-travis.

coverage logical. If TRUE it is assumed **covr** will be used. This information will be added to the ‘travis.yml’.

github.user The user’s GitHub user name. This can be set in the user’s options in the ‘.Rprofile’; for example:
options(github.user = "trinker").

samples logical. If TRUE a sample ‘.R’ regular expression file will be placed in the ‘R’ directory. Additionally, if testthat = TRUE, a sample ‘.R’ unit test will be placed in the ‘./tests/testthat’ directory. This can be set in the user’s options in the ‘.Rprofile’; for example:
options(samples = TRUE).

tweak Additional user supplied function that can be sourced at the end of the package creation. The following parameters are passed to your function automatically: (1) the package’s name, (2) qpath (a function that binds together path pieces; the starting piece is supplied by the path argument, (3) name (vector of 2: first and last), (4) your email, (5) path, codeand (6) github.user. This can be argument can be set in the user’s options in the ‘.Rprofile’; for example:
options(tweak = "C:/Users/Tyler/Copy/Public Scripts/augpax.R").
This argument can be a path to or url to a user specified ’tweaking’ function. The user can also pass the function directly to tweak.

... Other arguments passed to the user supplied tweak function.

Examples

```
## Not run:
pax("DELETE_ME")

## Set a package location in .Rprofile and use `ploc` to conveniently
## complete the full path to where the package should be created
##
options(dir = file.path(Sys.getenv("USERPROFILE"), "Desktop"))
pax(ploc("DELETE_ME"))

## End(Not run)
```

---

pax_options   **pax** *Options Template for* ‘.Rprofile’

Description

Generates a script template of blank **pax** options that can be added to the ‘.Rprofile’.

Usage

```
pax_options(copy2clip = interactive())
```

Arguments

copy2clip logical. If TRUE attempts to copy the output to the clipboard.

Value

Returns a script template of blank **pax** options that can be added to the '.Rprofile' for greater customization (optionally copies to the clipboard).

Examples

```
pax_options()
```

---

ploc *Package File Path Location*

Description

A wrapper for `file.path` that allows the user to (1) set a package directory location and (2) supply a package name and a file path is constructed. This is the inverse of `basename` + `dirname`.

Usage

```
ploc(base, dir = getOption("dir"))
```

Arguments

`base`	The base name (package name).
`dir`	The location to place base. This can be set in the user's `options` in the '.Rprofile'; for example: `options(dir = file.path(Sys.getenv("USERPROFILE"), "GitHub")).`

Value

Returns a concatenated file path using `dir` and `base`.

See Also

`file.path`, `basename`, `dirname`

Examples

```
## Not run:
options(dir = file.path(Sys.getenv("USERPROFILE"), "Desktop"))
pax(ploc("DELETE_ME"))

## End(Not run)
```

rox_data *Generate* **roxygen2** *Style Data Documentation*

Description

Generate **roxygen2** style data documentation.

Usage

```
rox_data(data, copy2clip = TRUE, verbose = TRUE, ...)
```

Arguments

data	A data set (environment, list, data.frame, vector).
copy2clip	logical. If TRUE copies output to the clipboard.
verbose	logical. If TRUE prints output to the console.
...	ignored.

Examples

```
## Not run:
rox_data(mtcars)

## End(Not run)
```

Index