

Package ‘cabinets’

January 27, 2020

Title Project Specific Workspace Organization Templates

Version 0.4.0

Description Creates project specific directory and file templates that are written to a .Rprofile file. Upon starting a new R session, these templates can be used to streamline the creation of new directories that are standardized to the user's preferences.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.99.9001

Imports here, rjson, glue, R6, usethis, rstudioapi, crayon, fs, stringr, utils, purrr, withr, git2r

Suggests testthat (>= 2.1.0), covr

NeedsCompilation no

Author Nick Williams [aut, cre] (<<https://orcid.org/0000-0002-1378-4831>>)

Maintainer Nick Williams <ntwilliams.personal@gmail.com>

Repository CRAN

Date/Publication 2020-01-27 15:10:02 UTC

R topics documented:

cabinets	2
cabinets_options_set	2
create_cabinet	3
create_r_proj	3
edit_r_profile	5
FileCabinet	5
get_cabinets	6
new_cabinet_proj	7

Index	8
--------------	----------

cabinets

Streamlined organization with project specific custom templates

Description

Cabinets makes it easy to create project specific file structure templates that can be referenced at the start of any R session. Cabinets works by writing project specific file templates to the .Rprofile file of the default working directory. Doing so allows the templates to be accessed in new R sessions without having to redefine them. On first use, users will be prompted for explicit permission to write to .Rprofile. Permission to write can be revoked at any time by setting the permission option in the .Rprofile file to FALSE. Due to these explicit permission requirements, cabinets will only work in interactive R sessions.

cabinets_options_set *Set cabinets options*

Description

Set cabinets options

Usage

```
cabinets_options_set(..., .envir = NULL)
```

Arguments

...	Options to be set
.envir	Environment to set options in; if NULL, will use environment at parent.frame()

Details

Mainly used for specifying if cabinets has permission to write to .Rprofile. Permission can be revoked at any time by opening the .Rprofile file and setting "cabinets.permission" = FALSE.

Value

If no options are set, returns the options specified in options.

create_cabinet	<i>Create a cabinet template</i>
----------------	----------------------------------

Description

create_cabinet writes code to the .Rprofile file so that when new R sessions are started, the newly created cabinet, an R6 object of class FileCabinet, is available in the global environment as a hidden object. The cabinet simply stores file location and file template information that new_cabinet_proj uses to create new projects with the pre-defined structure.

Usage

```
create_cabinet(name, directory, structure)
```

Arguments

name	Name of the cabinet; character of length 1. This is how the cabinet will be referenced, so best to chose something memorable.
directory	The file path for where the cabinet will exist.
structure	A list of paths of folders/files to create. See details for further explanation.

Details

Before writing to or creating a .Rprofile file, cabinets will explicitly ask for the user's permission to pn exit. The cabinet structure should be defined using a list with the names defining folder paths. List values should be set to NULL.

Value

An R6 object of class FileCabinet. The code to generate this object is written to the .Rprofile file of the home directory.

See Also

[new_cabinet_proj](#)

create_r_proj	<i>R project settings</i>
---------------	---------------------------

Description

create_r_proj is a helper function for new_cabinet_proj. Calling it outside the scope of new_cabinet_proj will only print the specified settings the console.

Usage

```

create_r_proj(
  version = "1.0",
  restore_workspace = c("No", "Yes", "Default"),
  save_workspace = c("No", "Yes", "Default"),
  save_history = c("Default", "No", "Yes"),
  enable_code_indexing = c("Yes", "No", "Default"),
  spaces_for_tab = c("Yes", "No", "Default"),
  num_spaces_for_tab = 2,
  encoding = "UTF-8",
  rnw_weave = "Sweave",
  latex = "pdfLaTeX",
  auto_append_new_line = "Yes",
  strip_trailing_white_space = "Yes"
)

```

Arguments

version	R project version number, to be passed as character string.
restore_workspace	Load the .Rdata file (if any) found in the initial working directory into the R workspace. Options are "No" (default), "Yes", and "Default". If "Default", global behaviour settings are inherited.
save_workspace	Save .RData on exit. Options are "No" (default), "Yes", and "Default". If "Default", global behaviour settings are inherited.
save_history	Always save history (even when not saving .RData). Options are "Default" (default), "Yes", and "No". If "Default", global behaviour settings are inherited.
enable_code_indexing	Determines whether R source files within the project directory are indexed for code navigation. Options are "Yes" (default), "No", and "Default". If "Default", global behaviour settings are inherited.
spaces_for_tab	Determine whether the tab key inserts multiple spaces rather than a tab character (soft tabs). Options are "Yes" (default), "No", and "Default". If "Default", global behaviour settings are inherited.
num_spaces_for_tab	Specify the number of spaces per soft-tab, integer.
encoding	Specify the default text encoding for source files. Default is "UTF-8".
rnw_weave	Specify how to weave Rnw files. Default is "Sweave".
latex	Specify LaTeX processing. Default is "pdfLaTeX".
auto_append_new_line	Ensure that source files end with a new line. Default is "Yes".
strip_trailing_white_space	Strip trailing horizontal white space when saving. Default is "Yes".

Value

The settings used to write a .Rproj file.

See Also

<https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects> for more information on these settings.

Examples

```
create_r_proj()
```

edit_r_profile	<i>Open .Rprofile for editing</i>
----------------	-----------------------------------

Description

edit_r_profile opens the .Rprofile file for editing. If the .Rprofile file doesn't exist an error message will be returned. This is essentially a wrapper function for file.edit.

Usage

```
edit_r_profile()
```

Value

A message that .Rprofile is being opened or that it doesn't exist.

FileCabinet	<i>R6 class for a cabinet</i>
-------------	-------------------------------

Description

Constructs an R6 class of FileCabinet. Objects of class FileCabinet contain information that is used by new_cabinet_proj() to create project directories.

Public fields

name cabinet name.

directory the path to where future directories will be created, a string.

structure the directory structure, a list.

Methods**Public methods:**

- `FileCabinet$new()`
- `FileCabinet$print()`

Method `new()`:*Usage:*`FileCabinet$new(name, directory, structure)`*Arguments:*

name cabinet name.

directory the path to where future directories will be created, a string.

structure the directory structure, a list.

Details: Create a new 'FileCabinet' object.*Returns:* A cabinet object.*Examples:*`FileCabinet$new("test", "a/path", list(code = NULL, 'data/derived' = NULL, 'data/source' = NULL))`**Method** `print()`:*Usage:*`FileCabinet$print()`*Details:* Print an object of class FileCabinet.**Examples**

```
## -----
## Method `FileCabinet$new`
## -----

FileCabinet$new("test", "a/path", list(code = NULL, 'data/derived' = NULL, 'data/source' = NULL))
```

`get_cabinets`*Print available cabinets*

Description`get_cabinets` returns objects of class FileCabinet found in the global environment.**Usage**`get_cabinets()`**Value**

Objects of class FileCabinet found in the global environment.

Examples

```
get_cabinets()
```

```
new_cabinet_proj      Create a new project using a cabinet template
```

Description

Generate new project directories using cabinet templates.

Usage

```
new_cabinet_proj(  
  cabinet,  
  project_name,  
  r_project = TRUE,  
  open = TRUE,  
  git = TRUE,  
  git_root = NULL,  
  git_ignore = NULL,  
  ...  
)
```

Arguments

<code>cabinet</code>	The name of the cabinet template. Available cabinets can be found using <code>get_cabinets()</code> .
<code>project_name</code>	The name of the project to store in the cabinet, a character string.
<code>r_project</code>	Logical, should an Rproject be created. Default is TRUE if working in RStudio (only works in RStudio).
<code>open</code>	Logical, if creating an Rproject, should that project be opened once created. Default is TRUE if working in RStudio (only works in RStudio).
<code>git</code>	Logical, should a git repository be initiated.
<code>git_root</code>	A path relative to the project to initiate the git repository. Default is NULL and the repository is initiated at the root of the project.
<code>git_ignore</code>	Character vector of files and directories to add to .gitignore file.
<code>...</code>	Extra arguments to pass to <code>create_r_proj</code> .

Value

Creates a new directory at the path specified in the cabinet template. If `r_project` is set to TRUE, a .Rproj file will also be created using the project name. If `open` is set to TRUE, the new R project will be opened in a new R session.

See Also

[create_cabinet](#)

Index

`cabinets`, [2](#)
`cabinets_options_set`, [2](#)
`create_cabinet`, [3](#), [7](#)
`create_r_proj`, [3](#)

`edit_r_profile`, [5](#)

`FileCabinet`, [5](#)

`get_cabinets`, [6](#)

`new_cabinet_proj`, [3](#), [7](#)