

Package ‘rprojroot’

January 3, 2018

Title Finding Files in Project Subdirectories

Version 1.3-2

Description Robust, reliable and flexible paths to files below a project root. The 'root' of a project is defined as a directory that matches a certain criterion, e.g., it contains a certain regular file.

Depends R (>= 3.0.0)

Imports backports

Suggests testthat, mockr, knitr, withr, rmarkdown

VignetteBuilder knitr

License GPL-3

LazyData true

Encoding UTF-8

URL <https://github.com/krlmlr/rprojroot>,
<https://krlmlr.github.io/rprojroot>

BugReports <https://github.com/krlmlr/rprojroot/issues>

RoxygenNote 6.0.1

Collate 'rmake.R' 'criterion.R' 'file.R' 'has-file.R' 'root.R'
'rprojroot-package.R' 'shortcut.R' 'thisfile.R'

NeedsCompilation no

Author Kirill Müller [aut, cre]

Maintainer Kirill Müller <krlmlr+r@mailbox.org>

Repository CRAN

Date/Publication 2018-01-03 15:36:24 UTC

R topics documented:

rprojroot-package	2
criteria	3
find_root	4
find_root_file	5
root_criterion	6
thisfile	8
Index	9

rprojroot-package	<i>rprojroot: Finding Files in Project Subdirectories</i>
-------------------	---

Description

Robust, reliable and flexible paths to files below a project root. The 'root' of a project is defined as a directory that matches a certain criterion, e.g., it contains a certain regular file.

Details

See the "Value" section in `root_criterion()` for documentation of root criterion objects, and `criteria()` for useful predefined root criteria.

Author(s)

Maintainer: Kirill Müller <krlmlr+r@mailbox.org>

See Also

Useful links:

- <https://github.com/krlmlr/rprojroot>
- <https://krlmlr.github.io/rprojroot>
- Report bugs at <https://github.com/krlmlr/rprojroot/issues>

Examples

```
criteria
## Not run:
is_r_package$find_file("NAMESPACE")
root_fun <- is_r_package$make_fix_file()
root_fun("NAMESPACE")

## End(Not run)
```

criteria	<i>Prespecified criteria</i>
----------	------------------------------

Description

This is a collection of commonly used root criteria.

Usage

```
criteria
is_rstudio_project
is_r_package
is_remake_project
is_projectile_project
is_git_root
is_svn_root
is_vcs_root
is_testthat
from_wd
```

Format

An object of class `root_criteria` of length 9.

Details

`is_rstudio_project` looks for a file with extension `.Rproj`.
`is_r_package` looks for a `DESCRIPTION` file.
`is_remake_project` looks for a `remake.yml` file.
`is_projectile_project` looks for a `.projectile` file.
`is_git_project` looks for a `.git` directory.
`is_svn_project` looks for a `.svn` directory.
`is_vcs_project` looks for the root of a version control system, currently only Git and SVN are supported.
`is_testthat` looks for the `testthat` directory, works when developing, testing, and checking a package.

from_wd uses the current working directory.

find_root	<i>Find the root of a directory hierarchy</i>
-----------	---

Description

A *root* is defined as a directory that contains a regular file whose name matches a given pattern and which optionally contains a given text. The search for a root starts at a given directory (the working directory by default), and proceeds up the directory hierarchy.

get_root_desc() returns the description of the criterion for a root path. This is especially useful for composite root criteria created with `|.root_criterion()`.

Usage

```
find_root(criterion, path = ".")
```

```
get_root_desc(criterion, path)
```

Arguments

criterion	A criterion, will be coerced using <code>as.root_criterion()</code>
path	The start directory

Details

Starting from the working directory, the `find_root()` function searches for the root. If a root is found, the `...` arguments are used to construct a path; thus, if no extra arguments are given, the root is returned. If no root is found, an error is thrown.

Value

The normalized path of the root as specified by the search criterion. Throws an error if no root is found

See Also

`utils::glob2rx()` `file.path()`

Examples

```
## Not run:
find_root(glob2rx("DESCRIPTION"), "^Package: ")

## End(Not run)
```

find_root_file	<i>File paths relative to the root of a directory hierarchy</i>
----------------	---

Description

Append an arbitrary number of path components to the root using `base::file.path()`.

Usage

```
find_root_file(..., criterion, path = ".")
```

```
find_rstudio_root_file(..., path = ".")
```

```
find_package_root_file(..., path = ".")
```

```
find_remake_root_file(..., path = ".")
```

```
find_testthat_root_file(..., path = ".")
```

Arguments

<code>...</code>	Further path components passed to <code>file.path()</code>
<code>criterion</code>	A criterion, will be coerced using <code>as.root_criterion()</code>
<code>path</code>	The start directory

Details

The `find_root_file()` function is a simple wrapper around `find_root()` that appends an arbitrary number of path components to the root using `base::file.path()`.

Value

The normalized path of the root as specified by the search criteria, with the additional path components appended. Throws an error if no root is found

See Also

`find_root()` `utils::glob2rx()` `base::file.path()`

Examples

```
## Not run:  
find_package_root_file("tests", "testthat.R")  
has_file("DESCRIPTION", "^Package: ")$find_file  
has_file("DESCRIPTION", "^Package: ")$make_fix_file(".")  
  
## End(Not run)
```

root_criterion	<i>Is a directory the project root?</i>
----------------	---

Description

Objects of the root_criterion class decide if a given directory is a project root.

Usage

```

root_criterion(testfun, desc, subdir = NULL)

is.root_criterion(x)

as.root_criterion(x)

## S3 method for class 'character'
as.root_criterion(x)

## S3 method for class 'root_criterion'
as.root_criterion(x)

## S3 method for class 'root_criterion'
x | y

has_file(filepath, contents = NULL, n = -1L)

has_dir(filepath)

has_file_pattern(pattern, contents = NULL, n = -1L)

has_dirname(dirname, subdir = NULL)

```

Arguments

testfun	A function with one parameter that returns TRUE if the directory specified by this parameter is the project root, and FALSE otherwise. Can also be a list of such functions.
desc	A textual description of the test criterion, of the same length as testfun
subdir	Subdirectories to start the search in, if found
x	An object
y	An object
filepath	File path (can contain directories)
contents	Regular expression to match the file contents
n	integer. The (maximal) number of lines to read. Negative values indicate that one should read up to the end of input on the connection.

pattern	Regular expression to match the file name
dirname	A directory name, without subdirectories

Details

Construct criteria using `root_criterion` in a very general fashion by specifying a function with a path argument, and a description.

The `as.root_criterion()` function accepts objects of class `root_criterion`, and character values; the latter will be converted to criteria using `has_file`.

Root criteria can be combined with the `|` operator. The result is a composite root criterion that requires either of the original criteria to match.

The `has_file()` function constructs a criterion that checks for the existence of a specific file (which itself can be in a subdirectory of the root) with specific contents.

The `has_dir()` function constructs a criterion that checks for the existence of a specific directory.

The `has_file_pattern()` function constructs a criterion that checks for the existence of a file that matches a pattern, with specific contents.

The `has_dirname()` function constructs a criterion that checks if the `base::dirname()` has a specific name.

Value

An S3 object of class `root_criterion` with the following members:

`testfun` The `testfun` argument

`desc` The `desc` argument

`subdir` The `subdir` argument

`find_file` A function with `...` argument that returns for a path relative to the root specified by this criterion. The optional path argument specifies the starting directory, which defaults to `"."`.

`make_fix_file` A function with a path argument that returns a function that finds paths relative to the root. For a criterion `cr`, the result of `cr$make_fix_file(".")(...)` is identical to `cr$find_file(...)`. The function created by `make_fix_file` can be saved to a variable to be more independent of the current working directory.

Examples

```
root_criterion(function(path) file.exists(file.path(path, "somefile")), "has somefile")
has_file("DESCRIPTION")
is_r_package
is_r_package$find_file
## Not run:
is_r_package$make_fix_file(".")

## End(Not run)
```

thisfile	<i>Determines the path of the currently running script</i>
----------	--

Description

R does not store nor export the path of the currently running script. This is an attempt to circumvent this limitation by applying heuristics (such as call stack and argument inspection) that work in many cases. **CAVEAT:** Use this function only if your workflow does not permit other solution: if a script needs to know its location, it should be set outside the context of the script if possible.

Usage

```
thisfile()
thisfile_source()
thisfile_r()
thisfile_rscript()
thisfile_knit()
```

Details

This functions currently work only if the script was sourced, processed with knitr, or run with Rscript or using the `--file` parameter to the R executable. For code run with Rscript, the exact value of the parameter passed to Rscript is returned.

Value

The path of the currently running script, NULL if it cannot be determined.

Author(s)

Kirill Müller, Hadley Wickham, Michael R. Head

References

<http://stackoverflow.com/q/1815606/946850>

See Also

[base::source\(\)](#), [utils::Rscript\(\)](#), [base::getwd\(\)](#)

Examples

```
## Not run: thisfile()
```


Index

*Topic **datasets**

- criteria, 3
- as.root_criterion (root_criterion), 6
- as.root_criterion(), 4, 5
- base::dirname(), 7
- base::file.path(), 5
- base::getwd(), 8
- base::source(), 8
- criteria, 3
- criteria(), 2
- file.path(), 4, 5
- find_package_root_file
 (find_root_file), 5
- find_remake_root_file (find_root_file),
 5
- find_root, 4
- find_root(), 5
- find_root_file, 5
- find_rstudio_root_file
 (find_root_file), 5
- find_testthat_root_file
 (find_root_file), 5
- from_wd (criteria), 3
- get_root_desc (find_root), 4
- has_dir (root_criterion), 6
- has_dirname (root_criterion), 6
- has_file (root_criterion), 6
- has_file_pattern (root_criterion), 6
- is.root_criterion (root_criterion), 6
- is_git_root (criteria), 3
- is_projectile_project (criteria), 3
- is_r_package (criteria), 3
- is_remake_project (criteria), 3
- is_rstudio_project (criteria), 3
- is_svn_root (criteria), 3
- is_testthat (criteria), 3
- is_vcs_root (criteria), 3
- root_criterion, 6
- root_criterion(), 2
- rprojroot (rprojroot-package), 2
- rprojroot-package, 2
- thisfile, 8
- thisfile_knit (thisfile), 8
- thisfile_r (thisfile), 8
- thisfile_rscript (thisfile), 8
- thisfile_source (thisfile), 8
- utils::glob2rx(), 4, 5
- utils::Rscript(), 8