

Package ‘attachment’

May 5, 2019

Title Deal with Dependencies

Version 0.0.9

Description Tools to help manage dependencies during package development. This can retrieve all dependencies that are used in R files in the ``R" directory, in Rmd files in ``vignettes" directory and in 'roxygen2' documentation of functions. There is a function to update the Description file of your package and a function to create a file with the R commands to install all dependencies of your package. All functions to retrieve dependencies of R scripts and Rmd files can be used independently of a package development.

License GPL-3

URL <https://github.com/Thinkr-open/attachment>

BugReports <https://github.com/Thinkr-open/attachment/issues>

Depends R (>= 3.3)

Imports desc (>= 1.2.0), devtools (>= 1.13.6), glue (>= 1.3.0), knitr (>= 1.20), magrittr (>= 1.5), rmarkdown (>= 1.10), stats, stringr (>= 1.3.1), utils

Suggests covr, pkgdown (>= 1.3.0), testthat (>= 2.0.0)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Vincent Guyader [cre, aut] (<<https://orcid.org/0000-0003-0671-9270>>),
Sébastien Rochette [aut] (<<https://orcid.org/0000-0002-1565-9313>>),
ThinkR [cph, fnd]

Maintainer Vincent Guyader <vincent@thinkr.fr>

Repository CRAN

Date/Publication 2019-05-05 08:20:03 UTC

R topics documented:

| | |
|------------------------------------|----------|
| att_from_description | 2 |
| att_from_namespace | 3 |
| att_from_rmd | 3 |
| att_from_rmds | 4 |
| att_from_rscript | 5 |
| att_from_rscripts | 5 |
| att_to_description | 6 |
| create_dependencies_file | 7 |
| install_from_description | 7 |
| install_if_missing | 8 |
| Index | 9 |

att_from_description *Return all package dependencies from current package*

Description

Return all package dependencies from current package

Usage

```
att_from_description(path = "DESCRIPTION", dput = FALSE,
  field = c("Depends", "Imports", "Suggests"))
```

Arguments

path path to the DESCRIPTION file
dput if FALSE return a vector instead of dput output
field DESCRIPTION field to parse, Import and Depends by default

Value

A character vector with packages names

Examples

```
dummyspackage <- system.file("dummyspackage",package = "attachment")
# browseURL(dummyspackage)
att_from_description(path = file.path(dummyspackage,"DESCRIPTION"))
```

att_from_namespace *return package dependencies from NAMESPACE file*

Description

return package dependencies from NAMESPACE file

Usage

```
att_from_namespace(path = "NAMESPACE", document = TRUE)
```

Arguments

| | |
|----------|---|
| path | path to NAMESPACE file |
| document | Run function document of devtools package |

Value

a vector

Examples

```
tmpdir <- tempdir()
file.copy(system.file("dummypackage", package = "attachment"), tmpdir,
  recursive = TRUE)
dummypackage <- file.path(tmpdir, "dummypackage")
# browseURL(dummypackage)
att_from_namespace(path = file.path(dummypackage, "NAMESPACE"))
```

att_from_rmd *Get all dependencies from a Rmd file*

Description

Get all dependencies from a Rmd file

Usage

```
att_from_rmd(path, temp_dir = tempdir(), warn = -1,
  encoding = getOption("encoding"))
```

Arguments

| | |
|----------|--|
| path | Path to a Rmd file |
| temp_dir | Path to temporary script from purl vignette |
| warn | -1 for quiet warnings with purl, 0 to see warnings |
| encoding | Encoding of the input file; see file . |

Examples

```
dummyspackage <- system.file("dummyspackage",package = "attachment")  
# browseURL(dummyspackage)  
att_from_rmd(path = file.path(dummyspackage,"vignettes/demo.Rmd"))
```

| | |
|---------------|--|
| att_from_rmds | <i>Get all packages called in vignettes folder</i> |
|---------------|--|

Description

Get all packages called in vignettes folder

Usage

```
att_from_rmds(path = "vignettes", recursive = TRUE, warn = -1)
```

Arguments

| | |
|-----------|---|
| path | path to directory with Rmds or vector of Rmd files |
| recursive | logical. Should the listing recurse into directories? |
| warn | -1 for quiet warnings with purl, 0 to see warnings |

Value

Character vector of packages called with library or require. knitr and rmarkdown are added by default to allow building the vignettes if the directory contains "vignettes" in the path

Examples

```
dummyspackage <- system.file("dummyspackage",package = "attachment")  
# browseURL(dummyspackage)  
att_from_rmds(path = file.path(dummyspackage,"vignettes"))
```

| | |
|------------------|---|
| att_from_rscript | <i>Look for functions called with '::' and library/requires in one script</i> |
|------------------|---|

Description

Look for functions called with '::' and library/requires in one script

Usage

```
att_from_rscript(path)
```

Arguments

| | |
|------|-----------------------|
| path | path to R script file |
|------|-----------------------|

Details

Calls from pkg::fun in roxygen skeleton and comments are ignored

Value

a vector

Examples

```
dummypackage <- system.file("dummypackage", package = "attachment")
# browseURL(dummypackage)

att_from_rscript(path = file.path(dummypackage, "R", "my_mean.R"))
```

| | |
|-------------------|--|
| att_from_rscripts | <i>Look for functions called with '::' and library/requires in folder of scripts</i> |
|-------------------|--|

Description

Look for functions called with '::' and library/requires in folder of scripts

Usage

```
att_from_rscripts(path = "R", pattern = "*.r|R$", recursive = TRUE)
```

Arguments

| | |
|-----------|---|
| path | directory with R scripts inside |
| pattern | pattern to detect R script files |
| recursive | logical. Should the listing recurse into directories? |

Examples

```
dummypackage <- system.file("dummypackage",package = "attachment")
# browseURL(dummypackage)

att_from_rscripts(path = dummypackage)
```

att_to_description *Add packages to description*

Description

Add packages to description

Usage

```
att_to_description(path = ".", path.n = "NAMESPACE",
  path.d = "DESCRIPTION", dir.r = "R", dir.v = "vignettes",
  dir.t = "tests", extra.suggests = NULL, pkg_ignore = NULL,
  document = TRUE)
```

Arguments

| | |
|----------------|--|
| path | path to the root of the package directory. Default to current directory. |
| path.n | path to namespace file. |
| path.d | path to description file. |
| dir.r | path to directory with R scripts. |
| dir.v | path to vignettes directory. Set to empty (dir.v = "") to ignore. |
| dir.t | path to tests directory. Set to empty (dir.t = "") to ignore. |
| extra.suggests | vector of other packages that should be added in Suggests (pkgdown, covr for instance) |
| pkg_ignore | vector of packages names to ignore. |
| document | Run function document of devtools package |

Examples

```
tmpdir <- tempdir()
file.copy(system.file("dummypackage",package = "attachment"), tmpdir,
  recursive = TRUE)
dummypackage <- file.path(tmpdir, "dummypackage")
# browseURL(dummypackage)
att_to_description(path = dummypackage)
```

`create_dependencies_file`*Create a dependencies.R in the 'inst' folder*

Description

Create a dependencies.R in the 'inst' folder

Usage

```
create_dependencies_file(path = "DESCRIPTION", field = c("Depends",
  "Imports"), to = "inst/dependencies.R", open_file = TRUE,
  ignore_base = TRUE)
```

Arguments

| | |
|--------------------------|---|
| <code>path</code> | path to the DESCRIPTION file |
| <code>field</code> | DESCRIPTION field to parse, "Import" and "Depends" by default. Can add "Suggests" |
| <code>to</code> | path to dependencies.R. "inst/dependencies.R" by default |
| <code>open_file</code> | Logical. Open the file created in an editor |
| <code>ignore_base</code> | Logical. Whether to ignore package coming with base, as they cannot be installed |

Examples

```
tmpdir <- tempdir()
file.copy(system.file("dummyspackage", package = "attachment"), tmpdir,
  recursive = TRUE)
dummyspackage <- file.path(tmpdir, "dummyspackage")
# browseURL(dummyspackage)

create_dependencies_file(path = file.path(dummyspackage, "DESCRIPTION"),
  to = file.path(dummyspackage, "inst/dependencies.R"),
  open_file = FALSE)
```

`install_from_description`*Install missing package from DESCRIPTION*

Description

Install missing package from DESCRIPTION

Usage

```
install_from_description(path = "DESCRIPTION", field = c("Depends",  
  "Imports", "Suggests"), ...)
```

Arguments

| | |
|-------|--|
| path | path to the DESCRIPTION file |
| field | DESCRIPTION fields to parse, "Depends", "Imports", "Suggests" by default |
| ... | Arguments to be passed to install.packages |

Examples

```
## Not run:  
dummypackage <- system.file("dummypackage", package = "attachment")  
# browseURL(dummypackage)  
  
install_from_description(path = file.path(dummypackage, "DESCRIPTION"))  
  
## End(Not run)
```

install_if_missing *install packages if missing*

Description

install packages if missing

Usage

```
install_if_missing(to_be_installed, ...)
```

Arguments

| | |
|-----------------|--|
| to_be_installed | a character vector containing required packages names |
| ... | Arguments to be passed to install.packages |

Examples

```
## Not run:  
install_if_missing(c("dplyr", "fcuk", "rusk"))  
  
## End(Not run)
```


Index

`att_from_description`, 2

`att_from_namespace`, 3

`att_from_rmd`, 3

`att_from_rmds`, 4

`att_from_rscript`, 5

`att_from_rscripts`, 5

`att_to_description`, 6

`create_dependencies_file`, 7

`file`, 3

`install.packages`, 8

`install_from_description`, 7

`install_if_missing`, 8