Package ‘fakemake’

February 23, 2020

Title  Mock the Unix Make Utility
Version  1.7.0
Description  Use R as a minimal build system. This might come in handy if you are developing R packages and can not use a proper build system. Stay away if you can (use a proper build system).
License  BSD_2_clause + file LICENSE
URL  https://gitlab.com/fvafrcu/fakemake
Depends  R (>= 3.3.0)
Imports  callr, graphics, igraph, MakefileR, utils, withr
Suggests  cleanr, covr, cyclocomp, devtools, hunspell, knitr, lintr, pkgbuild, pkgload, rmarkdown, roxygen2, rprojroot, RUnit, spelling, testthat, usethis
VignetteBuilder  knitr
Encoding  UTF-8
LazyData  true
RoxygenNote  7.0.2
NeedsCompilation  no
Author  Andreas Dominik Cullmann [aut, cre]
Maintainer  Andreas Dominik Cullmann <fvafrcu@mailbox.org>
Repository  CRAN
Date/Publication  2020-02-23 16:20:02 UTC

R topics documented:

  fakemake-package ....................................................... 2
  add_target ............................................................... 2
  get_target ............................................................. 3
  make ................................................................. 4
  provide_make_list ..................................................... 5
  remove_target ......................................................... 6
  visualize ............................................................. 6
add_target

Index

fakemake-package	Mock the Unix Make Utility

Description

Use R as a minimal build system. This might come in handy if you are developing R packages and can not use a proper build system. Stay away if you can (use a proper build system).

Details

You will find the details in vignette("An_Introduction_to_fakemake",package = "fakemake").

add_target	Add a Target to a Makelist

Description

Add a target to an existing makelist.

Usage

add_target(
  makelist,
  target,
  code,
  prerequisites = NULL,
  prerequisite_to = NULL,
  sink = NULL,
  alias = sub("\.(Rout|log)$", "", basename(target))
)

Arguments

makelist	A list for make.
target	The target to remove from makelist.
code	The code for the new target.
prerequisites	The prerequisites for the new target.
prerequisite_to	The targets the new target is a prerequisite to. Set to TRUE to add it as a prerequisite to all existing targets.
sink	The sink for the new target.
alias	The alias for the new target.
get_target

Value

A list for make.

See Also

Other functions to manipulate makelists: get_target(), remove_target()

get_target  Get a Makelist's Target

Description

Get a single target from a makelist by alias.

Usage

get_target(makelist, alias)

Arguments

makelist A list for make.
alias The alias of the target in question.

Value

A list (the target requested).

See Also

Other functions to manipulate makelists: add_target(), remove_target()

Examples

ml <- provide_make_list()
visualize(ml, root = "all.Rout")
i <- which(sapply(ml, "[[", "target") == "b1.Rout")
ml[[i]][['alias']] <- "b1"
t <- get_target(ml, "b1")
ml <- remove_target(ml, t[['target']])
visualize(ml)
ml <- add_target(ml, target = t[['target']],
                 code = t[['code']],
                 sink = t[['sink']],
                 prerequisite_to = "a1.Rout", alias = NULL)
all.equal(ml, provide_make_list())
Mock the Unix Make Utility

Usage

```r
make(
   name,
   make_list,
   force = FALSE,
   recursive = force,
   verbose = FALSE,
   dry_run = FALSE
)
```

Arguments

- `name`: The name or alias of a make target.
- `make_list`: The makelist (a listed version of a Makefile).
- `force`: Force the target to be build?
- `recursive`: Force the target to be build recursively (see Note)?
- `verbose`: Be verbose?
- `dry_run`: Run dry? Mock make's -n option.

Value

A character vector containing the targets made during the current run.

Note

Forcing a target mocks adding .PHONY to a GNU Makefile if you set recursive to FALSE. If recursive is TRUE, then the whole make chain will be forced.

Examples

```r
str(make_list <- provide_make_list("minimal"))
# build all
withr::with_dir(tempdir(), print(make("all.Rout", make_list)))
# nothing to be done
withr::with_dir(tempdir(), print(make("all.Rout", make_list)))
# forcing all.Rout
withr::with_dir(tempdir(), print(make("all.Rout", make_list, force = TRUE, recursive = FALSE)))
# forcing all.Rout recursively
```
provide_make_list

Load an Example Makelist Provided by fakemake.

Description

Load an Example Makelist Provided by fakemake.

Usage

provide_make_list(
  type = c("minimal", "cran", "vignette", "standard", "package"),
  prune = TRUE,
  clean_sink = FALSE
)

Arguments

type
  The type of makelist. package makelist.

prune
  Prune the makelist of NULL items?

clean_sink
  Remove sinks identical to corresponding targets from the list? Since makelists
  are parsed, missing sinks are set to the corresponding targets, but this makes
  them harder to read.

Value

A makelist.

Examples

str(provide_make_list("minimal"))
visualize(provide_make_list("minimal"))
remove_target  Remove a Target From a Makelist

Description
Remove a target and all its appearances as other targets’ dependencies from a makelist.

Usage
remove_target(makelist, target)

Arguments
makelist A list for make.

Argument
target The target to remove from makelist.

Value
A list for make.

See Also
Other functions to manipulate makelists: add_target(), get_target()

visualize  Visualize a Makelist

Description
Parse a makelist, convert it into an igraph and plot it.

Usage
visualize(make_list, root = NULL)

Arguments
make_list The makelist.

Argument
root The root of a tree.

Value
Invisibly an igraph representation of the makelist.
Examples

```r
str(ml <- provide_make_list())
visualize(ml)
visualize(ml, root = "all.Rout")
```
Index

*Topic package
  fakemake-package, 2

add_target, 2, 3, 6
fakemake-package, 2
get_target, 3, 3, 6
make, 2, 3, 4, 6
provide_make_list, 5
remove_target, 3, 6
TRUE, 2
visualize, 6