

Package ‘disposables’

March 20, 2017

Title Create Disposable R Packages for Testing

Version 1.0.3

Author Gábor Csárdi

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Description Create disposable R packages for testing.

You can create, install and load multiple R packages with a single function call, and then unload, uninstall and destroy them with another function call. This is handy when testing how some R code or an R package behaves with respect to other packages.

License MIT + file LICENSE

LazyData true

URL <https://github.com/gaborcsardi/disposables>

BugReports <https://github.com/gaborcsardi/disposables/issues>

Imports utils

Suggests covr, testthat

Depends methods

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-20 12:30:58 UTC

R topics documented:

dispose_packages	2
make_packages	3
Index	5

dispose_packages *Get rid of temporary packages*

Description

Get rid of temporary packages

Usage

```
dispose_packages(packages, unattach = TRUE, unload = unattach,
  delete = TRUE, delete_lib_dir = delete)
```

Arguments

packages	A list returned by make_packages .
unattach	Whether to unattach the packages.
unload	Whether to unload the packages. It is not possible to unload without unattaching.
delete	Whether to delete the installed packages from the lib_dir. If delete_lib_dir is TRUE, then this should be TRUE as well.
delete_lib_dir	Whether to delete the the whole lib_dir.

Examples

```
pkg <- make_packages(
  foo1 = { f <- function() print("hello!") ; d <- 1:10 },
  foo2 = { f <- function() print("hello again!") ; d <- 11:20 }
)

foo1::f()
foo2::f()
foo1::d
foo2::d

## Unattach only
dispose_packages(pkg, unload = FALSE, delete = FALSE)
"package:foo1" %in% search()
"foo1" %in% loadedNamespaces()
dir(pkg$lib_dir)

## Unload
dispose_packages(pkg, delete = FALSE)
"package:foo1" %in% search()
"foo1" %in% loadedNamespaces()
dir(pkg$lib_dir)

## Delete completely
```

```
dispose_packages(pkg)
"package:foo1" %in% search()
"foo1" %in% loadedNamespaces()
file.exists(pkg$lib_dir)
```

make_packages	<i>Create, install, load and attach multiple disposable packages</i>
---------------	--

Description

If a package with the same name as a disposable one, is loaded, then it will be unloaded. If a package with same name as a disposable one is installed in `lib_dir`, then it will be overwritten. (`lib_dir` is usually a temporary directory, so this is not a big problem.)

Usage

```
make_packages(..., lib_dir = tempfile(), imports = character(),
  quiet = TRUE)
```

Arguments

<code>...</code>	Named expressions. A separate package with the given name is created for each.
<code>lib_dir</code>	Directory to install the package to. Defaults to a temporary directory that is deleted once the R session is over.
<code>imports</code>	The 'Imports' field in the DESCRIPTION file, the packages to import in each disposable package. It can be a character vector, which applies to all packages, or a list of character vectors, one for each package.
<code>quiet</code>	Whether to show the installation process of disposable packages.

Details

Note that if you specify `lib_dir` and it points to an existing directory, `make_package` overwrites the packages there. If an error happens during installation or loading of the disposables packages, then it will *not* restore the original contents of `lib_dir`, but it will remove all newly installed disposable packages, even the ones that were installed cleanly.

Value

A named list with entries:

- `lib_dir` The directory in which the packages are installed.
- `package` The named of the packages.

Examples

```
pkg <- make_packages(  
  foo1 = { f <- function() print("hello!") ; d <- 1:10 },  
  foo2 = { f <- function() print("hello again!") ; d <- 11:20 }  
)  
foo1::f()  
foo2::f()  
foo1::d  
foo2::d  
dispose_packages(pkg)
```

See Also

[dispose_packages](#)

Index

dispose_packages, [2](#), [4](#)

make_packages, [2](#), [3](#)