Package 'roxygen2'

September 2, 2014

Title In-source documentation for R

Description A Doxygen-like in-source documentation system for Rd, collation, and NAMESPACE.

URL https://github.com/klutometis/roxygen

Version 4.0.2

License GPL (>= 2)

Depends R (>= 3.0.2)

Imports stringr (>= 0.5), brew, digest, methods, Rcpp (>= 0.11.0)

Suggests testthat (>= 0.8.0), knitr

VignetteBuilder knitr

LinkingTo Rcpp

Collate 'RcppExports.R' 'alias.R' 'description.R' 'family.R'
'inherit-params.R' 'minidesc.R' 'object-defaults.R' 'object-from-call.R' 'object.R' 'order-params.R'
'parse-preref.R' 'parse-registry.R' 'parse.R' 'rc.R'
'rd-escape.R' 'rd-file-api.R' 'rd-parse.R' 'rd-tag-api.R'
'roclet-collate.R' 'roclet-namespace.R' 'roclet-rd.R'
'roclet.R' 'roxygen.R' 'roxygenize.R' 's3.R' 'safety.R'
'source.R' 'template.R' 'topic-name.R' 'topo-sort.R' 'usage.R' 'util-locale.R' 'utils.R'

Author Hadley Wickham [aut, cre, cph],Peter Danenberg [aut, cph],Manuel Eugster [aut, cph],RStudio [cph]

Maintainer Hadley Wickham <h.wickham@gmail.com>

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-09-02 19:05:51

R topics documented:

`is_s3_generic` *Determine if a function is an S3 generic or S3 method.*

Description

`is_s3_generic` compares name to `.knownS3Generics` and `.S3PrimitiveGenerics`, then looks at the function body to see if it calls `UseMethod`.

`is_s3_method` builds names of all possible generics for that function and then checks if any of them actually is a generic.

Usage

```
is_s3_generic(name, env = parent.frame())

is_s3_method(name, env = parent.frame())
```

Arguments

`name`	Name of function.
`env`	Base environment in which to look for function defintion.

`namespace_roclet` *Roclet: make NAMESPACE.*

Description

This roclet automates the production of a 'NAMESPACE' file, see *Writing R Extensions* (http://cran.r-project.org/doc/manuals/R-exts.pdf) for details.

Usage

```
namespace_roclet()
```

See Also

```
vignette("namespace", package = "roxygen2")
```

Other roclets: `rd_roclet`

rd_roclet *Roclet: make Rd files.*

Description

This roclet is the workhorse of **roxygen**, producing the Rd files that document that functions in your package.

Usage

```
rd_roclet()
```

See Also

```
vignette("rd", package = "roxygen2")
```

Other roclets: S3method, export, exportClass, exportMethod, import, importClassesFrom, importFrom, importMethodsFrom, namespace_roclet

roxygen *In-line documentation for R.*

Description

Roxygen is a Doxygen-like documentation system for R; allowing in-source specification of Rd files, collation and namespace directives.

Details

If you have existing Rd files, check out the Rd2roxygen package for a convenient way of converting Rd files to roxygen comments.

Author(s)

Hadley Wickham <h.wickham@gmail.com>, Peter Danenberg <pcd@roxygen.org>, Manuel Eugster <Manuel.Eugster@stat.uni-muenchen.de>

Maintainer: Hadley Wickham <h.wickham@gmail.com>

See Also

See vignette("roxygen2", package = "roxygen2") for an overview of the package, vignette("rd", package = "roxy for generating documentation, and vignette("namespace", package = "roxygen2") for generating the namespace specification.

Examples

```
## Not run: roxygenize('pkg')
```

roxygenize *Process a package with the Rd, namespace and collate roclets.*

Description

This is the workhorse function that uses roclets, the built-in document tranformation functions, to build all documentation for a package. See the documentation for the individual roclets, rd_roclet, namespace_roclet, and for update_collate, for more details.

Usage

```
roxygenize(package.dir = ".", roclets = NULL, load_code = source_package,
  clean = FALSE)

roxygenise(package.dir = ".", roclets = NULL, load_code = source_package,
  clean = FALSE)
```

Arguments

package.dir	Location of package top level directory. Default is working directory.
roclets	Character vector of roclet names to use with package. This defaults to NULL, which will use the roclets fields in the list provided in the Roxygen DESCRIPTION field. If none are specified, defaults to c("collate", "namespace", "rd").
load_code	A function used to load all the R code in the package directory. It is called with the path to the package, and it should return an environment containing all the sourced code.
clean	If TRUE, roxygen will delete all files previously created by roxygen before running each roclet.

Details

Note that roxygen2 is a dynamic documentation system: it works using by inspecting loaded objects in the package. This means that you must be able to load the package in order to document it. source_package provides a simple simulation of package loading that works if you only have R files in your package. For more complicated packages, I recommend using devtools::document which does a much better job at simulating package install and load.

Value

NULL

update_collate *Update Collate field in DESCRIPTION.*

Description

Topologically sort R files and record in Collate field. The topological sort is based on the `@include` tag, which should specify the filenames (space separated) that should be loaded before the current file - these are typically necessary if you're using S4 or RC classes (because super classes must be defined before subclasses). If there are no `@include` tags Collate will be left blank, indicating that the order of loading does not matter.

Usage

```
update_collate(base_path)
```

Arguments

base_path	Path to package directory.

Details

This is not a roclet because roclets need the values of objects in a package, and those values can not be generated unless you've sourced the files, and you can't source the files unless you know the correct order.

Examples

```
#' `example-a.R', `example-b.R' and `example-c.R' reside
#' in the `example' directory, with dependencies
#' a -> {b, c}. This is `example-a.R'.
#' @include example-b.R
#' @include example-c.R
NULL

## Not run:
  update_collate("my_package")

## End(Not run)
```

Index