

# Package ‘roxygen2’

February 6, 2017

**Title** In-Line Documentation for R

**Description** Generate your Rd documentation, 'NAMESPACE' file, and collation field using specially formatted comments. Writing documentation in-line with code makes it easier to keep your documentation up-to-date as your requirements change. 'Roxygen2' is inspired by the 'Doxygen' system for C++.

**Version** 6.0.1

**License** GPL (>= 2)

**Depends** R (>= 3.0.2)

**Imports** stringr (>= 0.5), stringi, brew, digest, methods, Rcpp (>= 0.11.0), R6, desc, commonmark, xml2, utils

**Suggests** testthat (>= 0.8.0), knitr, devtools, rmarkdown, covr

**VignetteBuilder** knitr

**LinkingTo** Rcpp

**RoxygenNote** 6.0.1

**URL** <https://github.com/klutometis/roxygen>

**BugReports** <https://github.com/klutometis/roxygen/issues>

**NeedsCompilation** yes

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

**Maintainer** Hadley Wickham <hadley@rstudio.com>

**Repository** CRAN

**Date/Publication** 2017-02-06 16:37:05

## R topics documented:

|                            |   |
|----------------------------|---|
| roxygen2-package . . . . . | 2 |
| double_escape_md . . . . . | 3 |
| is_s3_generic . . . . .    | 3 |

|                            |   |
|----------------------------|---|
| namespace_roclet . . . . . | 4 |
| object_format . . . . .    | 4 |
| rd_roclet . . . . .        | 5 |
| roclet_find . . . . .      | 5 |
| roxygenize . . . . .       | 6 |
| update_collate . . . . .   | 7 |
| vignette_roclet . . . . .  | 8 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>9</b> |
|--------------|----------|

---

roxygen2-package      *roxygen2: In-Line Documentation for R*

---

## Description

Generate your Rd documentation, 'NAMESPACE' file, and collation field using specially formatted comments. Writing documentation in-line with code makes it easier to keep your documentation up-to-date as your requirements change. 'Roxygen2' is inspired by the 'Doxygen' system for C++.

## Details

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

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

## Author(s)

**Maintainer:** Hadley Wickham <hadley@rstudio.com> [copyright holder]

Authors:

- Peter Danenberg <pcd@roxygen.org> [copyright holder]
- Manuel Eugster [copyright holder]

Other contributors:

- RStudio [copyright holder]

## See Also

Useful links:

- <https://github.com/klutometis/roxygen>
- Report bugs at <https://github.com/klutometis/roxygen/issues>

---

|                  |   |
|------------------|---|
| double_escape_md | <i>Escape % and \\$ and \_ once more, because commonmark removes the escaping. We do this everywhere currently.</i> |
|------------------|---|

---

**Description**

Escape % and \\$ and \\_ once more, because commonmark removes the escaping. We do this everywhere currently.

**Usage**

```
double_escape_md(text)
```

**Arguments**

|      |             |
|------|-------------|
| text | Input text. |
|------|-------------|

**Value**

Double-escaped text.

---

|               |   |
|---------------|---|
| is_s3_generic | <i>Determine if a function is an S3 generic or S3 method.</i> |
|---------------|---|

---

**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 definition. |

---

|                  |                                |
|------------------|--------------------------------|
| namespace_roclet | <i>Roclet: make NAMESPACE.</i> |
|------------------|--------------------------------|

---

### Description

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

### Usage

```
namespace_roclet()
```

### See Also

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

Other roclets: [rd\\_roclet](#), [vignette\\_roclet](#)

---

|               |                                |
|---------------|--------------------------------|
| object_format | <i>Default format for data</i> |
|---------------|--------------------------------|

---

### Description

This function is called to generate the default "Format" section for each data object. The default implementation will return the class and dimension information.

### Usage

```
object_format(x)
```

### Arguments

|   |               |
|---|---------------|
| x | A data object |
|---|---------------|

### Value

A character value with valid Rd syntax, or NULL.

---

|           |                               |
|-----------|-------------------------------|
| rd_roclet | <i>Roclet: make Rd files.</i> |
|-----------|-------------------------------|

---

### 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: [namespace\\_roclet](#), [vignette\\_roclet](#)

---

|             |                                       |
|-------------|---------------------------------------|
| roclet_find | <i>Create a roclet from a string.</i> |
|-------------|---------------------------------------|

---

### Description

This provides a flexible way of specifying a roclet in a string.

### Usage

```
roclet_find(x)
```

### Arguments

x                   Arbitrary R code evaluated in roxygen2 package.

### Examples

```
# rd, namespace, and vignette work for backward compatibility
roclet_find("rd")

# But generally you should specify the name of a function that
# returns a roclet
roclet_find("rd_roclet")

# If it lives in another package, you'll need to use ::
roclet_find("roxygen2::rd_roclet")

# If it takes parameters (which no roclet does currently), you'll need
# to call the function
roclet_find("roxygen2::rd_roclet()")
```

---

 roxygenize

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


---

## Description

This is the workhorse function that uses roclets, the built-in document transformation 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

|                          |  |
|--------------------------|--|
| <code>package.dir</code> | Location of package top level directory. Default is working directory.   |
| <code>roclets</code>     | Character vector of roclet names to use with package. This defaults to <code>NULL</code> , which will use the <code>roclets</code> fields in the list provided in the Roxygen DESCRIPTION field. If none are specified, defaults to <code>c("collate", "namespace", "rd")</code> . |
| <code>load_code</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.   |
| <code>clean</code>       | If <code>TRUE</code> , 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 | <i>Update Collate field in DESCRIPTION.</i> |
|----------------|---|

---

### 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).

### Usage

```
update_collate(base_path)
```

### Arguments

base\_path      Path to package directory.

### Details

If there are no @include tags, roxygen2 will leave collate as is. This makes it easier to use roxygen2 with an existing collate directive, but if you remove all your @include tags, you'll need to also manually delete the collate field.

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)
```

---

|                 |                                     |
|-----------------|-------------------------------------|
| vignette_roclet | <i>Re-build outdated vignettes.</i> |
|-----------------|-------------------------------------|

---

### Description

This rebuilds outdated vignettes with [tools::buildVignette](#). By default, it will rebuild all vignettes if the source file is newer than the output pdf or html. (This means it will automatically re-build the vignette if you change the vignette source, but *not* when you change the R code). If you want finer control, add a Makefile to vignettes/ and roxygen2 will use that instead.

### Usage

```
vignette_roclet()
```

### Details

To prevent RStudio from re-building the vignettes again when checking your package, add `--no-build-vignettes` to the "Build Source Package" field in your project options.

### See Also

Other roclets: [namespace\\_roclet](#), [rd\\_roclet](#)



# Index

`double_escape_md`, 3

`export (namespace_roclet)`, 4  
`exportClass (namespace_roclet)`, 4  
`exportMethod (namespace_roclet)`, 4

`import (namespace_roclet)`, 4  
`importClassesFrom (namespace_roclet)`, 4  
`importFrom (namespace_roclet)`, 4  
`importMethodsFrom (namespace_roclet)`, 4  
`is_s3_generic`, 3  
`is_s3_method (is_s3_generic)`, 3

`namespace_roclet`, 4, 5, 8  
`namespace_roclet()`, 6

`object_format`, 4

`rawNamespace (namespace_roclet)`, 4  
`rd_roclet`, 4, 5, 8  
`rd_roclet()`, 6  
`roclet_find`, 5  
`roxygen2 (roxygen2-package)`, 2  
`roxygen2-package`, 2  
`roxygenise (roxygenize)`, 6  
`roxygenize`, 6

`S3method (namespace_roclet)`, 4  
`source_package()`, 6

`tools::buildVignette`, 8

`update_collate`, 7  
`update_collate()`, 6  
`useDynLib (namespace_roclet)`, 4  
`UseMethod()`, 3

`vignette_roclet`, 4, 5, 8