

Package ‘Rd2md’

July 21, 2025

Title Markdown Reference Manuals

Version 1.0.1

Description Native R only allows PDF exports of reference manuals.
The 'Rd2md' package converts the package documentation files into
markdown files and combines them into a markdown version of the package
reference manual.

Depends R (>= 3.6)

Imports tools

Suggests knitr, testthat, rmarkdown, devtools

License GPL

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.2.3

URL <https://github.com/quantsch/rd2md>

BugReports <https://github.com/quantsch/rd2md/issues>

NeedsCompilation no

Author Julian Busch [aut, cre]

Maintainer Julian Busch <jb@quants.ch>

Repository CRAN

Date/Publication 2024-06-17 20:00:02 UTC

Contents

Rd2md-package	2
as_markdown	2
md_document	2
output_format	3
rd_str_to_md	3
read_rdfile	4
render_refman	4

Index	6
--------------	----------

Rd2md-package	<i>Reference Manual in Markdown</i>
---------------	-------------------------------------

Description

Functions to parse and convert Rd files to markdown format reference manual.

Author(s)

Julian Busch jb@quants.ch

See Also

knitr

as_markdown	<i>Parse into Markdown</i>
-------------	----------------------------

Description

S3 Methods to parse different classes into Markdown.

Usage

```
as_markdown(x, ...)
```

Arguments

x	Object
...	Further arguments passed to methods

md_document	<i>Output Format Markdown</i>
-------------	-------------------------------

Description

Specifies markdown output format parameters.

Usage

```
md_document(
  file_ext = ".md",
  section_level = 1,
  date_format = "%F",
  toc = FALSE
)
```

Arguments

file_ext	File extension, if default output_file name
section_level	Initial section level, subsection level n will consequently be section_level + n
date_format	See format argument of <code>base::strptime()</code>
toc	Currently not implemented. If table of contents should be included in reference manual.

output_format	<i>Output Format Definition</i>
---------------	---------------------------------

Description

Specifies output format parameters.

Usage

```
output_format(output_class_suffix, file_ext, parse_fun, ...)
```

Arguments

output_class_suffix	Suffix to class name. E.g. md for output_format_md.
file_ext	File extension, if default output_file name.
parse_fun	Parsing function, should be a generic S3 method.
...	Named arguments inserted into list structure output and eventually passed as arguments to parse_fun.

rd_str_to_md	<i>Translate an Rd string to markdown</i>
--------------	---

Description

Note that this will always end in one newline \n.

Usage

```
rd_str_to_md(x, fragment = TRUE, ...)
```

Arguments

x	Rd string. Backslashes must be double-escaped ("\\").
fragment	logical indicating whether this represents a complete Rd file
...	additional arguments for as_markdown

Examples

```
rd_str_to_md("a\n%b\nc")

rd_str_to_md("a & b")

rd_str_to_md("\\strong{\\emph{x}}")

rd_str_to_md("\\enumerate{\\item test 1\n\\item test 2}")
rd_str_to_md("wrapped \\itemize{\\item test 1\n\\item test 2} in text")
```

read_rdfile	<i>Read .Rd file</i>
-------------	----------------------

Description

And return object of class rdfile

Usage

```
read_rdfile(path, subclass = NULL)
```

Arguments

path	File path of *.Rd-file
subclass	Potential subclass of returned rdfile object

render_refman	<i>Create Markdown Reference Manual</i>
---------------	---

Description

This is a wrapper to combine the Rd files of a package source or binary into a reference manual in markdown format.

Usage

```
render_refman(pkg = getwd(), output_file = NULL, output_format = md_document())
```

Arguments

pkg	Full path to package directory. Default value is the current working directory. Alternatively, a package name can be passed as string. If this is the case, base::find.package() is applied.
output_file	Specify outputfile, including file extension
output_format	Output format function. See output_format() . Default is md_document() , but one could provide a custom output format.

References

Murdoch, D. (2010). [Parsing Rd files](#)

See Also

- Package [Rd2markdown](#), on which the original version was based on.
- Package [pkgdown](#), on which the current (refactored) version is based on.

Examples

```
## give source directory of your package
pkg_dir = "~/git/MyPackage"

## create reference manual
## render_refman(pkg_dir)
```

Index

- * **Rd**
 - Rd2md-package, [2](#)
- * **help**
 - Rd2md-package, [2](#)
- * **manual**
 - Rd2md-package, [2](#)
- * **markdown**
 - Rd2md-package, [2](#)
- * **reference**
 - Rd2md-package, [2](#)
- as_markdown, [2](#)
- base::find.package(), [4](#)
- base::strptime(), [3](#)
- md_document, [2](#)
- md_document(), [4](#)
- output_format, [3](#)
- output_format(), [4](#)
- Rd2md (Rd2md-package), [2](#)
- Rd2md-package, [2](#)
- rd_str_to_md, [3](#)
- read_rdf, [4](#)
- render_refman, [4](#)