

Package: redoc (via r-universe)

May 26, 2026

Version 0.1.0.9000

Title Reversible Reproducible Documents

Description Implements a reversible 'R Markdown' to 'Microsoft Word' pipeline.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

ByteCompile true

URL <https://github.com/noamross/redoc>

BugReports <https://github.com/noamross/redoc/issues>

RoxygenNote 7.3.2

Imports rmarkdown (>= 1.12), knitr, officer (>= 0.3.2), diffobj (>= 0.2.3), stringi, mime, xml2, whoami, yaml, jsonlite, sessioninfo

Suggests spelling, testthat (>= 3.0.0), roxygen2, covr, pkgdown, rstudioapi

SystemRequirements pandoc (>= 2.1.2) - <http://pandoc.org>

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Language en-US

Config/testthat/edition 3

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev

Repository <https://redoc-dev.r-universe.dev>

Date/Publication 2025-09-28 16:25:02 UTC

RemoteUrl <https://github.com/redoc-dev/redoc>

RemoteRef HEAD

RemoteSha 1206310e05b3e5ab8a9843689765a773fb0d8743

Contents

redoc-package	2
dedoc	2
is_redoc	4
make_preknitter	4
make_wrapper	5
pandoc_ast	6
redoc	7
redoc_diff	8
redoc_example_rmd	9
redoc_extract_rmd	10
roundtrip_active_file	11
Index	12

redoc-package	<i>Reversible Reproducible Documents</i>
---------------	--

Description

Implements a reversible 'R Markdown' to 'Microsoft Word' pipeline.

Support

The development repository for the **redoc** package is found at <https://github.com/noamross/redoc>. Please file bug reports or other feedback at <https://github.com/noamross/redoc/issues>

Author(s)

Noam Ross <noam.ross@gmail.com>

dedoc	<i>Convert an Reversible Document back to R Markdown</i>
-------	--

Description

Converts a document originally created with `redoc()` back to R Markdown, including changes made to text in MS Word.

Usage

```
dedoc(
  docx,
  to = NULL,
  dir = ".",
  track_changes = "comments_only",
  block_missing = "comment",
  inline_missing = "omit",
  wrap = getOption("redoc.wrap", 80),
  overwrite = FALSE,
  orig_docx = NULL,
  orig_codefile = NULL,
  verbose = FALSE
)
```

Arguments

docx	The ‘.docx’ file to convert
to	the filename to write the resulting .Rmd file. The default is to use the same basename as the docx document
dir	The directory to write the ‘.Rmd’ to. Defaults to current working directory
track_changes	How to deal with tracked changes and comments in the .docx file. "accept" accepts all changes, and "reject" rejects all of them. The default, "criticmarkup", converts the tracked changes to Critic Markup syntax . "comments_only" will only convert comments, as other changes can be viewed with <code>redoc_diff()</code> . "all" marks up tracked changes and comments in <code></code> tags and is useful for debugging. See the pandoc manual for details.
block_missing, inline_missing	What to do about code blocks or inline code whose output has been removed in the editing of the Word document. "restore" attempts to restore the code as close to its original location in the document as possible. "comment" will do so but wrap it in HTML comments. "omit" will not restore the code at all.
wrap	The width at which to wrap text. If NA, text is not wrapped. Set the default with "redoc.wrap" in <code>options()</code> .
overwrite	Whether to overwrite an existing file
orig_codefile, orig_docx	The original <code>.codelist.yml</code> or Word document created when the document was first knit. Useful for debugging, or in cases where the word file has been corrupted or transformed, for instance, by copy-and-pasting the content into a new file. If provided, dedoc will use this codefile or word file to re-create the .Rmd file with the text of the input.
verbose	whether to print pandoc progress text

Details

R chunks may be lost in the editing process if using non-Microsoft word processors (e.g. LibreOffice or in copy-and-pasting text into a new document).

is_redoc	<i>Is this a reversible document?</i>
----------	---------------------------------------

Description

A function for testing if the file can be de-rendered. If not, un-knitting may be attempted with the `orig_chunkfile` or `orig_docx` files in `dedoc()`.

Usage

```
is_redoc(docx)
```

Arguments

`docx` A path to a `.docx` file or an `rdocx` object produced by `officer::read_docx()`

Value

a logical value

Examples

```
is_redoc(redoc_example_docx())
```

make_preknitter	<i>Generate a function to run prior to knitting</i>
-----------------	---

Description

redoc modifies R Markdown documents prior to knitting to capture and store document elements that will need to be restored later. It does this via a function passed to the `pre_knit` argument of `rmarkdown::output_format()`. `make_preknitter` generates this function from a list of [wrapper functions](#).

Usage

```
make_preknitter(wrappers = list())
```

Arguments

`wrappers` a list of [wrapper functions](#).

Details

rmarkdown does not provide a mechanism to modify the file prior to knitting, so **redoc** accomplishes this by reaching up the call stack and modifying the environment in the `rmarkdown::render()` function. The function generated by `make_preknitter` will do this by generating the pre-processed R Markdown file and associated list of code chunks, adding these the list of intermediates for cleanup, and switching the document input to this pre-processed document.

Value

A function to be used in `'output_format(pre_knit = ...)'`

Examples

```
make_preknitter(wrappers = list(htmlcommentwrap, latexwrap))
```

make_wrapper	<i>Make a wrapper function</i>
--------------	--------------------------------

Description

Wrapper functions are passed to the `wrapper=` argument of `redoc()` to specify what text in R Markdown files should be captured and restored when de-rendering (in addition to code chunks, inline code, and YAML blocks). `make_wrapper()` simplifies creation of these wrapper functions and returns a function that can be passed to `redoc()`.

Wrappers included in the package, all of which are used by default, are:

Type of markup	Function	Label	Type
HTML Comments	<code>htmlcommentwrap()</code>	"htmlcomment"	inline
Single-line LaTeX	<code>latexwrap()</code>	"latex"	block
Pandoc citations	<code>citationwrap()</code>	"citation"	inline
Raw Blocks	<code>rawblockwrap()</code>	"rawblock"	block
Raw Spans	<code>rawspanwrap()</code>	"rawspan"	inline

Usage

```
make_wrapper(label, regex, type = c("block", "inline"))
```

Arguments

label	the label to use for chunks of this type, e.g., "citation", "table". These should be unique for each type of wrapper used. <code>redoc()</code> will throw an error otherwise.
regex	A regular expression that identifies the text to be wrapped and restored, including all delimiters. It will search in a single string with line breaks. ICU regular expressions are used. The <code>(?s)</code> flag is recommended for multi-line expressions.
type	whether the text should be treated as inline or block text. Inlines are wrapped in span elements and blocks are wrapped in divs. These are treated differently in several ways, especially when restoring text in <code>dedoc()</code> .

Details

Some captured text can not be selected by regular expressions, in which case custom functions can be provided to parse out the relevant text. See the vignette "Developing with redoc" for more detail.

Value

A function of class `redoc_wrapper`

Examples

```
rmarkdown::render(
  redoc_example_rmd(),
  output_format = redoc(
    wrappers = list(
      htmlcommentwrap,
      latexwrap)))

# This is how each of these functions are defined in the redoc package
htmlcommentwrap <- make_wrapper(
  label = "htmlcomment",
  regex = "(?s)<!--.*?-->",
  type = "inline")

latexwrap <- make_wrapper(
  label = "latex",
  regex = "(?<=[\\n\\r])\\\\\\\\\\\\w+.*?(?=[\\n\\r])",
  type = "block")

rawblockwrap <- make_wrapper(
  label = "rawblock",
  regex = "(?s)```\\{=\\w+\\}\\..*?```\\h*",
  type = "block")

rawspanwrap <- make_wrapper(
  label = "rawspan",
  regex = "(?s)`[^`]*`\\{=\\w+?\\}\\}",
  type = "inline")

citationwrap <- make_wrapper(
  label = "citation",
  regex = "(?:@\\w+|\\[\\.?!-@\\w+.*?\\](?!\\[\\(\\{))",
  type = "inline")
```

Description

This is a convenience function for testing and development.

Usage

```
pandoc_ast(file, from = NULL, tolist = TRUE)
```

Arguments

file	the file to convert using pandoc.
from	the format to convert from. If NULL (default) File type will be auto-detected by extension. .Rmd files will be treated as .md.
tolist	whether to return the AST as an R list. If FALSE, will return length-1 character vector of raw JSON.

Value

A list containing the structured document

Examples

```
ast <- pandoc_ast(redoc_example_docx())
```

redoc

R Markdown format for Reversible Reproducible Word Documents

Description

Format for converting from R Markdown to a Microsoft Word Document that can be reversed using [dedoc\(\)](#) after editing in Word.

Usage

```
redoc(  
  highlight_outputs = FALSE,  
  wrap = getOption("redoc.wrap", 80),  
  margins = NULL,  
  line_numbers = NULL,  
  comment_author = NULL,  
  keep_md = FALSE,  
  wrappers = list(htmlcommentwrap, latexwrap, rawblockwrap, rawspanwrap, cmwrap,  
                  citationwrap),  
  diagnostics = TRUE,  
  ...  
)
```

Arguments

highlight_outputs	whether to highlight outputs from chunks and inline code in the final document
wrap	when round-tripping the document, at what width to wrap the markdown output? Set the default with "redoc.wrap" in options(). See dedoc() .
margins	page margin size. Can be a single value or a named vector with values, top, bottom, left, right, gutter, header, and footer. If NULL defaults to the reference document.
line_numbers	either TRUE or list with any of the arguments start, by, restart, and distance
comment_author	The name to attribute any CriticMarkup tracked changes to. Defaults to whoami::fullname() .
keep_md	whether to keep the intermediate markdown document
wrappers	a list of wrapper functions to capture text to protect when rendering and de-rendering. See make_wrapper() .
diagnostics	Whether to embed diagnostic information in the output file. If TRUE, this will save session information and the current pandoc and (if used) RStudio versions inside the Word document for later bug-checking.
...	other parameters passed to rmarkdown::word_document()

redoc_diff	<i>Compare changes in an edited redoc Word Document against the original</i>
------------	--

Description

redoc_diff() produces a diff object comparing the current contents of a Word document originally generated by [redoc\(\)](#) to the original R markdown file used to create it.

Usage

```
redoc_diff(
  docx,
  target = "original",
  current = "current",
  track_changes = "comments_only",
  block_missing = "comment",
  inline_missing = "omit",
  wrap = getOption("redoc.wrap", 80),
  mode = "sidebyside",
  context = "auto",
  tar.banner = NULL,
  cur.banner = NULL,
  ...
)
```

Arguments

docx	Path to an MS Word .docx file originally generated with <code>redoc()</code> and since edited.
target, current	Which versions of the document to compare. One of "original", "roundtrip", or "current".
track_changes, block_missing, inline_missing	Arguments passed to <code>dedoc()</code> to determine how to handle edits in the Word document.
wrap	Width to wrap text lines when converting from docx to markdown. If NULL, no wrapping. Set the default with "redoc.wrap" in <code>options()</code> .
mode, context, tar.banner, cur.banner, ...	Arguments passed to <code>diffobj::diffFile()</code> to customize printing of the diff.

Details

When an .docx file is created with `redoc()`, it internally stores the original R Markdown file as well as a version that is round-tripped to .docx and back. `redoc_diff()` de-renders the current .docx to R Markdown (with `dedoc()`) and compares against these versions.

Value

A `Diff` object, which will be displayed in the RStudio Viewer, a browser, or the console, depending on the context.

redoc_example_rmd *Files for examples and testing*

Description

Files for examples and testing

Usage

```
redoc_example_rmd()

redoc_example_docx()

redoc_example_edited_docx()
```

Examples

```
redoc_example_rmd()
redoc_example_docx()
redoc_example_edited_docx()
```

redoc_extract_rmd *Extract the Rmd used to to produce a Reversible Word Doc*

Description

Documents produced with `redoc()` store an copy of the original `.Rmd` files used to produce them. This is useful for diffing against the version created with `dedoc()`, especially if tracked changes have not been used.

Usage

```
redoc_extract_rmd(  
  docx,  
  type = c("original", "roundtrip"),  
  dir = ".",  
  to = NULL,  
  overwrite = FALSE  
)
```

Arguments

<code>docx</code>	A path to a word file or a an rdocx object created with <code>officer::read_docx()</code> .
<code>type</code>	One of "original" or "roundtrip". "original" extracts the exact document originally knit. "roundtrip" (default) extracts a document that has been converted to Word and back with no edits in between. The latter should be more useful for comparing against edits, as line-wrapping and placement of no-output chunks should match.
<code>dir</code>	The directory to write the <code>‘.Rmd‘</code> to. Defaults to current working directory
<code>to</code>	the filename to write the resulting <code>.Rmd</code> file. The default is to use the the original name with either <code>.original.Rmd</code> or <code>roundtrip.Rmd</code> extensions.
<code>overwrite</code>	whether to overwrite existing files

Value

The path to the extracted `.Rmd`

Examples

```
redoc_extract_rmd(redoc_example_docx(), dir = tempdir())
```

roundtrip_active_file *RStudio Addin Functions*

Description

These functions act on the RStudio text editor and can be used via the Addins menu or the R console. `roundtrip_active_file()` ("Render and Update") knits the current file using `redoc()` replaces the text with text that has been round-tripped to Word and back. This helps eliminate small formatting differences.

The `dedoc_*` functions de-render Word documents (using `dedoc()`), and place the results in the current or new file. By default, they will also display a diff (generated via `redoc_diff()`) of edits in the word file compared to its original version a compilation time. If called without an input file, they will prompt for file selection via the RStudio GUI.

Usage

```
roundtrip_active_file()
```

```
dedoc_to_active_file(docx = NULL, showdiff = TRUE)
```

```
dedoc_to_new_file(docx = NULL, showdiff = TRUE)
```

Arguments

<code>docx</code>	The input Word file, originally generated by <code>redoc()</code>
<code>showdiff</code>	Display a diff of the current version of the document against the rendered version?

Index

- * **package**
 - redoc-package, 2
- addins (roundtrip_active_file), 11
- citationwrap (make_wrapper), 5
- dedoc, 2
- dedoc(), 4, 5, 7–11
- dedoc_to_active_file
 - (roundtrip_active_file), 11
- dedoc_to_new_file
 - (roundtrip_active_file), 11
- Diff, 9
- diffobj::diffFile(), 9

- htmlcommentwrap (make_wrapper), 5

- ICU regular expressions, 5
- is_redoc, 4

- latexwrap (make_wrapper), 5

- make_preknitter, 4
- make_wrapper, 5
- make_wrapper(), 8

- officer::read_docx(), 4, 10

- pandoc_ast, 6

- rawblockwrap (make_wrapper), 5
- rawspanwrap (make_wrapper), 5
- redoc, 7
- redoc(), 2, 5, 8, 10, 11
- redoc-package, 2
- redoc_addins (roundtrip_active_file), 11
- redoc_diff, 8
- redoc_diff(), 3, 11
- redoc_example_docx (redoc_example_rmd),
9
- redoc_example_edited_docx
 - (redoc_example_rmd), 9
- redoc_example_rmd, 9
- redoc_examples (redoc_example_rmd), 9
- redoc_extract_rmd, 10
- rmarkdown::output_format(), 4
- rmarkdown::render(), 5
- rmarkdown::word_document(), 8
- roundtrip_active_file, 11

- whoami::fullname(), 8
- wrapper functions, 4
- wrappers (make_wrapper), 5