

Package ‘RmdConcord’

May 7, 2026

Type Package

Title Concordances for 'R Markdown'

Version 0.3

Description Supports concordances in 'R Markdown' documents. This currently allows the original source location in the '.Rmd' file of errors detected by 'HTML tidy' to be found more easily, and potentially allows forward and reverse search in 'HTML' and 'LaTeX' documents produced from 'R Markdown'. The 'LaTeX' support has been included in the most recent development version of the 'patchDVI' package.

License GPL-2

Encoding UTF-8

URL <https://github.com/dmurdoch/RmdConcord>,
<https://dmurdoch.github.io/RmdConcord/>

BugReports <https://github.com/dmurdoch/RmdConcord/issues>

Imports rmarkdown, tools, knitr (>= 1.42)

Suggests markdown

VignetteBuilder knitr, rmarkdown

SystemRequirements pandoc (>=2.11.3 with 'commonmark_x' processing and 'sourcepos' extension, needed for 'rmarkdown' formats).

NeedsCompilation no

Author Duncan Murdoch [aut, cre],
Heather Turner [ctb]

Maintainer Duncan Murdoch <murdoch.duncan@gmail.com>

Repository CRAN

Date/Publication 2023-12-10 17:20:06 UTC

Contents

commonmark_document	2
html_with_concordance	3
processConcordance	4
processLatexConcordance	5
test_packages	6
tidy_validate	7

Index	8
--------------	----------

commonmark_document	<i>R Markdown drivers to add concordance</i>
---------------------	--

Description

These drivers replace the like-named **rmarkdown** or **markdown** drivers with ones that output Commonmark rather than Pandoc Markdown. Commonmark is a dialect of Markdown. The Pandoc driver for Commonmark supports output of source position information. By using this function as your output driver, you can get that in your own documents.

A replacement for `markdown::latex_format` is not planned; see the note below.

Usage

```
html_documentC(sourcepos = TRUE, ...)
html_vignetteC(sourcepos = TRUE, ...)
pdf_documentC0(latex_engine = "pdflatex",
               sourcepos = TRUE,
               defineSconcordance = TRUE,
               ...)
html_formatC(options = list(sourcepos = TRUE), ...)
```

Arguments

<code>latex_engine</code>	Command to convert ‘.tex’ file to ‘.pdf’.
<code>defineSconcordance</code>	If TRUE, insert a definition of the <code>\Sconcordance</code> macro just before <code>\begin{document}</code> .
<code>sourcepos</code>	Whether to include source position information.
<code>options</code>	The options argument to pass to the base driver. If <code>options\$sourcepos</code> is not specified, it will default to TRUE.
<code>...</code>	Other arguments to pass to the base driver.

Details

Each driver modifies the standard driver from **rmarkdown** or **markdown**, e.g. `html_documentC` is similar to [html_document](#), but uses Commonmark and adds concordances.

Value

An R Markdown output format object which will add concordance information.

Note

The `pdf_documentC0` function adds the concordances, but they won't be interpreted by LaTeX or PDF previewers. To get that to happen, use `patchDVI:::pdf_documentC`.

The `html_formatC` function requires **markdown** version 1.12.1 or higher. If a lower version of that package is installed `html_formatC` will still run, but will issue a warning and not add any concordances.

The concordances produced by `html_formatC` tend to be off by a few lines as the underlying Commonmark processor only issues source position records once per paragraph.

A `latex_formatC` driver appears as if it would be quite messy and is not currently planned. The issues are:

- **commonmark** doesn't support source position attributes in LaTeX output
- It doesn't allow edits between the parsing and rendering steps. This is what `pdf_documentC` does.
- A possible strategy would be to render first to XML (which does keep source position attributes), then convert the XML to LaTeX with macros inserted to record source positions. However, `markdown:::latex_format` produces the final LaTeX document in several steps, and would have to support this two-stage rendering on some but not all of the steps.

Author(s)

Duncan Murdoch

html_with_concordance *Convert an R Markdown driver to one that handles concordances.*

Description

These functions produce a new driver which matches the old one in most respects, but adds an argument `sourcepos` (and possibly others). If that argument is `TRUE` (the default) then concordances are handled by the new driver.

These functions are used to produce `html_documentC` and similar drivers in this package, but should also work on other drivers that produce HTML output using Pandoc.

Usage

```
html_with_concordance(driver)
pdf_with_concordance(driver)
```

Arguments

`driver` An R Markdown driver that produces HTML, LaTeX or PDF using Pandoc.

Value

A new driver function.

Examples

```
html_with_concordance(rmarkdown::html_fragment)
pdf_with_concordance(rmarkdown::latex_fragment)
```

processConcordance *Get the concordance from the "datapos" values and record it in a file.*

Description

Pandoc can record concordance information in datapos attributes when converting Commonmark documents to HTML. This retrieves that information, and rewrites it as standard R concordance data.

Usage

```
processConcordance(filename, newfilename = filename,
                   rename = NULL,
                   followConcordance = TRUE)
```

Arguments

filename	The filename of the HTML file produced by Pandoc.
newfilename	A filename in which to write the changed data.
rename	A named character vector. Names are the names in the datapos attributes; values are the names that should be included in the concordance instead. This might be used since knitr produces a Markdown file and renames it later.
followConcordance	If filename already contains concordance data, assume that the Rmd file was produced automatically, and chain the concordances.

Value

Called for the side effect of rewriting the concordance, it returns newfilename invisibly.

Author(s)

Duncan Murdoch

Examples

```

# This example works on the file inst/sample/Sample.Rmd,
# which should be a copy of the vignette Sample.Rmd. This
# is convenient because RStudio doesn't install vignettes by default.

# First, see the results without concordances:

library(RmdConcord)
dir <- tempdir()
intermediates <- tempfile()

infile <- system.file("sample/Sample.Rmd", package = "RmdConcord")
outfile1 <- file.path(dir, "html_vignette.html")

rmarkdown::render(infile,
                  intermediates_dir = intermediates,
                  output_file = outfile1,
                  quiet = TRUE)
tidy_validate(outfile1)

# Next, see them with concordances by setting
# the output format to use RmdConcord::html_documentC
# which post-processes the document with processConcordance.

dir <- tempdir()
outfile2 <- file.path(dir, "commonmark.html")
rmarkdown::render(infile,
                  intermediates_dir = intermediates,
                  output_file = outfile2,
                  output_format = html_documentC(),
                  quiet = TRUE)
tidy_validate(outfile2)
unlink(c(intermediates, outfile1, outfile2), recursive = TRUE)

```

```
processLatexConcordance
```

Process a LaTeX concordance file.

Description

Pandoc can record concordance information in `\datapos` macros when converting Commonmark documents to LaTeX. This retrieves that information, and rewrites it as standard R concordance data.

Usage

```
processLatexConcordance(filename,
                       newfilename = filename,
                       rename = NULL,
```

```
followConcordance = NULL,
defineSconcordance = TRUE)
```

Arguments

filename	The filename of the LaTeX file produced by Pandoc.
newfilename	A filename in which to write the changed data.
rename	A named character vector. Names are the names in the datapos attributes; values are the names that should be included in the concordance instead. This might be used since knitr produces a Markdown file and renames it later.
followConcordance	If filename already contains concordance data, assume that the Rmd file was produced automatically, and chain the concordances.
defineSconcordance	Whether to insert the definition of the <code>\Sconcordance</code> macro.

Value

Called for the side effect of rewriting the concordance, it returns `newfilename` invisibly.

Author(s)

Duncan Murdoch

test_packages	<i>Test for sufficient versions of supporting packages</i>
---------------	--

Description

This function tests for the presence of packages that will support **RmdConcord** functions.

Usage

```
test_packages(error = TRUE, pandoc = TRUE)
```

Arguments

error	If TRUE, missing requirements will result in an error. If FALSE, they result in a warning.
pandoc	Is Pandoc needed?

Value

A logical value indicating that the requirements are present. Will never return FALSE if error is TRUE, it will just trigger an error.

Examples

```
test_packages()
```

tidy_validate	<i>Reproduce tidy validation done in R 4.3.0.</i>
---------------	---

Description

This function is taken from R-devel, to test the use of **RmdConcord** in checking packages.

Usage

```
tidy_validate(f, tidy = "tidy")
```

Arguments

f	An 'HTML' file to test.
tidy	The name of the HTML Tidy executable.

Value

An object showing issues in the file and locations in the source file.

Index

`commonmark_document`, [2](#)

`html_document`, [2](#)
`html_documentC`, [3](#)
`html_documentC (commonmark_document)`, [2](#)
`html_formatC (commonmark_document)`, [2](#)
`html_vignetteC (commonmark_document)`, [2](#)
`html_with_concordance`, [3](#)

`latex_formatC (commonmark_document)`, [2](#)

`pdf_documentC0 (commonmark_document)`, [2](#)
`pdf_with_concordance`
 (`html_with_concordance`), [3](#)

`processConcordance`, [4](#)
`processLatexConcordance`, [5](#)

`test_packages`, [6](#)
`tidy_validate`, [7](#)