Package 'pharmaRTF'

September 22, 2026

Type Package

Title Enhanced RTF Wrapper for Use with Existing Table Packages

Version 0.1.5

Description Enhanced RTF wrapper written in R for use with existing R tables packages such as 'Huxtable' or 'GT'. This package fills a gap where tables in certain packages can be written out to RTF, but cannot add certain metadata or features to the document that are required/expected in a report for a regulatory submission, such as multiple levels of titles and footnotes, making the document landscape, and controlling properties such as margins.

Depends R (>= 3.5.0)

License MIT + file LICENSE

BugReports https://github.com/atorus-research/pharmaRTF/issues

Encoding UTF-8

Imports assertthat (>= 0.2.1), stringr (>= 1.4.0), purrr (>= 0.3.3), huxtable (>= 4.7.1)

Suggests testthat (>= 2.1.0), dplyr (>= 0.8.4), readr (>= 1.3.1), gt (>= 0.2.0), magrittr (>= 1.5), knitr (>= 1.28), rmarkdown (>= 2.1), readxl (>= 1.3.1), kableExtra (>= 1.1.0), plyr (>= 1.8.5), tidyverse (>= 1.3.0)

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Eli Miller [aut] (ORCID: <https://orcid.org/0000-0002-2127-9456>), Ashley Tarasiewicz [aut], Michael Stackhouse [aut, cre] (ORCID: <https://orcid.org/0000-0001-6030-723X>), Atorus Research LLC [cph]

Maintainer Michael Stackhouse <mike.stackhouse@atorusresearch.com>

Repository CRAN

Date/Publication 2026-09-22 16:30:02 UTC

Contents

pharmaRTF-package . . . 2
add_titles . . . 3
align . . . 4
bold . . . 5
column_header_buffer . . . 6
font . . . 7
font_size . . . 8
header_height . . . 9
header_rows . . . 10
hf_line . . . 11
ignore_cell_padding . . . 13
index . . . 15
italic . . . 16
margins . . . 17
orientation . . . 18
pagesize . . . 19
rtf_doc . . . 20
text . . . 22
titles_and_footnotes_from_df . . . 23
view_footnotes . . . 24
view_titles . . . 25
write_rtf . . . 25

Index **27**

`pharmaRTF-package` *pharmaRTF*

Description

Enhanced RTF wrapper written in R for use with existing R tables packages such as huxtable or GT. This package fills a gap where tables in certain packages can be written out to RTF, but cannot add certain metadata or features to the document that are required/expected in a report for a regulatory submission, such as multiple levels of titles and footnotes, making the document landscape, and controlling properties such as margins.

This package intends to provide a flexible and reliable framework to connect R to a pharmaceutical reporting workflow.

Future Plans

- colors
- graphs

Author(s)

Maintainer: Michael Stackhouse <mike.stackhouse@atorusresearch.com> (ORCID)

Authors:

- Eli Miller <eli.miller@atorusresearch.com> (ORCID)
- Ashley Tarasiewicz <Ashley.Tarasiewicz@atorusresearch.com>

Other contributors:

- Atorus Research LLC [copyright holder]

See Also

Useful links:

- Report bugs at https://github.com/atorus-research/pharmaRTF/issues

---

`add_titles` *Add* `hf_line` *title(s) and footnote(s) to a* `rtf_doc` *object*

Description

Add `hf_line` titles/footnote objects to a `rtf_doc` object

Usage

```
add_titles(doc, ..., replace = FALSE)

add_footnotes(doc, ..., replace = FALSE)
```

Arguments

`doc`	`rtf_doc` on which hf_line object(s) (i.e. titles/footnotes) will be attached
`...`	A vector of `hf_line` objects to add passed to `add_hf()`
`replace`	If FALSE, lines will be appended/ordered with current header/footer lines. If TRUE, lines will replace the existing content.

Value

`hf_line` object(s) (i.e. titles/footnotes) to be added

Examples

```
# Adding titles after rtf_doc construction
ht <- huxtable::huxtable(
 column1 = 1:5,
 column2 = letters[1:5]
)
rtf <- rtf_doc(ht)

rtf <- add_titles(rtf, hf_line("The Title"))

# Adding footnotes after rtf_doc construction
ht <- huxtable::huxtable(
 column1 = 1:5,
 column2 = letters[1:5]
)
rtf <- rtf_doc(ht)

rtf <- add_footnotes(rtf, hf_line("The Footnote"))
```

`align` *Return or set text alignment*

Description

These property functions modify or return the text alignment in a `hf_line` object. Supported options are: 'left', 'right', 'center', and 'split'.

An alignment option of 'split' requires that two text elements are provided in the titles/footnotes `hf_line` object. The first text element will be aligned on the left and the second on the right.

Usage

```
align(x, ...)

align(x) <- value

set_align(x, value)
```

Arguments

`x`	`hf_line` object
`...`	Additional arguments passed to method dispatch.
`value`	A string representing the alignment.

Value

For `align()`, the alignment of the supplied `hf_line` object. For `set_align()` and `` `align<-`() ``, the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

pharmaRTF::align(rtf$titles[[1]])
# Returns "center"

pharmaRTF::align(rtf$titles[[1]]) <- "left"
# Sets alignment to 'left'

pharmaRTF::text(rtf$titles[[1]]) <- c("Left Text", "Right Text")
pharmaRTF::align(rtf$titles[[1]]) <- "split"
# When rtf is printed, "Left Text" will be left aligned, and "Right Text"
# will be right aligned. Both will appear on the same line in the document.
```

`bold` *Return or set bold*

Description

These property functions modify or return the bold attribute of a `hf_line` object.

Usage

```
bold(x, ...)

bold(x) <- value

set_bold(x, value)
```

Arguments

`x`	A `hf_line` object
`...`	Additonal arguments passed to method dispatch
`value`	A logical vector to set the value of the bold attribute

Value

For `bold()`, the bold attribute of the supplied `hf_line` object. For `` `bold<-`() `` and `set_bold()`, the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

pharmaRTF::bold(rtf$titles[[1]])
# Returns FALSE

pharmaRTF::bold(rtf$titles[[1]]) <- TRUE
# Sets bold to TRUE
```

`column_header_buffer` *Return or set column_header_buffer attributes*

Description

These property functions modify and return the column header buffers of a `rtf_doc` object. These are stored as a named vector. Names should be `top` and `bottom`. This attribute adds rows to the top or bottom of the table column headers to pad it from the titles above or the table below.

Usage

```
column_header_buffer(x, ...)

column_header_buffer(x) <- value

set_column_header_buffer(x, ...)
```

Arguments

`x`	A `rtf_doc` object
`...`	Additonal arguments passed to method dispatch. Should include argument top and bottom with numeric elements.
`value`	A named vector detailing the top and bottom buffer.

Value

For `column_header_buffer`, the column_header_buffer attribute of the supplied `rtf_doc`. For `` `column_header_buffer<-`() `` and `set_column_header_buffer()`, the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

column_header_buffer(rtf)
# Returns c(top = 0, bottom = 0), the default

column_header_buffer(rtf) <- c(bottom = 1)
# Sets the bottom column_header_buffer to 1
```

---

font *Return or set font*

Description

These property functions modify or return the fonts of an `rtf_doc` object or individual titles/footnotes objects of the `hf_line` class.

When used on an `rtf_doc` object to retrieve fonts, the distinct set of fonts of all objects contained within the rtf_doc are returned. When used on an `rtf_doc` to set fonts, the default font for the RTF document is set.

When used on titles/footnotes (`hf_line` objects), the font is either returned of set for that individual line.

Usage

```
font(x, ...)

font(x) <- value

set_font(x, value)
```

Arguments

`x`	`rtf_doc` object, the table of a `rtf_doc` object, or a `hf_line` object
`...`	Additional arguments passed to method dispatch
`value`	A string representing a font

Value

For `font()`, the font attribute of the object in the case of `hf_line` and the table, or each unique font in the table, titles, footnotes, and the overall document in the case of `rtf_doc`. For `set_font()` and `` `font<-`() ``, the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("text", font = "Comic Sans")))

pharmaRTF::font(rtf)
# Returns "Courier New" "Comic Sans"

pharmaRTF::font(rtf) <- "Times"

pharmaRTF::font(rtf)
# Returns "Times" "Comic Sans"

pharmaRTF::font(rtf$titles[[1]]) <- "Windings"

pharmaRTF::font(rtf)
# Returns "Times" "Windings"
```

`font_size` *Return or set font size*

Description

These property functions modify or return the font sizes of an `rtf_doc` object or individual titles/footnotes objects of the `hf_line` class.

When used on an rtf_doc object to retrieve font sizes, the document level default font size within the `rtf_doc` is returned. When used on an `rtf_doc` to set fonts, the default font size for the RTF document is set.

When used on titles/footnotes (`hf_line` objects), the font size is either returned of set for that individual line.

Usage

```
font_size(x, ...)

font_size(x) <- value

set_font_size(x, value)
```

Arguments

`x`	`rtf_doc` object or a `hf_line` object.
`...`	Additonal arguments passed to method dispatch.
`value`	A numeric value for font size in points.

Value

For `font_size()`, the font_size attribute of the supplied `rtf_doc` or `hf_line`. For `` `font_size<-`() `` and `set_font_size`, the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle", font_size = 15)))

pharmaRTF::font_size(rtf)
# Returns 12

pharmaRTF::font_size(rtf) <- 14

pharmaRTF::font_size(rtf$titles[[1]])
# Returns 15
pharmaRTF::font_size(rtf)
# Returns 14
```

`header_height` *Return or set header/footer height*

Description

These property functions modify or return the header_height/footer_height attribute of a `rtf_doc` object. The header/footer height is the default amount of space allocated to the header/footer from the margin. If the content of the header/footer exceeds this amount of space, it will be expanded.

Usage

```
header_height(x, ...)

header_height(x) <- value

set_header_height(x, value)

footer_height(x, ...)

footer_height(x) <- value

set_footer_height(x, value)
```

Arguments

x	A rtf_doc object
...	Additonal arguments passed to method dispatch
value	A numeric value to set the header_height/footer_height

Value

For header_height()/footer_height(), the header_height/footer_height attribute of thesupplied rtf_doc object. For `header_height<-()`/`footer_height<-`() and set_header_height()/ set_footer_height(), the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

header_height(rtf)
# Returns 0.5, the default

footer_height(rtf) <- 1
# Sets footer_height to 1
```

---

header_rows *Return or set the header_rows*

Description

These property functions modify or return the header_rows attribute of a rtf_doc object. Only required and valid when the rtf_doc table object is a huxtable.

The header rows control the number of rows taken from a huxtable table into the header of the document as the column header. When pulled into the headers, these rows are repeated on each page. Can be set to 0 to disable repeating column headers.

Usage

```
header_rows(x, ...)

header_rows(x) <- value

set_header_rows(x, value)
```

Arguments

x	A rtf_doc object
...	Additonal arguments passed to method dispatch
value	A numeric value to change the header_rows attribute.

Value

For header_rows(), the header_rows attribute of the rtf_doc object. For `header_rows<-`() and set_header_rows(), the modified object.

Examples

```
ht <- huxtable::huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

header_rows(rtf)
# This is a wrapper for header_rows(x$table)
header_rows(rtf$table)
# Both of these return 1, the default

header_rows(rtf$table) <- 0
# Sets header_rows to 0
```

---

hf_line *Create a title line container*

Description

hf_line objects represent individual title or footnote lines and their associated metadata. These objects are passed to an rtf_doc for display in the header or footer of an RTF document.

A character vector of length <= 2 describes the text to display. Using a single text element, the text can be aligned left, right, or center. Using two text elements, the alignment can be set to "split", which will left align the first element, and right align the second. If alignment is set to anything else, the text elements will be pasted together.

Lines can either be passed to the titles/footnotes arguments in the call to rtf_doc or added later with the add_titles or add_footnotes functions. Supported properties are detailed in the arguments section.

Usage

```
hf_line(
  ...,
  align = c("center", "left", "right", "split"),
  bold = FALSE,
  italic = FALSE,
  font = NA,
  font_size = NA,
  index = NA
)
```

Arguments

...	A character list/vector. If `length(...)` is 2 and `align` is not 'split', values are pasted together.
align	Text alignment in document. Options are 'center', 'left', 'right', and 'split'. A 'split' alignment will left align the string in the first text item and right align the second. Defaults to center.
bold	`TRUE` or `FALSE`. Defaults to FALSE.
italic	`TRUE` or `FALSE`. Defaults to FALSE.
font	A string to specify the font display. Ensure the intended RTF reader can display the selected font. Fonts for all fields will default to the default font of the `rtf_doc` object, which unless otherwise assigned, is Courier New.
font_size	Font size in points. Font sizes for all fields will default to the default font size of the `rtf_doc` object, which unless otherwise assigned, is 12
index	Position to display header or footnote lines in the RTF document. Orders in ascending order with NAs last. Defaults to NA.

Value

An object of class `hf_line` with the properties described in the Arguments section.

Supported Formatting

Several special display formats are supported to display document data. When the `rtf_doc` is written, the package will determine if the text of an `hf_line` object starts with a keyword. Regular expression matching and replacement is used for formatting.

- PAGE_FORMAT: - Can take up to two replacements to format current page(first), and total number of pages(second). Page numbers are replaced in the string using %s For example, for a format of Page 1 of 5, use PAGE_FORMAT: Page %s of %s. For a format of just 5, use PAGE_FORMAT: %s.
- DATE_FORMAT: - Describes the date/time the document was generated. Formats are specified using standard R date formatting tokens. Details on formatting dates can be found here.
- FILE_PATH: - Describes the file path the R session was executed from. The location of the executing file will be populated over the token replacement string "%s". Formats can be

specified like "FILE_PATH: Executed from: %s" or simply "FILE_PATH: %s". Note that the location of the executing file in R may not be intuitive. There are multiple ways to determine the containing R file based on how it was executed.

- When the file is executed using `Rscript`, this field will populated as the executed Rscript file.
- When the file is sourced, this field will populate with the location of the sourced file.
- When a file is run interactively (i.e. from the R console), this field will populate as <run interactively>.

Examples

```
# Adding lines during rtf_doc construction
ht <- huxtable::huxtable(
 column1 = 1:5,
 column2 = letters[1:5]
)
titles_l <- list(
  hf_line(c("The Title Left", "The Titles Right"), align = "split"),
  hf_line("A Bold, italic Title", bold = TRUE, italic = TRUE,
    align = "left", font_size = 20, font = "Times New Roman")
)
rtf <- rtf_doc(ht, titles = titles_l)

# Adding lines after rtf_doc construction
rtf <- add_footnotes(rtf,
    hf_line("PAGE_FORMAT: Page %s of %s"),
    hf_line("DATE_FORMAT: %H:%M %A, %B %d, %Y"),
    hf_line("FILE_PATH: Source: %s")
  )
```

---

ignore_cell_padding *Return or set ignore_cell_padding*

Description

These property functions modify and return the ignore_cell_padding attribute of a `rtf_doc` object. By default, the huxtable package will pad rows of a table. This attribute will remove those default settings – which allow the cells to have a smaller amount of padding than setting the cell padding to 0. See the Details section for a more thorough description of the implementation.

Usage

```
ignore_cell_padding(x, ...)

ignore_cell_padding(x) <- value

set_ignore_cell_padding(x, value)
```

Arguments

x	A rtf_doc object
...	Additonal arguments passed to method dispatch
value	A logical value to set the attribute

Details

Cell padding in RTF code has multiple command words associated with it. Huxtable uses the command word \clpad<t,b,l,r>N to control the cell padding. This command word is additionally controlled by the command word \clpadf<t,b,l,r>N. There are two possible values for N in \clpadf<t,b,l,r>N:

- 0: Null. This ignores \clpad<t,b,l,r> in favor of \trgaph (Word 97 style cell padding).
- 3: Twips

The ignore_cell_padding function toggles the \clpadf<t,b,l,r>N command words in the RTF document to 0 instead of 3. By using Word 97 style cell padding, the minimum amount of space is closer than using the 0 twips setting when the RTF is rendered inside Word. This effectively closes the gap between rows, which may be a desirable appearance in some outputs.

Value

For ignore_cell_padding(), the ignore_cell_padding attribute of the supplied rtf_doc object. For `ignore_cell_padding<-`() and set_ignore_cell_padding(), the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

ignore_cell_padding(rtf)
# Returns FALSE, the default

ignore_cell_padding(rtf) <- TRUE
# Sets ignore_cell_padding to TRUE
```

index *Return or set index*

Description

These property functions modify or return the index of a hf_line object. The index sets the order in which a title will appear. Indicies can be any numeric value as long as they are not duplicated.

Usage

```
index(x, ...)

index(x) <- value

set_index(x, value)
```

Arguments

x	A hf_line object
...	Additonal arguments passed to method dispatch
value	Numeric value to order index

Value

For index(), the index attribute of the supplied hf_line object. For `index<-()` and set_index(), the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

index(rtf$titles[[1]])
# Returns NULL

index(rtf$titles[[1]]) <- 2
# Sets index of first titles to 2
```

`italic` *Return or set italics*

Description

These property functions modify or return the italics attribute of a `hf_line` object. The italic attribute takes on a logical value of TRUE or FALSE, where TRUE italicizes the text of the line.

Usage

```
italic(x, ...)

italic(x) <- value

set_italic(x, value)
```

Arguments

`x`	A `hf_line` object
`...`	Additonal arguments passed to method dispatch
`value`	A logical vector to set the value of the bold attribute

Value

For `italic()`, the italic attribute of the supplied `hf_line`. Fot `` `italic<-()` `` and `set_italic()` the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

pharmaRTF::italic(rtf$titles[[1]])
# Returns FALSE

pharmaRTF::italic(rtf$titles[[1]]) <- TRUE
# Sets italic to TRUE
```

`margins` *Return or set margins*

Description

These property functions return or set the margin attribute of a `rtf_doc` object. These are stored as a named vector. Names should be `top`, `bottom`, `left`, and `right`. Margins are measured in inches.

Usage

```
margins(x, ...)

margins(x) <- value

set_margins(x, value)
```

Arguments

`x`	A `rtf_doc` object
`...`	Additonal arguments passed to method dispatch
`value`	A named list or vector detailing the page margins

Value

For `margin()`, a named vector of the margin attribute of the supplied `rtf_doc`. For `` `margin<-()` `` and `set_margin()` the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

margins(rtf)
# Returns c(top = 1, bottom = 1, left = 1, right = 1)

margins(rtf) <- c(top = 2)
# Sets top margin to 2
```

`orientation` *Return or set orientation*

Description

These property functions modify or return the orientation attribute of a `rtf_doc` object. Options are landscape or portrait. See the `rtf_doc` help page for important notes about this property.

Usage

```
orientation(x, ...)

orientation(x) <- value

set_orientation(x, value)
```

Arguments

`x`	A `rtf_doc` object
`...`	Additonal arguments passed to method dispatch
`value`	A character vector of either 'landscape' or 'portrait'

Value

For `orientation()`, the orientation attribute of the supplied `rtf_doc` object. For `` `orientation<-()` `` and `set_orientation()` the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

orientation(rtf)
# Returns landscape

orientation(rtf) <- "portrait"
# Sets orientation to portrait
```

pagesize *Return or set pagesize*

Description

These property functions modify or return the pagesize attribute of a rtf_doc object. Stored as a named vector with height and width names.

Usage

```
pagesize(x, ...)

pagesize(x) <- value

set_pagesize(x, value)
```

Arguments

x	A rtf_doc object
...	Additonal arguments passed to method dispatch
value	A named numeric vector with the names height and width.

Details

Note that when the orientation of the document is switched to 'portrait', the height and width will reverse when the RTF document is being written - but the attribute values will not change. This is because the default rtf_doc orientation is 'landscape', and switching the attributes of the object allows for a possibility of inadvertently overriding the functionality of the orientation attribute.

Value

For pagesize(), the pagesize attribute of the rtf_doc object. For `pagesize<-`() and set_pagesize(), the modified object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

pagesize(rtf)
# Returns c(height = 8.5, width = 11.0)

pagesize(rtf) <- c(height = 12)
# Sets height of page to 12 inches
```

```
orientation(rtf) <- 'portrait'
pagesize(rtf)
#  width height
#    8.5   12.0
# Note: Despite changing orientation, attributes don't change
```

`rtf_doc` *Create a* `rtf_doc` *object*

Description

This constructs the main object that will be used for an RTF document. The object is composed of a table, titles(s), and footnote(s).

A table contained in the `rtf_doc` object should be a supported class. The huxtable package is the most supported, however our intention is to support other packages capable of writing RTF tables as well. Currently, it is planned to support the `gt` package, but the `gt` package's RTF methods are not functional.

The titles and footnotes are composed of `hf_line` objects.

See the vignette for a more complete view of intended usage.

Usage

```
rtf_doc(table, titles = list(), footnotes = list(), header_rows = 1)
```

Arguments

`table`	A table of a supported class.
`titles`	A list of `hf_line` objects containing table titles and associated formatting.
`footnotes`	A list of `hf_line` objects containing table footnotes and associated formatting.
`header_rows`	An integer determining how many rows of the table are column headers. Only used for huxtable tables. Can be set to 0 to disable repeating column headers.

Value

A list with a table, titles, and footnotes component. Class of "rtf_doc" with the properties described below.

`rtf_doc` Properties

Document level properties set the defaults and will be used where they are not overridden by `hf_line` or table properties.

- font - A string representing the font to display when it is not specified by the table or `hf_line`. Defaults to Courier New.

- font_size - A numeric value representing the size of the font in points. Defaults to 12.
- margins - Inches of margins in the document as a named vector. Names are `top`, `bottom`, `left`, and `right`. Defaults to 1 for all.
- orientation - Orientation of the document. Defaults to 'landscape'. When 'portrait', the height and width are switched while writing the document to effectively rotate the document 90 degrees. For example, if width is 11" and height is 8.5", while writing the document will have a height of 11" and a width of 8.5". Additionally, when 'landscape', a keyword is written to the RTF to indicate that the document is landscape. .
- header_height - Height of the header where the titles and column headers are displayed. Defaults to .5 inches.
- footer_height - Height of the footer where the footnotes are displayed. Defaults to .5 inches.
- pagesize - Size of the page in inches. Defaults to 8.5(height) by 11(width). These defaults align with the default orientation of 'landscape'. When the orientation is switched to 'portrait', the height and width will switch while the RTF document is being generated, but the document attributes themselves will not change.
- header_rows - Huxtable table only. Number of rows that are defined as the header that will be repeated across pages. Defaults to 1. Can be set to 0 to disable repeating column headers.
- ignore_cell_padding - Huxtable table only. Flag to ignore cell padding padding that is added during RTF encoding. Minimizes the amount of space between rows. Defaults to FALSE.
- column_header_buffer - This attribute adds rows to the top or bottom of the table column headers to pad it from the titles above or the table below. Defaults to 0 and 0.

See Also

hf_line

Examples

```
# Adding lines during rtf_doc construction
ht <- huxtable::huxtable(
 column1 = 1:5,
 column2 = letters[1:5]
)
# Set table properties
library(magrittr) #load in a pipe
ht %>%
  huxtable::set_bold(1, 1:ncol(ht), TRUE) %>%
  huxtable::set_escape_contents(TRUE) %>%
  huxtable::set_col_width(c(0.25, 0.75))

rtf <- rtf_doc(ht, titles = list(hf_line("My Header")))
# Set document properties
rtf <- rtf %>%
  set_font_size(15) %>%
  set_ignore_cell_padding(TRUE)

names(rtf)
```

```
write_rtf(rtf, file = tempfile())
```

`text` *Return or set text*

Description

These property functions modify or return the fonts of a `rtf_doc` object. `text()` will always return a vector of length 2. If the text is only of length one an empty string will be appended.

Usage

```
text(x, ...)

text(x) <- value

set_text(x, value)
```

Arguments

`x`	A `hf_line` object
`...`	Additonal arguments passed to method dispatch
`value`	A character vector of length 0, 1, or 2 to set the text value of a `hf_line` object.

Value

For `text()`, the text vector of the supplied `hf_line` object. For `` `text<-()` `` and `set_text()`, the modfied object.

Examples

```
library(huxtable)
ht <- huxtable(
  column1 = 1:5,
  column2 = letters[1:5]
)
rtf <- rtf_doc(ht, list(hf_line("aTitle")))

text(rtf$titles[[1]])
# Returns "aTitle" ""

text(rtf$titles[[1]]) <- "aDifferentTitle"
# Sets titles to "aDifferentTitle"
```

titles_and_footnotes_from_df *Read titles and footnotes from a dataframe*

Description

Reads a data frame with header/footnote information and attaches it to an `rtf_doc` object.The most effective way to use this function is to pass information to a custom reader for your process. See Details section for more information.

Usage

```
titles_and_footnotes_from_df(
  doc,
  from.df = NULL,
  from.file = NULL,
  reader = NULL,
  ...
)
```

Arguments

`doc`	A `rtf_doc` object to append header and footnote information.
`from.df`	A `data.frame` object with title and footnote information.
`from.file`	A file path to a file with title and footnote information.
`reader`	A function to read the data from the from.file argument.
`...`	Parameters passed to `read_hf` where they are processed and constructed into `hf_line` objects.

Details

Titles_and_footnotes_from_df allows you to attach titles and footnotes (as hf_line objects) from a data.frame. This data.frame could be a data.frame in your local environment, or read in from an external file. The best way to utilize this method is to create a custom reader function. This custom reader function is a function that you develop to:

- Read a source file into a data.frame
- Preprocess as necessary to keep only necessary records and variables
- Ensure that variables are the correct data type

Titles_and_footnotes_from_df allows you to pass arguments into the reader function, which gives you the capability to keep titles and footnotes for all of your outputs in a central file and pass a filtering option, or any additional parameters as necessary. For an example implementation, see our vignette.

Value

A rtf_doc object with header/footnote information attached.

Required Columns

The following columns are required fields in a data.frame passed to titles_and_footnotes_from_df:

- type(character - 'title' or 'footnote')
- text1(character)
- text2(character)
- align(character - left, right, center, or split)
- bold(logical)
- italic(logical)
- font(character)
- index(numeric)

---

view_footnotes *View footnote information*

Description

View footnotes attached to an rtf_doc as a data.frame.

Usage

```
view_footnotes(doc)
```

Arguments

doc	rtf_doc object

Value

data.frame of the footnote information

view_titles *View title information*

Description

View titles attached to an rtf_doc as a data.frame.

Usage

```
view_titles(doc)
```

Arguments

doc	rtf_doc object

Value

data.frame of the title information

write_rtf *Write RTF document*

Description

Writes the RTF document to a specified file.

Usage

```
write_rtf(doc, file = NULL)
```

Arguments

doc	The RTF document to be written.
file	A character string naming a file open for writing.

Value

File is written to the file provided by sinking the console output. No output is returned to the R environment.

Examples

```
## Create and write RTF document
ht <- huxtable::huxtable(
 column1 = 1:5,
 column2 = letters[1:5]
)
rtf <- rtf_doc(ht)

write_rtf(rtf, file=tempfile()) #writes a table with no header/footnotes to 'test.rtf'
```

Index

add_footnotes *(add_titles)*, 3
add_titles, 3
align, 4
align<- *(align)*, 4

bold, 5
bold<- *(bold)*, 5

column_header_buffer, 6
column_header_buffer<-
(column_header_buffer), 6

font, 7
font<- *(font)*, 7
font_size, 8
font_size<- *(font_size)*, 8
footer_height *(header_height)*, 9
footer_height<- *(header_height)*, 9

header_height, 9
header_height<- *(header_height)*, 9
header_rows, 10
header_rows<- *(header_rows)*, 10
hf_line, 11, *21*

ignore_cell_padding, 13
ignore_cell_padding<-
(ignore_cell_padding), 13
index, 15
index<- *(index)*, 15
italic, 16
italic<- *(italic)*, 16

margins, 17
margins<- *(margins)*, 17

orientation, 18
orientation<- *(orientation)*, 18

pagesize, 19
pagesize<- *(pagesize)*, 19
pharmaRTF *(pharmaRTF-package)*, 2
pharmaRTF-package, 2

rtf_doc, *18*, 20

set_align *(align)*, 4
set_bold *(bold)*, 5
set_column_header_buffer
(column_header_buffer), 6
set_font *(font)*, 7
set_font_size *(font_size)*, 8
set_footer_height *(header_height)*, 9
set_header_height *(header_height)*, 9
set_header_rows *(header_rows)*, 10
set_ignore_cell_padding
(ignore_cell_padding), 13
set_index *(index)*, 15
set_italic *(italic)*, 16
set_margins *(margins)*, 17
set_orientation *(orientation)*, 18
set_pagesize *(pagesize)*, 19
set_text *(text)*, 22

text, 22
text<- *(text)*, 22
titles_and_footnotes_from_df, 23

view_footnotes, 24
view_titles, 25

write_rtf, 25