Package: derrick (via r-universe)

October 14, 2026

Type Package

Title Export 'gtsummary' Tables with 'reporter'

Version 0.1.1

Date 2026-06-26

Maintainer Xiecheng Gu <guxc0405med@163.com>

Description Convert 'gtsummary' tables and plain data frames into clinical-style Rich Text Format, plain text, Word, PDF, and HTML outputs using 'reporter', with managed widths, pagination, indentation, spanning headers, and optional intermediate data exports.

License Apache License 2.0

URL https://github.com/openpharma/derrick/

BugReports https://github.com/openpharma/derrick/issues

Depends R (>= 4.4)

Imports dplyr, purrr, reporter (>= 1.4.7), rlang, stringi, tibble

Suggests cards (>= 0.8.0), gtsummary (>= 2.5.1), knitr, rmarkdown, testthat (>= 3.3.0)

Config/Needs/website pkgdown

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libicu-dev libjpeg-dev

Repository https://openpharma.r-universe.dev

Date/Publication 2026-07-16 11:03:54 UTC

RemoteUrl https://github.com/openpharma/derrick

RemoteRef HEAD

RemoteSha f86e5f353c115f3b9f68b01b7d86c7c0330cc4b5

Contents

derrick-package 2
gtsummary_reporter 2

Index **7**

`derrick-package` *Export gtsummary tables with reporter*

Description

Convert `gtsummary` tables and plain data frames into clinical-style `reporter` RTF, TXT, DOCX, PDF, and HTML outputs.

Author(s)

Maintainer: Xiecheng Gu <guxc0405med@163.com>

Authors:

- Joe Zhu <sha.joe.zhu@gmail.com>
- Bill Huang <Bill.Huang@Toastmasters.org.tw>

See Also

Useful links:

- https://github.com/openpharma/derrick/
- Report bugs at https://github.com/openpharma/derrick/issues

---

`gtsummary_reporter` *Export* `gtsummary` *tables to clinical-style* `reporter` *outputs*

Description

Convert a `gtsummary` object (or a plain `data.frame`) into a `reporter` table and write RTF, TXT, DOCX, PDF, and/or HTML outputs. The function is designed for clinical reporting workflows and supports column labels, spanning headers, pagination, indentation handling, and optional export of intermediate data.

Usage

```
gtsummary_reporter(
  gts_obj,
  file_path,
  max_table_width = NULL,
  min_col_width = 0.6,
  column_widths = NULL,
  column_labels = NULL,
  spanning_headers = NULL,
  report_orientation = "landscape",
  report_paper_size = "letter",
  report_units = "inches",
  report_margins = NULL,
  report_font_size = 9,
  indent_unit = 1,
  output_types = c("RTF", "TXT"),
  save_rds = TRUE,
  save_ard = FALSE,
  rds_dir = "rds",
  rows_per_page = NULL,
  max_chars_per_line = NULL,
  debug_indent = FALSE,
  debug_spanning = FALSE,
  group_columns = NULL,
  group_blank_after = TRUE,
  page_by_columns = NULL,
  page_by_label = ""
)
```

Arguments

gts_obj	A gtsummary object (with table_body/table_styling) or a plain data.frame to export.
file_path	Required output path. The extension is ignored and output files are written according to output_types.
max_table_width	Optional maximum total table width in report_units. If NULL, reporter chooses the automatic table width. Non-NULL values are capped at page_width - left_margin - right_margin. With the defaults (report_paper_size = "letter", report_orientation = "landscape", report_units = "inches", and default 1-inch left/right margins), the maximum effective value is 9. For common defaults in inches: letter landscape = 9, letter portrait = 6.5, A4 landscape = 9.69, A4 portrait = 6.27, RD4 landscape = 8.70, RD4 portrait = 5.70, legal landscape = 12, legal portrait = 6.5. The same defaults in centimeters are: letter landscape = 22.86, letter portrait = 16.51, A4 landscape = 24.62, A4 portrait = 15.92, RD4 landscape = 22.22, RD4 portrait = 14.52, legal landscape = 30.48, legal portrait = 16.51. report_paper_size = "none" gives an infinite page width.

If margins exceed the physical page width, the effective width is `0`. Use `NULL` unless you intentionally want to constrain the table.

`min_col_width` Minimum width allowed for manually supplied `column_widths`, in `report_units`.

`column_widths` Optional manual column widths in `report_units`, either a numeric vector or a `"|"`-delimited string. Values are applied in display column order, usually the `label` column first, followed by statistic or data columns. For example, on the default 9-inch usable width, a 4-column table could use `"3|2|2|2"`; a 5-column table could use `"3|1.5|1.5|1.5|1.5"`. The practical total range is from `n_cols * min_col_width` up to the effective `max_table_width`; if the supplied total is larger, widths are scaled down to fit while respecting `min_col_width` where possible. If `n_cols * min_col_width` is wider than the effective page width, the per-column floor is relaxed to `effective_width / n_cols`. If fewer widths than columns are supplied, the last width is repeated; if more are supplied, extras are ignored. Use `NULL` unless you need precise control over column allocation. When `NULL`, reporter calculates column widths using its output-specific layout metrics.

`column_labels` Optional column header overrides, as a named vector/list or a data frame with `column` and `label`.

`spanning_headers`
Optional spanning header definitions, as a data frame or list with fields `from`, `to`, and `label`.

`report_orientation`
Page orientation (`"landscape"` or `"portrait"`).

`report_paper_size`
Paper size (`"letter"`, `"legal"`, `"A4"`, `"RD4"`, `"none"`, or numeric length-2 vector).

`report_units` Units for dimensions (`"inches"` or `"cm"`).

`report_margins` Optional margins as named vector/list (`top`, `right`, `bottom`, `left`) or numeric length-4 vector.

`report_font_size`
Base report font size.

`indent_unit` Number of spaces per indent level in `label`.

`output_types` Output types to write; supported values are `"RTF"`, `"TXT"`, `"DOCX"`, `"PDF"`, and `"HTML"`.

`save_rds` Logical; when `TRUE`, save processed output data as an `.rds` file.

`save_ard` Logical; when `TRUE`, also save ARD from a `gtsummary` object. Default is `FALSE` because CTR templates usually save ARD before calling `gtsummary_reporter()`.

`rds_dir` Reserved argument for backward compatibility.

`rows_per_page` Optional maximum number of table rows per manual chunk. If `NULL`, no manual pre-splitting is done; reporter handles pagination in `write_report()` using its output-specific layout algorithm.

`max_chars_per_line`
Optional integer. For TXT output, constrains the total table width so that at most this many characters fit across one TXT line (Courier at 12 CPI). Applied after

max_table_width, so the stricter limit wins for TXT. Other output formats are not constrained by this argument. Title and footnote wrapping is delegated to reporter.

debug_indent — Logical; print indentation diagnostics.

debug_spanning — Logical; print spanning-header diagnostics.

group_columns — Optional grouping columns to hide and apply blank_after.

group_blank_after — Logical; whether to apply blank_after based on grouping columns.

page_by_columns — Optional columns to hide and pass to reporter::page_by() as per-category subtitles. Only the first column is used because reporter supports one page-by variable per table.

page_by_label — Prefix label for page_by_columns; use "" to display only the group value.

Details

Environment variables are automatically consumed when available: title1-title9, footnote1-footnote9, and progname.

Widths are resolved in this order:

1. Compute the usable page width as page_width - left_margin - right_margin.
2. If max_table_width is supplied, cap it at that usable width and pass it to reporter::create_table(). If it is NULL, no table width is passed and reporter calculates the automatic width for each output format.
3. For TXT output only, apply max_chars_per_line as an additional character-budget cap (max_chars_per_line / 12 inches), when supplied.
4. If column_widths is supplied, pass those manual widths to reporter after scaling them down to the effective table width when needed. If column_widths is NULL, no column widths are passed and reporter calculates them from the target format, font, font size, headers, and table contents.

Pagination is delegated to reporter by default. When rows_per_page is NULL, the table is passed to reporter::write_report() as one table, and reporter computes page breaks using output-specific fixed metrics plus the actual wrapped title, header, row, and footnote line counts. Set rows_per_page only when you need to force manual chunks before reporter's own pagination runs.

Value

A character vector containing generated output file paths.

Examples

```
gts_tbl <- gtsummary::trial |>
  dplyr::select(trt, age, grade) |>
  gtsummary::tbl_summary(by = trt) |>
  gtsummary::add_p()
```

```
out <- gtsummary_reporter(
  gts_obj = gts_tbl,
  file_path = tempfile("clinical_report_", fileext = ".rtf"),
  output_types = "TXT",
  save_rds = FALSE
)

out
```

Index

derrick *(*derrick-package*)*, 2
derrick-package, 2

gtsummary_reporter, 2