Package: knitroff (via r-universe)

October 17, 2026

Type Package

Title Dynamic Reports in 'troff' with 'knitr'

Version 0.1.1

Description Write reports in 'troff' with the 'ms' macros and R code chunks between '.SS' and '.SE', the way 'eqn' and 'tbl' blocks sit in a document, and turn them into PDF, PostScript or a terminal draft with GNU 'groff'. 'knitr' runs the code, and its results can also be placed in the text.

License MIT + file LICENSE

URL https://github.com/danielgccr/knitroff

BugReports https://github.com/danielgccr/knitroff/issues

Encoding UTF-8

Imports knitr

Suggests rstudioapi

SystemRequirements GNU groff (>= 1.22) for render_roff(); pdfinfo (poppler) for figures in PDF output

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

Date/Publication 2026-10-16 14:27:01 UTC

RemoteUrl https://github.com/danielgccr/knitroff

RemoteRef HEAD

RemoteSha beb56dfd61b31383b887b4df9798d0026e042c3f

Contents

knitroff-package 2
eqn 2
groff_available 3
hooks_roff 4
knit_roff 5
ms 6

render_roff 10
roff_table 11
tbl 12
troff 14

Index **17**

`knitroff-package` *Dynamic Reports in troff with knitr*

Description

Write a report in troff with the ms macros and put R code in it between `.SS` and `.SE`. `knit_roff` runs the code; `render_roff` typesets the result with GNU groff as a PDF, a PostScript file, or a draft for the terminal. The design follows troff's preprocessors, eqn and tbl, which each found their own blocks in a document and passed the rest through: here, one more preprocessor at the front of the pipeline, which runs R.

In RStudio, the Addins menu has *Insert troff chunk* and *Render troff document*, which stand in for Insert Chunk and the Knit button in `.Rms` files. Give them keys in Tools > Modify Keyboard Shortcuts. They need the **rstudioapi** package.

See Also

The reference for writing documents: `ms` (the ms macros, such as `.PP`), `troff` (requests, escapes, units and special characters), `eqn` (equations) and `tbl` (tables).

The example report: `system.file("examples", "states.Rms", package = "knitroff")`.

`eqn` *eqn: equations in knitroff documents*

Description

eqn, by Brian Kernighan and Lorinda Cherry, is the troff preprocessor for mathematics; its input reads much as the formula is spoken: `x sup 2 over 2`. `render_roff` runs it for PDF and PostScript output. This page follows eqn(1) for groff 1.24.1.

Displays and inline equations

`.EQ` **...** `.EN` a displayed equation, centred by ms; `.EQ L` left-aligns it, `.EQ I` indents it, and a second argument is a label at the right margin, as in `.EQ C (1)`.

`delim $$` inside an `.EQ` ... `.EN`, makes `$` delimit equations within the text: `the mean $x bar$ is`. `delim off` stops it.

knitroff in the `"utf8"` draft, displayed equations appear as their eqn source, because groff cannot set them on a terminal; inline equations appear between their delimiters.

Writing equations

eqn reads words separated by spaces; spaces themselves do not show. Braces group: `e sup {i pi}`. Quoted text is set as it is, in roman: `"for all"`. ~ is a space and ^ half a space. eqn sets letters in italic and digits, operators and function names such as `sin` in roman, and turns `+ - = <= >=` into their mathematical characters.

`x sub i`, `x sup 2`	subscript, superscript; `x sub i sup 2` both
`a over b`	a fraction
`sqrt x`	square root
`sum from i=1 to n x sub i`	limits under and over `sum`, `prod`, `int`, `lim`, `union`, `inter`
`left ( a over b right )`	brackets that grow to fit: ([{ | and their mates, and `left floo`
`x bar`, `x under`, `x hat`, `x dot`, `x vec`	marks over or under x; also `dotdot`, `tilde`, `dyad`
`pile { a above b above c }`	a vertical stack; `lpile`, `cpile`, `rpile` align it
`matrix { ccol { a above b } ccol { c above d } }`	a matrix of columns; `lcol`, `rcol` align them
`mark`, `lineup`	align successive equations at a point, such as their = signs
`roman`, `italic`, `bold`, `fat`	the face of the next item; `fat` overstrikes to embolden
`size 12 x`, `size +2 x`	type size of the next item
`up n`, `down n`, `fwd n`, `back n`	move by n hundredths of an em
`define name "text"`	a macro: `define var "x sub i"`; `ndefine` and `tdefine` only
`gsize n`, `gfont f`	the global type size and italic font of equations

Names eqn knows

Greek letters: `alpha beta gamma delta epsilon zeta eta theta iota kappa lambda mu nu xi omicron pi rho sigma` spelled with a capital (`Gamma`) or in capitals (`GAMMA`) for uppercase.

Symbols and functions: `inf partial grad del prime approx times nothing ... half`, `== != += -> <- << >>`, and `sin cos tan sinh cosh tanh arc exp log ln lim max min det Re Im` and `for if`.

Added by GNU eqn: `cdot cdots ldots utilde dollar`, and the primitives `accent uaccent big smallover vcenter spl`

See Also

`ms`, `troff`, `tbl`. In a terminal: `man eqn`; and B. W. Kernighan and L. L. Cherry, *A System for Typesetting Mathematics*, Communications of the ACM 18 (1975), and their *Typesetting Mathematics, User's Guide*.

Examples

```
eq <- c(".EQ", "s sup 2 ~=~ 1 over {n - 1} sum from i=1 to n ( x sub i - x bar ) sup 2", ".EN")
cat(eq, sep = "\n")
```

---

groff_available *Can groff typeset a document here?*

Description

Runs groff on an empty ms document, with the arguments `render_roff` uses and without output, and reports whether it succeeded. Finding the groff program is not enough: Debian's `groff-base`, installed for man pages, has no ms macros and no PDF driver.

Usage

```
groff_available(format = c("pdf", "ps", "utf8"), groff = "groff")
```

Arguments

format	"pdf", "ps" or "utf8", as in render_roff.
groff	The groff program.

Value

TRUE or FALSE.

See Also

render_roff, which stops with an explanation when this is FALSE.

Examples

```
groff_available("utf8")
```

---

hooks_roff *Set knitr's output hooks to write troff*

Description

Installs the hooks knit_roff uses, for those who drive knit themselves. Set knitr::opts_knit$set(roff.format = "pdf") to choose how figures are included.

Usage

```
hooks_roff()
```

Value

The hooks that were replaced, invisibly, for knitr::knit_hooks$restore().

knit_roff *Knit an R troff document*

Description

Runs the R code of a troff document and writes plain troff, ready for GNU groff with the `-ms` macros. Code chunks sit between `.SS` and `.SE`, the way eqn blocks sit between `.EQ` and `.EN`; chunk options follow `.SS`, as in `.SS scatter, fig.cap="Income", echo=FALSE`. Inline results are written `` `r expr` ``, and `.SR name expr` defines the troff string `\*[name]` with the value of `expr`.

Usage

```
knit_roff(input, output = NULL, format = c("pdf", "ps", "utf8"), highlight = TRUE,
          quiet = TRUE)
```

Arguments

input	Path of the document, usually '`.Rms`'.
output	Path of the troff file. By default `input` with extension '`.ms`'.
format	What the troff is for: `"pdf"` and `"ps"` include figures with `.PDFPIC` and `.PSPIC`; `"utf8"`, the terminal draft, leaves a box to paste each figure into and shows each equation as its eqn source.
highlight	Set R keywords in bold and comments in italic, in the constant-width bold and italic fonts, as printed listings did. `FALSE` gives the plain listing of 1981. A chunk can choose for itself with `.SS highlight=FALSE` or `TRUE`.
quiet	Passed to `knit`.

Details

Each chunk becomes one listing in a constant-width font: the code with `>` and `+` prompts, its output under it, kept together on a page. Figures are written to a directory named after the document. knitr's patterns, hooks and options are restored afterwards. No groff is needed to knit.

Value

The path of the troff file, invisibly.

See Also

`render_roff` to typeset the result, `roff_table`.

Examples

```
doc <- file.path(tempdir(), "mean.Rms")
writeLines(c(".PP", "The mean of 1 to 10 is `r mean(1:10)`.",
             ".SS", "x <- 1:10", "summary(x)", ".SE"), doc)
cat(readLines(knit_roff(doc, format = "utf8")), sep = "\n")
```

ms *The ms macros: a reference for knitroff documents*

Description

A knitroff document is troff with the ms macros, a package for papers, reports and memoranda. This page lists the macros, registers and strings of ms as GNU groff implements it, following groff_ms(7) for groff 1.24.1. Marked entries are extensions: *(Berkeley)* from 4.2BSD, *(Tenth Ed.)* from Research Unix, *(GNU)* from groff. Requests and escapes, such as `.sp` and `\fB`, are in `troff`.

Using the ms macros with knitroff

`render_roff()` runs `groff -ms -k -R -e -t -p`: the ms macros, UTF-8 input, and the refer, eqn, tbl and pic preprocessors.

`.SS`, `.SE`, `.SR` are knitroff's, not ms's: knitroff replaces them before groff sees the document. ms has no macros with these names, so there is no clash.

listings each R chunk becomes a `.DS L` ... `.DE` display in the constant-width font `CR`, so it is kept together on one page.

figures a plot becomes `.PDFPIC -C` (PDF) or `.PSPIC -C` (PostScript), centred, with its caption.

a skeleton a document needs at least one paragraph or heading macro; begin the text with `.LP` or `.PP`. After a heading, call a paragraph macro to end the heading text.

settings set registers and strings such as `.nr LL 6i` or `.ds CH` before the first ms macro other than `.RP`.

Document description

Call these in the order shown. `.TL` is required if any of `.RP`, `.AU`, `.AI`, `.AB` is used; `.AE` is required after `.AB`.

`.RP [no-repeat-info] [no-renumber]` report format: put the title, authors and abstract on a separate cover page. `no` is the AT&T form of `no-repeat-info`; the other options are (GNU).

`.TL` the title: the following input lines, until `.AU`, `.AB`, or a heading or paragraph macro.

`.AU` an author's name, on the following lines. Call it again for each author.

`.AI` the preceding author's institution, on the following lines.

`.DA [x ...]` the current date, or x, in the centre footer (and on the cover page with `.RP`).

`.ND [x ...]` no date in the footer; the date, or x, only on the cover page with `.RP`. This is groff's default.

`.AB [no]` begin the abstract, headed by an italic "ABSTRACT"; `no` leaves out the heading.

`.AE` end the abstract.

Paragraphs

Each paragraph macro breaks the line and adds the space in `\n[PD]`.

`.LP` a paragraph with no indentation.

`.PP` a paragraph whose first line is indented by `\n[PI]`.

`.IP [mark [width]]` an indented paragraph, with mark hanging in the margin: the way to make lists (`.IP \(bu`, `.IP 1.`). width (in ens) replaces `\n[PI]` and stays until the next heading or other paragraph macro.

`.QP` a paragraph indented from both margins by `\n[QI]`, for quotations.

`.QS`, `.QE` begin and end a region in which every paragraph is indented from both margins.

`.XP` an "exdented" paragraph: a hanging indent, every line but the first indented. (Berkeley)

Headings

`.NH [depth]` a numbered heading, 1., 1.1., 1.1.1. ..., at depth (default 1); the following lines are its text, ended by a paragraph macro. Skipping a depth draws a warning.

`.NH S n ...` set the heading numbers explicitly; numbering continues from them. (Berkeley)

`.SH [depth]` an unnumbered heading. depth (GNU) matches its size to `.NH` at that depth.

`\*[SN]`, `\*[SN-DOT]`, `\*[SN-NO-DOT]` the number of the last `.NH`, for captions and cross-references; with and without the final period. (GNU, `SN` Berkeley)

`.als SN-STYLE SN-NO-DOT` number headings without the final period. (GNU)

Registers `PSINCR` and `GROWPS` (GNU) make headings above a depth larger; `HORPHANS` (GNU) keeps a heading with lines of the next paragraph.

Typeface

In these macros, text is set in the face, then post in the previous face with no space, and pre before it: `.B word ,` gives a bold word followed by a roman comma. Without arguments, the face holds until the next paragraph, heading or typeface macro. pre is (GNU).

`.B [text [post [pre]]]` bold.

`.I [text [post [pre]]]` italic.

`.R [text [post [pre]]]` roman (upright). Its arguments are (GNU).

`.BI [text [post [pre]]]` bold italic. (Tenth Ed.)

`.CW [text [post [pre]]]` constant width, for code in a sentence. (Tenth Ed.)

`.BX [text]` text in a box (reverse video on a terminal). Use unbreakable spaces, such as `\~`, inside it.

`.UL [text [post]]` underlined text (between underscores on a terminal).

`.LG`, `.SM`, `.NL` type 2 points larger, 2 points smaller, or back to the normal size `\n[PS]`.

`\*{` **...** `\*}` superscript. (GNU)

`\*<` **...** `\*>` subscript. (GNU)

Indented regions

`.RS` begin a region where headings, paragraphs and displays are indented further by `\n[PI]`. Regions nest. Use this, not `.in`, which ms resets.

`.RE` end the innermost region.

Keeps, boxes and displays

A keep holds text on one page; a display also turns filling off, so lines break as in the source.

`.KS`, `.KE` begin and end a keep: if it does not fit, the page breaks before it.

`.KF`, `.KE` a floating keep: if it does not fit, the text after it fills the page and the keep goes to the top of the next.

`.B1`, `.B2` a keep with a box drawn round it. Put it inside `.KF` ... `.KE` to let it float.

`.DS L`, **or** `.LD` a left-aligned display (`.DS`: kept; `.LD`: may break across pages). knitroff's listings are these.

`.DS [I [indent]]`, **or** `.ID [indent]` an indented display, by indent or `\n[DI]`.

`.DS B`, **or** `.BD` a block display: left-aligned, with its longest line centred.

`.DS C`, **or** `.CD` a centred display: each line centred.

`.DS R`, **or** `.RD` a right-aligned display. (GNU)

`.DE` end any display.

Displays and preprocessor regions get `\n[DD]` of space before and after (Berkeley); `\n[DI]` (GNU) is the default indentation.

Tables, pictures, equations and citations

Preprocessors find these tokens only at the start of a line, with nothing between the dot and the name.

`.TS [H]` **...** `.TE` a table for tbl (see `tbl`). With `H`, the rows before `.TH` repeat at the top of each page.

`.PS h v` **...** `.PE` **or** `.PF` a pic diagram; `.PF` ends it and returns to the top of the picture.

`.EQ [align [label]]` **...** `.EN` an equation for eqn (see `eqn`), centred unless align is `L` or `I`, with label at the right margin.

`.[` **...** `.]` a citation for refer: keywords that find one entry in the database named by `database` between `.R1` and `.R2`. With `accumulate` there, `$LIST$` between `.[` and `.]` prints the references, under ms's own heading. The database path is relative to R's working directory.

Footnotes

`\**` an automatically numbered footnote mark in the text. (Berkeley)

`.FS [mark]`, `.FE` begin and end the footnote text, set at the foot of the page. Without mark, the next automatic number is used.

`\n[FF]` sets the style: 0, the mark as a superscript (default); 1, as text with a period; 2, as 1 without indentation; 3, as 1 with the mark hanging. `FI`, `FPS`, `FVS`, `FPD` (the last three GNU) are the footnote indentation, type size, vertical spacing and paragraph distance. `FS-MARK` is a hook macro called by `.FS`. (GNU)

Headers and footers

`.ds LH`, `CH`, `RH` the left, centre and right header; `CH` is `-\n[%]-`, the page number between dashes, by default.

`.ds LF`, `CF`, `RF` the left, centre and right footer, empty by default.

`.OH 'left'centre'right'`, `.EH`, `.OF`, `.EF` headers and footers for odd and even pages. Any character can replace the apostrophes. (Berkeley)

`%` in header or footer text, the page number. ms prints no header on page 1.

`.P1` print the header on page 1 too. (Berkeley)

`PT`, `BT`, `HD` the macros that print the header and footer, which you may redefine; `HD` is called after the header. (`HD` Berkeley)

Columns, tabs and contents

`.1C`, `.2C` one column (default) or two columns.

`.MC [width [gutter]]` as many columns of width as fit, at least gutter apart; without arguments, two columns. `\n[MINGW]` (GNU) is the minimum gutter.

`.TA` reset tab stops to every 5 ens. Set others with `.ta`.

`.XS [page]`, `.XA [page [indent]]`, `.XE` begin, add to, and end a table of contents entry; the lines between are its text. (Berkeley)

`.PX [no]`, `.TC [no]` print the table of contents at the end of the document; `.TC` also restarts page numbers in roman numerals. (Berkeley)

`.XN text`, `.XH depth text` set a heading and make its contents entry at once, after `.NH` or `.SH`. (GNU)

Registers and strings

Set with `.nr name value` or `.ds name text` before the first ms macro. Defaults are for typesetters; terminals use the value in parentheses.

`\n[PO]`	page offset (left margin)	1i (0)
`\n[LL]`	line length	6.5i (65n)
`\n[LT]`	title line length	6.5i (65n)
`\n[HM]`, `\n[FM]`	top and bottom margins	1i
`\n[PS]`	type size	10p
`\n[VS]`	vertical spacing (leading)	12p
`\n[HY]`	hyphenation mode; 0 turns it off (Tenth Ed.)	6
`\*[FAM]`	font family: T Times, H Helvetica, C Courier, ... (GNU)	T
`\n[PI]`	paragraph indentation	5n
`\n[PD]`	space between paragraphs	0.3v (1v)
`\n[QI]`	quotation indentation	5n
`\n[PORPHANS]`	first lines of a paragraph kept together (GNU)	1
`\n[PSINCR]`	heading size increment (GNU)	1p
`\n[GROWPS]`	depth above which headings grow (GNU)	0
`\n[HORPHANS]`	lines kept with a heading (GNU)	1
`\n[FI]`	footnote indentation	2n

\n[FF]	footnote format, 0 to 3	0
\n[FPS], \n[FVS]	footnote type size and spacing (GNU)	PS-2p, FPS+2p
\n[FPD]	footnote paragraph distance (GNU)	PD/2
*[FR]	footnote line length, as a ratio	11/12
\n[DD]	space around displays (Berkeley)	0.5v (1v)
\n[DI]	display indentation (GNU)	0.5i
\n[MINGW]	minimum gutter between columns (GNU)	2n
\n[GS]	1 under groff ms, for a document to test	1

Strings for symbols: *- an em dash; *Q and *U left and right quotation marks. Strings to translate: *[ABSTRACT], *[TOC], *[REFERENCES], and *[MONTH1] to *[MONTH12].

Seventh Edition compatibility

groff ms is a reimplementation of AT&T ms, the package Mike Lesk wrote at Bell Laboratories and shipped with the Seventh Edition (1979). Marked entries are extensions: *(Berkeley)* from 4.2BSD, *(Tenth Ed.)* from Research Unix, *(GNU)* from groff. Leave them out and the document is one 1979 ms could have formatted. Some Seventh Edition macros were specific to Bell Laboratories and are not in groff: the document types EG IM MF MR TM TR, AT CS CT OK SG, the Bell Labs addresses HO IH MH PY WH, and UX. The old accent strings, such as *' (acute) and *: (umlaut), and .AM still work, but UTF-8 input or \[...] characters (see troff) are better.

See Also

troff, eqn, tbl, render_roff. In a terminal: man groff_ms; and M. E. Lesk, *Typing Documents on the UNIX System: Using the -ms Macros with Troff and Nroff*, Bell Laboratories, 1978.

Examples

```
doc <- c(".TL", "A short report", ".AU", "A. Student", ".AB",
         "One paragraph of abstract.", ".AE", ".NH", "Method", ".PP",
         "The mean is `r mean(1:10)`.", ".IP \\(bu", "a list item")
writeLines(doc, file.path(tempdir(), "report.Rms"))
```

---

render_roff — *Knit an R troff document and typeset it with groff*

Description

Knits the document with knit_roff, then runs groff -ms -k -R -e -t -p: the ms macros, UTF-8 input, and the refer, eqn, tbl and pic preprocessors.

Usage

```
render_roff(input, output = NULL, format = c("pdf", "ps", "utf8"), highlight = TRUE,
            groff = "groff")
```

Arguments

input	Path of the document, usually '.Rms'.
output	Path of the result. By default input with extension '.pdf', '.ps' or '.txt'.
format	"pdf", "ps", or "utf8": the draft for a terminal, as nroff gave, to read with less -R.
highlight	Passed to knit_roff: bold keywords and italic comments in the listings, or plain with FALSE.
groff	The groff program.

Details

Needs GNU groff in full; groff_available says whether this machine has it. On Debian and Ubuntu that is the groff package: groff-base, installed for man pages, has no ms macros. PDF output uses groff's gropdf; figures in it need pdfinfo from poppler, and groff runs with -U because .PDFPIC asks pdfinfo for each picture's size. A terminal cannot set equations, so the "utf8" draft shows their eqn source, which was designed to read the way it is spoken.

On Windows, groff comes from MSYS2: in the Rtools45 bash shell, pacman -Sy groff. Check with Sys.which("groff"), or pass the path to groff.exe as groff. pdfinfo, for figures in PDF, is not in the part of MSYS2 that Rtools allows; the package README describes other ways to get it.

Value

The path of the result, invisibly.

Examples

```
if (groff_available("utf8")) {
  doc <- file.path(tempdir(), "states.Rms")
 file.copy(system.file("examples", "states.Rms", package = "knitroff"), doc, overwrite = TRUE)
  draft <- render_roff(doc, format = "utf8")
  cat(head(readLines(draft), 40), sep = "\n")
}
```

---

roff_table *A data frame as a tbl table*

Description

Returns troff for the tbl preprocessor, a centred, boxed table with a bold header and numeric columns aligned, to be set when the document goes through groff -t (which render_roff does). Make it the value of a chunk.

Usage

```
roff_table(x, caption = NULL, digits = getOption("digits"))
```

Arguments

`x`	A data frame or matrix.
`caption`	Optional caption, set in bold, centred above the table.
`digits`	Significant digits for numeric columns.

Value

Troff text marked to go into the document as is.

Examples

```
cat(roff_table(head(mtcars[, 1:3]), caption = "Three columns of mtcars"))
```

---

tbl *tbl: tables in knitroff documents*

Description

tbl, by Mike Lesk, is the troff preprocessor for tables. `render_roff` runs it for every output, the terminal draft included, and `roff_table` writes its input from a data frame. This page follows tbl(1) for groff 1.24.1.

A table

A table sits between `.TS` and `.TE` and has three parts: options ending in `;`, one format line per row ending in `.`, and the data, one line per row with entries separated by tabs, or here by `@`, which survives copying from this page.

```
.TS
center box tab(@);
cB cB
l n.
State@Income
_
Alaska@6315
Maine@3694
.TE
```

The last format line applies to all remaining rows. `roff_table` writes tables like this from a data frame.

Options

`center`	centre the table (otherwise it is at the left margin)
`expand`	make the table as wide as the line
`box`, `doublebox`, `allbox`	a box round the table, a double one, or a box round every entry
`tab(x)`	separate entries with x instead of a tab
`linesize(n)`	lines n points thick
`delim(xy)`	eqn delimiters inside the table
`decimalpoint(c)`, `nospaces`, `nokeep`, `nowarn`	(GNU) decimal separator, trim spaces, allow page breaks within rows, quiet

Format: one letter per column

`l`, `r`, `c`	left, right, centred
`n`	numbers aligned on the decimal point
`a`	left-aligned and indented, for sub-entries
`s`	span: the entry to the left extends into this column
`^`	span: the entry above extends into this row
`_`, `=`	a single or double rule in place of the entry
`|`, `||`	a vertical line between columns

After the letter, modifiers: `b` bold, `i` italic, `f(CR)` any font, `p12` type size, `w(1i)` minimum width, `e` equal widths, `x` expand this column, `t` top of a vertical span, and a number for the gap to the next column in ens (default 3). Rows of a format can be on separate lines or separated by commas: `cB cB, l n.`

Data

`_`, `=` alone on a line, a horizontal rule across the table; alone in an entry, a rule across that cell.

`\^` in an entry, the entry above spans down into it.

`T{` **...** `T}` a text block: an entry of several lines, filled.

`\&` an empty entry made visible, or before a leading `.` in an entry.

`.T&` a new format for the rows that follow, with no more columns than before.

`.TS H`**,** `.TH` with ms, the rows before `.TH` repeat at the top of each page the table spans.

See Also

`roff_table`, `ms`, `troff`, `eqn`. In a terminal: `man tbl`; and M. E. Lesk, *Tbl: A Program to Format Tables*, Bell Laboratories, 1976.

`troff` *troff requests, escapes and characters for knitroff documents*

Description

Below the ms macros (`ms`), a knitroff document is plain troff: requests on lines that begin with a dot, and escapes that begin with a backslash. This page covers what reports need, following groff(7) and groff_char(7) for groff 1.24.1.

Writing troff text

requests a line beginning with `.` or `'` is a request or a macro call, such as `.PP`. To begin a text line with either character, put `\&` (a zero-width character) before it.

escapes a backslash begins an escape, such as `\fB`. Write a literal backslash as `\e` or `\[rs]`.

breaks a blank line, or a line beginning with a space, breaks the paragraph. In filled text, start each sentence on a new line: troff then puts the right space after its period.

comments `.\"` begins a comment line; `\"` begins a comment at the end of a line.

knitroff inline results (`` `r expr` ``) and `.SR` strings are escaped for you; output in listings is too.

Requests

ms resets indentation, line length and type size at each paragraph and heading, so inside an ms document prefer its own macros and registers (`.RS`, `\n[LL]`, `.LG`) to `.in`, `.ll` and `.ps`.

`.br`	break the output line
`.sp [N]`	break, and space down N (default one line)
`.bp`	begin a new page
`.ne N`	begin a new page unless N of space remains
`.ce [N]`	centre the next N input lines (default 1); `.ce 0` stops
`.nf`, `.fi`	no filling (lines as in the source) / filling again
`.ad [l|r|c|b]`, `.na`	adjust lines left, right, centre or both margins / no adjustment
`.ft F`	change to font F: R, I, B, BI, CR (constant width), or P (previous)
`.ps N`	type size N points; `.ps +2` larger
`.vs N`	vertical spacing (baseline to baseline)
`.ll N`	line length
`.in N`, `.ti N`	indentation / indentation of the next line only
`.po N`, `.pl N`	page offset (left margin) / page length
`.ta N ...`	tab stops, such as `.ta 1i 2i` or `.ta 2iR` (right-aligned)
`.tl 'l'c'r'`	a three-part title line
`.hy [N]`, `.nh`	hyphenation on / off
`.ul [N]`	underline (italic in troff) the next N lines
`.ds name text`	define a string, used as `\*[name]`; a leading `"` keeps leading spaces
`.nr name N`	set a number register, used as `\n[name]`
`.de name ... ..`	define a macro; its arguments are `\\$1`, `\\$2` ...
`.so file`	read in another file here
`.ig ... ..`	ignore input up to `..`

Escapes

`\fB`, `\fI`, `\fR`, `\fP`	bold, italic, roman, previous font; `\f(CR` or `\f[CR]` constant width
`\s+2`, `\s-2`, `\s0`	type size up, down, back to the previous
`\*x`, `\*(xx`, `\*[name]`	the string x, xx, or name; `.SR` defines them
`\nx`, `\n(xx`, `\n[name]`	the value of a register; `\n[%]` is the page number
`\(xx`, `\[name]`	a special character, such as `\(em` (below)
`\e`	a backslash
`\-`	a minus sign. knitroff writes the - of code this way, so groff does not set it as a hyphen; in the term
`\&`	nothing, zero width: protects a leading . or '
`\~`, `\ ` (backslash, space)	an unbreakable space, adjustable / of fixed width
`\0`, `\\|`, `\^`	spaces the width of a digit, 1/6 em, 1/12 em
`\u`, `\d`	up / down half an em, for simple super- and subscripts
`\v'N'`, `\h'N'`	move vertically (down if positive) / horizontally
`\l'N'`, `\L'N'`	draw a horizontal / vertical line of length N
`\w'text'`	the width of text, for use in measurements
`\zc`	print c without moving: zero width
`\c`	join the next input line to this one
`\%`	a hyphenation point; at the start of a word, do not hyphenate it
`\"`	a comment, to the end of the line

Units

Measurements take a unit after the number, as in `.sp 0.5v` or `.in 2n`.

`i`, `c`, `p`, `P`	inch, centimetre, point (1/72 inch), pica (12 points)
`m`, `n`	em and en: the width of M and half of it, at the current type size
`v`	one line of vertical spacing (a "vee")
`u`	the device's basic unit

Without a unit, `.sp` and `.ne` count lines (v), `.ps` and `.vs` points, and most other requests ems.

Special characters

UTF-8 input works (knitroff runs groff with `-k`), but these names are portable to any troff and to the terminal. Write `\(xx` for two-character names or `\[name]` for any.

`\(em`	em dash	`\(en`	en dash
`\(hy`	hyphen	`\(mi`	minus
`\(bu`	bullet	`\(de`	degree
`\(co`, `\(rg`, `\(tm`	copyright, registered, trademark	`\(sc`	section
`\(dg`, `\(dd`	dagger, double dagger	`\(ct`, `\(Po`, `\(Eu`	cent, pound, euro
`\(mu`, `\(di`	times, divide	`\(+-`	plus or minus
`\(<=`, `\(>=`	less, greater or equal	`\(!=`, `\(==`, `\(~=`	not equal, equivalent, approximately
`\(->`, `\(<-`	arrows	`\(if`, `\(pd`, `\(gr`	infinity, partial, gradient
`\(sr`, `\(is`	square root, integral	`\(sq`, `\(ci`	square, circle

`\(lq`, `\(rq`	double quotes	`\(oq`, `\(cq`	single quotes
`\(aq`, `\(dq`	ASCII ' and "	`\(ga`, `\(ha`, `\(ti`	ASCII ‘, ^, ~
`\(rs`, `\(sl`	backslash, slash	`\(fm`	prime (foot mark)
`\(*a` ... `\(*w`	Greek alpha to omega	`\(*A` ... `\(*W`	capital Greek

The full list is in `man groff_char`.

See Also

`ms`, `eqn`, `tbl`. In a terminal: `man 7 groff`, `man groff_char`, `info groff`.

Index

AB *(ms)*, 6
AE *(ms)*, 6
AI *(ms)*, 6
AU *(ms)*, 6

DE *(ms)*, 6
DS *(ms)*, 6

eqn, 2, *2*, *8*, *10*, *13*, *16*
equations *(eqn)*, 2

FE *(ms)*, 6
FS *(ms)*, 6

groff *(troff)*, 14
groff_available, 3, *11*

hooks_roff, 4

IP *(ms)*, 6

KE *(ms)*, 6
knit, *4*, *5*
knit_roff, *2*, *4*, 5, *10*, *11*
knitroff *(knitroff-package)*, 2
knitroff-package, 2
KS *(ms)*, 6

LP *(ms)*, 6

ms, *2*, *3*, 6, *13*, *14*, *16*
ms-macros *(ms)*, 6

neqn *(eqn)*, 2
NH *(ms)*, 6

PP *(ms)*, 6

QP *(ms)*, 6

render_roff, *2*–*5*, 10, *10*–*12*
roff *(troff)*, 14
roff_table, *5*, 11, *12*, *13*

SH *(ms)*, 6

tables *(tbl)*, 12
tbl, *2*, *3*, *8*, *10*, 12, *16*
TL *(ms)*, 6
troff, *2*, *3*, *6*, *10*, *13*, 14
troff-requests *(troff)*, 14