me Reference Manual

for groff me *version 1.23.0**

Eric P. Allman

Adapted for groff *by James Clark and G. Branden Robinson*

This document describes in terse form the features of the *me* macro package for *groff*. Some familiarity with the latter is assumed. Specifically, the reader should understand breaks; filling; adjustment; font selection; type sizes; the definition and use of registers, strings, and macros; and typographical units of measurement: points, ems, ens, and vees. For a more casual introduction to text processing, refer to "Writing Papers with *groff* using −*me*".

Many of the package's rendering parameters can be adjusted through registers, strings, and macro calls. Macro arguments are denoted with lowercase letters for numeric values (suffixed with a scaling unit if appropriate) and with uppercase for text arguments. Default parameter values are shown in square brackets. The notation $\pm n$ indicates a numerical value with an optional leading sign. Without the sign, it assigns a value *n*. With it, it expresses alteration of an existing value by the amount *n*. *me*'s font registers may be set only to mounting positions. Position zero tells *me* to perform no font change; the font of the preceding text is used instead. Position zero is a "pseudo-font"; that is, it is interpreted and handled by the macros. In contrast, using position zero may produce unexpected results in a font selection escape sequence such as **\f0**, because those are interpreted directly by *troff*. *me*'s type size registers support only integral values in points. Changes to parameters that affect the layout of the page (notably page length and vertical margins) should be done before calling any paragraphing or sectioning macros. Normally, vertical space is suppressed at the top of a page if no text has yet appeared (page headers notwithstanding).

groff measures distances in device-specific basic units, so it is often necessary to specify measurements with a scaling unit. For instance, to set the paragraph indentation to eight ens, input ".nr pi 8n". ".nr pi 8" makes the paragraph indentation eight basic units, or 8/72,000 inches on *grops, groff*'s PostScript output driver.

Registers and strings with names of the form **$*x*** may be used in expressions and text but should not be changed. Macros with names of the form **$*x*** are *hooks,* called internally by *me*, and can be redefined to alter their function. This may be a sensitive operation; look at the macro definition in the *e.tmac* file before changing it. If you don't require a given hook, (re)define it as empty instead of removing it.

Objects in *me* follow a rigid naming convention. To avoid conflict, any user-defined register, string, or macro names should be single numerals or uppercase letters, or any longer character sequence including at least one uppercase letter.[1] The names employed by any preprocessors in use should also not be repurposed.

---

* based on *e.tmac* 8.1 from 4.4BSD

[1] For portability between BSD and *groff me*, limit names to two characters, and avoid names starting with **[** (left square bracket).

1. Paragraphing

.lp — Begin left-aligned paragraph. Centering, right-alignment, and underlining are canceled. The font is set to **\n(pf** [1], the type size to **\n(pp** [10p], and vertical spacing (text baseline spacing) to **\n(tv** [120]% of the type size. Vertical space of **\n(ps** [0.35v] is inserted before the paragraph. The indentation is reset to **\n($i** [0] plus **\n(po** [0] unless the paragraph is inside a display (see **.ba**). At least the first two output lines of the paragraph are kept together on a page.

.pp — Like **.lp**, but apply a (further) indentation of **\n(pi** [5n] to the first output line.

.ip *T i* — Like **.lp**, but set the paragraph with an optional hanging tag *T* [empty] and the remainder indented by *i* [**\n(ii** [5n]]. The tag *T* is "exdented" (the opposite of *in*dented). Any spaces in *T* must be unbreakable. If *T* does not fit in the space *i,* **.ip** breaks the line after *T*.

.np — Like **.ip**, but tag the paragraph with a number in parentheses, starting at 1. Each subsequent **.np** call increments it. The paragraph number is stored in **\n($p** and reset to 1 by **.lp**, **.pp**, **.bu**, **.sh**, and **.sx**.

.bu — Like **.ip**, except that the tag is a bullet (•). No vertical space is inserted between adjacent bulleted paragraphs, enabling the construction of compact itemized lists.

2. Sectioning

Numbered sections are similar to paragraphs except that a section number of the form "1.2.3" is automatically generated for each. *me* supports up to six levels of sectioning; any given section has a *depth* that determines the quantity of components (separated by dots) shown in its section number. Unnumbered section headings are similar, except that no number is prefixed to the section title.

.sh ±*n T a b c d e f* — Begin section with numbered heading of depth ±*n* [+0] and optional title *T* [empty]. Vertical space of **\n(ss** [1v] precedes the heading. The components of the section number are maintained in **\n($1** through **\n($6**; combined, they are available in ***($n**. Automatic section numbering at all depths begins at 1. The heading is set in font **\n(sf** [3] at size **\n(sp** [10p] and uses the indentation of its parent depth; a further indentation of **\n(so** [0] is applied only to its output line. The section number is followed by a period, after which the typeface is set to the paragraph font **\n(pf** at size **\n(pp** so that content may follow immediately on the same output line. **.sh** ensures that there is enough room to format the section heading with the beginning of a paragraph on the same page (about 3 lines total). If any of *a* through *f* [all empty] is specified, each component of the section number is assigned the corresponding argument instead of being automatically adjusted. A hyphen ("−") for a component argument prevents its alteration. If *T* is an underscore ("_"), the section depth and numbering are reset, the base indentation is not, and nothing is output—this is useful to automatically coordinate section numbers with chapter numbers. If all components are zero, no number is output.

.sx ±*n* — Go to section depth ±*n* [−1], but emit no section heading and do not increment the section number. This has the effect of an **.lp** call at the new depth.

.uh *T* — Begin unnumbered section. Like **.sh**, without the section numbering features.

.$p *T B n* — Output section heading. *T* is the title, *B* is the concatenated number, and *n* is the depth. These parameters are not always present; **.sh** passes all three, **.uh** only the first, and **.sx** all three, but the first two are empty. If *n* is present, the section indentation is set to **\n(si** times *n*.

.$0 *T B n* — This hook macro, normally empty, is called automatically by **.sh** and **.uh** after they call **.$p**, and is passed the same arguments. You can define it to, for instance, automatically put each section title into a table of contents using **.(x** and **.)x**.

.$*n* — These hook macros (where *n* is an integer 1–6) are called by **.$p** just before it outputs a section heading of depth *n*. They could be used to obtain section depth-dependent spacing.

3. Headers, Footers, and Margins

Headers and footers are known as *three-part titles* for their left, middle, and right-hand components *L, M,* and *R.* The components are separated by a *delimiter,* shown here as a neutral apostrophe, but which can be any character that does not occur in any of the components. Any defined title is placed on the page automatically, set in font **\n(tf** [3] at size **\n(tp** [10p]. Each title definition applies starting with the *next* page; titles intended for output on the first page should be defined prior to any sectioning or paragraphing macro calls. A title must be quoted if it contains more than two adjacent spaces or more than eight spaces total.

The placement of headers and footers is controlled by four registers initialized to values scaled with respect to the default type size and vertical spacing. **\n(hm** [4v] is the distance from the top of the page to the top of the header, **\n(fm** [3v] that from the bottom of the page to the bottom of the footer, **\n(tm** [7v] that from the top of the page to the top of the text, and **\n(bm** [6v] that from the bottom of the page to the (nominal) bottom of the text.

me uses *troff*'s environment 0 for body text, 1 for keeps (see below), and 2 for headers and footers. The default line length is two inches less than the page width (e.g., 6.5i on U.S. letter paper). The default page offset (left margin) is determined by the output device; it is typically 1.0i on typesetting devices and 0 on terminals.

.xl ±*n* — Set line length to *n* [see above] only in the current environment.

.ll ±*n* — Set line length to to *n* [see above] in all environments used by *me*,[2] and store it in **\n($l**. This macro should not be used after output has begun, and particularly not in multi-column layouts.

.he '*L*'*M*'*R*' — Define header to be placed at the top of every page.

.fo '*L*'*M*'*R*' — Define footer to be placed at the bottom of every page.

.eh '*L*'*M*'*R*' — Define header to be placed at the top of every even-numbered page.

.oh '*L*'*M*'*R*' — Define header to be placed at the top of every odd-numbered page.

.ef '*L*'*M*'*R*' — Define footer to be placed at the bottom of every even-numbered page.

.of '*L*'*M*'*R*' — Define footer to be placed at the bottom of every odd-numbered page.

.hx — Suppress headers and footers on the next page.

.$h — Output the page header. Care should be taken when redefining this macro. Doing so can disable the **.he**, **.fo**, **.eh**, **.oh**, **.ef**, and **.of** macros, as well as the chapter-style title feature of **.+c**.

.$f — Output the page footer; the same comments apply as to **.$h**.

.$H — This hook macro is called at the top of each page (after the header and any pending floating keeps are emitted) and of each column in multi-column mode. Use it for column headings.

4. Displays

Display macros enclose material; their names are of the form **.(*x*** and **.)*x***, and must be paired. Not all display types can nest with themselves or each other. All displays except centered blocks and long quotations are preceded and followed by an extra **\n(bs** [0.35v] of vertical space and use a vertical spacing of **\n(dv** [120]%. Long quotation pre- and post-space is stored in a dedicated register, while centered blocks have none; both use the vertical spacing of normal text, **\n(tv**.

.(l *A F* — Begin list. Text until **.)l** is set in font **\n(df** [0] with filling disabled. If *A* [**I**] is **I**, the list is indented by **\n(bi** [4m]; if **M**, it is set at the left margin; if **L**, it is left-aligned (distinguishably from **M** only if the base indentation is not zero); and if **C**, the list is centered on a line-by-line basis. If *F* [empty] is **F**, filling is enabled.

.)l — End list.

---

[2] *me* uses only the three environments of AT&T *troff*, but in GNU *troff*, the user can create additional ones. **.ll** works like **.xl** for the latter.

.(q Begin long quotation. Text until **.)q** is filled, indented on the left and right by **\n(qi** [4n], preceded and followed by **\n(qs** [0.35v] space, and set at type size **\n(qp** [−1p].

.)q End long quotation.

.(b *A F* Begin a block, a form of *keep: me* tries to avoid breaking a page or column between **.(b** and **.)b**. Such a break is allowed anyway if respecting the keep would leave more than **\n(bt** [0] vees of blank space below it. If **\n(bt** is zero, this threshold feature is disabled: the break will not occur within the keep. The font and *A* and *F* arguments are handled as with **.(l**.

.)b End block.

.(z *A F* Begin floating keep. Like **.(b**, except that *A* defaults to **M** and the keep *floats* to the bottom of the page or column if it fits, or the top of the next otherwise. Its position relative to adjacent formatted text is thus flexible. **\n(zs** [1v] space precedes and follows a floating keep.

.)z End floating keep.

.(c Begin centered block. Input until **.)c** is collected, its longest output line centered, and the remainder aligned with respect to that line; contrast with the line-by-line centering obtained with "**.(b C**".

.)c End centered block.

5. Annotations

Some annotation macros are also mandatorily paired enclosures, as display macros are.

.(d Begin delayed text. Input until **.)d** is saved for later output by **.pd**. Delayed text regions are numbered starting at 1; the number is kept in **\n($d** and in the associated string ***#**. Endnotes are one application.

.)d End delayed text. **\n($d** and the associated string ***#** are incremented.

.pd Emit delayed text: output everything accumulated with **.(d** since the last call to **.pd**. The delayed text number is reset to 1. This might be used at the end of each chapter.

.(f Begin footnote. Input until **.)f** is saved for output when the foot of the page is reached. Footnotes are numbered starting at 1; the number is kept in **\n($f** and in the associated string *****. The text of the footnote floats to the bottom of the page and set in font **\n(ff** [1] at size **\n(fp** [8p]. Each entry is preceded by **\n(fs** [0.2v] space, indented **\n(fi** [3n] on its first line, and "undented" (indented from the right margin) by **\n(fu** [0]. Footnotes line up underneath columns in the page layout. If the footnote's content does not fit on the page, it carries over to the next.

.)f End footnote. **\n($f** and the associated string ***** are incremented if the latter has been interpolated outside a footnote.

.$s Output footnote separator: draw a horizontal line up to 2 inches wide. Called by **.(f**. Any redefinition should produce output no more than one vee in height.

.(x *X* Begin index entry. Input until **.)x** is saved in an index named *X* [**x**] until called up with **.xp.** Each entry is preceded by **\n(xs** [0.2v] of vertical space and undented by **\n(xu** [0.5i]; that amount of space is reserved on the right end of the line for the (possibly affixed) page number.

.)x *P A* End index entry. The entry is rendered with a *leader* (row of dots) up to an affix (such as a volume number and separating punctuation) *A* [empty] followed by *P* [**\n%**] right-aligned on the last line of the entry. If *A* is specified, *P* must be as well. If *P* is an underscore, the leader, *P,* and any *A* are suppressed.

.xp *X* Emit index *X* [**x**] using the current font and type size with display vertical spacing (**\n(dv**).

6. Columnated Output

.2c ±*s n* — Enter multi-column mode, formatting text in *n* [2] columns of equal width. The column separation ("gutter") is set to ±*s* [4n], saved in **\n($s**, and persists until changed by this macro. The *n* columns with $n-1$ gutters fill the single-column line length; each column's line length is stored in **\n($l**. The current column number is in **\n($c** and the number of columns in **\n($m**.

.1c — Leave multi-column mode; return to single-column layout.

.bc — Begin new column. Like **.bp**, but begin a column on a new page only if necessary.

7. Type Size and Font Styles

.sz ±*p* — Set the type size to ±*p* [10p], and the vertical spacing per **\n(tv**. These do *not* persist beyond many *me* macro calls: paragraph macros reset the type size and vertical spacing. Similarly, headers/footers, section headings, long quotations, footnotes, and chapter titles alter them.

me assumes that font positions 1 through 4 correspond to roman, italic, bold, and bold-italic typefaces, respectively. Use the **.fp** request to mount your preferred fonts in these positions.

The following macros style or decorate an argument *W* and accept an optional second argument *X,* formatted as the previous text without intervening space. To suppress a word break after the call, populate or suffix *X* with **\c**.

.r *W X* — Set *W* in the roman style. If no parameters, change to the roman style.

.i *W X* — Set *W* in italics. If no parameters, change to the italic style.

.b *W X* — Set *W* in boldface. If no parameters, switch to the bold style.

.bi *W X* — Set *W* in bold italics. If no parameters, switch to the bold-italic style.

.q *W X* — Quote *W*, surrounding it with with **“** and **”**.

.sm *W X* — Set *W* one point smaller than the surrounding type. It has no effect in *nroff* mode.

.u *W X* — Underline *W.* On typesetting output devices, this is true underlining, in contrast to the **.ul** request, which changes to the "underline font" (usually italics in *groff*). On terminal output devices, *W* is surrounded by underscores. This argument should not be subject to adjustment, hyphenation, or breaking; **.u** is reliable only when filling is disabled.

.bx *W X* — Set *W* in a box. On typesetting output devices, *W* is surrounded by rules (lines). On terminal output devices, *W* is simulated with ISO 6429 color escape sequences or surrounded by vertical bars. It is not marked specially on *groff*'s **html** output device. This argument should not be subject to adjustment, hyphenation, or breaking; **.bx** is reliable only when filling is disabled.

8. Preprocessor Support

.EQ *A T* — Begin *eqn*(1) equation. If *A* [**C**] is **C**, the equation is centered; if **I**, indented by **\n(bi** [4m]; and if **L**, left-aligned. *T* [empty] is a title aligned to the right margin next to the equation. See "Typesetting Mathematics – User's Guide" by Brian W. Kernighan and Lorinda L. Cherry.

.EN *C* — End *eqn* equation. If *C* [empty] is **C**, and the equation is followed immediately by another **.EQ**, the texts of each are centered together. Otherwise, the equation is typeset within the page or column with **\n(es** [0.5v] space above and below it.

.TS *H* — Begin (start) *tbl*(1) table. Tables are preceded and followed by **\n(bs** of space. If *H* [empty] is **H**, you can call **.TH** after some table rows to repeat them across columns or pages. If you want a table to float, surround the **.TS** and **.TE** calls with **.(z** and **.)z** calls.

.TH — With "**.TS H**", end the *tbl* table header.

.TE — End *tbl* table.

.PS *h w* — Begin *pic*(1) picture. *h* [empty] and *w* [empty], if present, scale the image.

.PE — End *pic* picture, leaving the drawing position at the bottom of the picture.

.PF End *pic* picture, leaving the drawing position at the top of the picture.

.IS Begin *ideal*(1)[3] picture.

.IE End *ideal* picture, leaving the drawing position at the bottom of the picture.

.IF End *ideal* picture, leaving the drawing position at the top of the picture.

.GS *A* Begin *grn*(1) picture. *A* [**C**] can be **C**, **L**, or **R** to center, left-, or right-align the picture.

.GE End *grn* picture, leaving the drawing position at the bottom of the picture.

.GF End *grn* picture, leaving the drawing position at the top of the picture.

9. Formal Documents

.tp Begin title page. Unusually, **.sp** calls at the top of the page *are* honored. Headers and footers are suppressed. The page number is not incremented for this page.

.++ *S H* Begin a *segment* of an organized document, affecting the values and formatting of chapter and page numbers. The chapter number in **\n(ch** is reset to 0. A segment uses Arabic numerals for chapter and page numbers except where noted. The mandatory segment type argument *S* must be one of **C** for chapters (of the main matter), **A** for appendices (uppercase alphabetical chapter numbers), **P** for preliminary ("front") matter (such as a foreword; lowercase Roman page numbers), **AB** for an abstract (page numbering restarts at 1), or **B** for "back" matter, such as a bibliography. **C** or **A** may be prefixed with **R**, which specifies a restart of page numbering when **.+c** is called within the applicable segment. If present, the *H* [empty] parameter defines the new header, which must be delimited as a three-part title—if it contains spaces, it must furthermore be quoted. To include the chapter number in *H,* use the input sequence **\En(ch**.[4] For example, "**.++ RA |||\En(ch.%|**" numbers appendices and pages as ***A.n*** in the right-hand header, where *A* is the appendix letter and *n* is the page number. Each subdivision of a segment (each chapter, appendix, etc.) should be preceded by a **.+c** call. It is easier when using *troff* to put the front matter at the end of the paper, so that the table of contents can be collected and put out; this material can then be resequenced by a tool that processes the output format, or physically moved to the beginning of the printed document.

.+c *T* Begin chapter (or appendix). Reset the footnote number in **\n($f** to 1. If the segment type is **RA** or **RC**, reset the page number in **\n%** to 1. If **.++** has not been called, a segment type of **C** is assumed. If a header is defined, it replaces the footer on the first page of each chapter. If a title *T* [empty] is supplied, **.$c** is called.

.$c *T* Format a chapter heading centered in boldface. The text ***(wc** [**Chapter**] is output if the segment type is **C** or **RC,** or that in ***(wa** [**Appendix**] if the segment type is **A** or **RA.** The chapter number is incremented and output. If a chapter title *T* is present, it is formatted the same way, preceded by vertical space. If the segment type is any of **C**, **RC**, **A**, or **RA**, **$C** is called.

.$C *K N T* This hook macro is called by **.+c** and **.$c**. *K* is the chapter or appendix term supplied by **$c**, *N* is the chapter or appendix number, and *T* is its title. This hook can be used to insert chapter titles into a table of contents.

10. Miscellaneous

.ld Update the **\n(y2** and **\n(y4** registers based on the current value of **\n(yr**, reinitialize the localized strings ***(dw**, ***(mo**, ***(wa**, and ***(wc**, and reset ***(td** based on the foregoing.

.re Reset tab stops to every 0.5i.

---

[3] *groff* does not provide an implementation of the *ideal* preprocessor.

[4] If you wish to avoid use of the *groff* **\E** extension, fully *eight* backslashes must precede "n(ch".

.ba ±*n* — Set the base indentation to ±*n* [0]. Paragraphs, sections, and displays are indented by at least this amount, stored in **\n($i**. Titles and footnotes are unaffected.

.hl — Break and draw a horizontal line of the current line length. You might call **.hl** at the beginning and/or end of a floating keep to differentiate the text from a figure.

.ep — End this page, but do not begin the next page. Useful for forcing out footnotes, but other than that hardly ever used. Must be followed by a **.bp** request or the end of input.

11. *roff* Support[5]

.ix ±*n* — Equivalent to " **'in** ±*n*".

.bl *n* — Equivalent to "**.sp** *N*" inside a **.(b** block.

.m1 ±*n* — Set/adjust the space between the top of the page and the header to/by ±*n* [4v].

.m2 ±*n* — Set/adjust the space between the header and the first line of text to/by ±*n* [2v].

.m3 ±*n* — Set/adjust the space between the bottom of the text and the footer to/by ±*n* [2v].

.m4 ±*n* — Set/adjust the space between the footer and the bottom of the page to/by ±*n* [4v].

.pa ±*n* — Equivalent to "**.bp** ±*n*".

.ro — Equivalent to "**.af % i**".

.ar — Equivalent to "**.af % 1**".

.n1 *C* — Number output lines starting from 1. If *C* [empty] is **C**, the line length is reduced and correspondingly indented; otherwise, the page offset is reduced to accommodate the line number. The amount in all cases is **\n(no** [\w'0000'u (the width of four numerals in the current font)].

.n2 ±*n* — Stop numbering output lines if *n* missing, otherwise continue with the number modified by ±*n*.

.sk — Leave the next output page blank except for headers and footers. This can reserve space for a full-page diagram produced externally and composited in later. (To reserve only a partial page, use "**.sv** ***n***", where *n* is the amount of space required. This space will be output immediately if there is room, and otherwise at the top of the next page. If *n* is greater than the space available on an empty page, none will be output.)

12. Predefined Strings

****** — Interpolate footnote number bracketed by superscripting strings: ***{\n($f*}**.

***#** — Interpolate delayed text number surrounded by square brackets: **[\n($d]**.

***{** — Begin superscripting: move upward and reduce the type size. The line height is increased above by **\n(sx** [0.2m] to accommodate the superscript. In *nroff* mode, output a left square bracket.

***}** — End superscripting: reverse the motion and type size change of ***{**. In *nroff* mode, output a right square bracket.

***<** — Begin subscripting: move downward and reduce the type size. The line height is increased below by **\n(sx** to accommodate the subscript. In *nroff* mode, output a left angle bracket.

***>** — End subscripting: reverse the motion and type size change of ***<**. In *nroff* mode, output a right angle bracket.

---

[5] These facilities are provided to aid migration of "old *roff*" documents. See *roff*(7) for a history of *roff*-related typesetting systems.

groff me provides localized strings to support documents in several input languages; see the *groff_tmac*(5) man page. The date-related strings are initialized based on the current date when the formatter runs. To encode a different date in your document, change the *troff* date registers as desired and then call **.ld** to update the strings. The day of the week is *not* recomputed from a new calendar date; **\n(dw** must be modified manually. The strings can also be defined directly.

***(dw** Weekday name.

***(mo** Month name.

***(td** The date in a form suitable for correspondence. Construct other date formats with ***(dw**, **\n(dy** (day of the month), ***(mo**, **\n(mo** (month of the year), **\n(y4** (year), and **\n(y2** (year of the century).

***(wa** The term **.$c** uses for "appendix".

***(wc** The term **.$c** uses for "chapter".

13. Special Characters

For modern documents, see the *groff_char*(7) man page.

For rendering of legacy *me* documents, the macro package defines several strings that construct accent marks and two symbols from mathematical set theory. These strings are limited in multiple respects: they can have a crude appearance, they are unrecognizable on character-cell video terminals because they rely on overstriking, and they cannot in general be "stacked", as is required to correctly render words in (for example) Vietnamese. Preferable *groff* alternatives are presented alongside them in the following table.

Name	Usage	Example	Output	*groff*	Output
Acute	*'	a*'	á	\[a aa]	á
Grave	*`	e*`	è	\[e ga]	è
Umlaut	*:	u*:	ü	\[u ad]	ü
Tilde	*~	n*~	ñ	\[n a~]	ñ
Caret	*^	e*^	ê	\[e a^]	ê
Cedilla	*,	c*,	ç	\[c ac]	ç
Caron	*v	s*v	š	\[s ah]	š
Ring	*o	A*o	Å	\[A ao]	Å
There exists	*(qe		∃	\[te]	∃
For all	*(qa		∀	\[fa]	∀

Three further string definitions cause no particular problems but have no advantage over *groff* special character escape sequences apart from portability to BSD Unix systems. (The last lacks even that distinction, since Unix Version 7 *troff* supported the **\(em** special character.)

***(lq** Left double quotation mark. Use **\(lq** instead.

***(rq** Right double quotation mark. Use **\(rq** instead.

***–** Em dash. Use **\(em** instead.

Acknowledgments

Allman would like to thank Bob Epstein, Bill Joy, and Larry Rowe for having the courage to use the *me* macros to produce non-trivial papers during the development stages; Ricki Blau, Pamela Humphrey, and Jim Joyce for their help with the documentation phase; peter kessler for numerous complaints, most accompanied by fixes; and the plethora of people who have contributed ideas and have given support for the project.

Summary

Almost all macros, strings, and registers available in the *me* package are summarized below; the legacy strings from section “Special Characters” are omitted. Selected *troff* requests, registers, and escape sequences are included as well; those listed can generally be used with impunity. The list is sorted by Unicode code point (basic Latin subset) keyed on the identifier: escape and control characters are disregarded, as are the escape function selectors for register and string interpolation, **n** and *****.

Each row contains the name of the macro (or request), register, string, or escape function; a code for the name’s type (one of the foregoing); and its description. The type codes are **M** for a macro or *troff* request (shown prefixed with the control character **.**, but the no-break control character **'** can also be used), **S** for a string (shown using its interpolation syntax, ***** or ***(**), **R** for a register (shown using its interpolation syntax, **\n** or **\n(**), and **F** for a *troff* built-in function (shown prefixed with the escape character ****).

The type code can carry a variety of annotations. A section sign § indicates features interpreted by *troff* rather than the *me* package. The dagger † and double dagger ‡ identify user-definable macros. Those marked with ‡ have default definitions; changing them may have unexpected side effects. The degree sign ° notes features requiring AT&T device-independent *troff* extensions (ca. 1981), and a square □ labels *groff* extensions.

Name	Type	Description
newline	F§	continue input line
\"	F§	comment (to end of input line)
*#	S	delayed text marker
\$*n*	F§	interpolate *n*th macro argument
\n($0	R	section depth
.$0	M†	post-section heading hook
\n($1	R	first section number component
.$1	M†	pre-section heading depth 1 hook
\n($2	R	second section number component
.$2	M†	pre-section heading depth 2 hook
\n($3	R	third section number component
.$3	M†	pre-section heading depth 3 hook
\n($4	R	fourth section number component
.$4	M†	pre-section heading depth 4 hook
\n($5	R	fifth section number component
.$5	M†	pre-section heading depth 5 hook
\n($6	R	sixth section number component
.$6	M†	pre-section heading depth 6 hook
.$C	M†	post-chapter title heading hook
.$H	M†	page/column heading hook
\n($c	R	current column number
.$c	M‡	output chapter number and title
\n($d	R	delayed text number
\n($f	R	footnote number
.$f	M‡	output footer
.$h	M‡	output header
\n($i	R	paragraph base indentation
\n($l	R	column width
\n($m	R	number of available columns
*($n	S	concatenated section number
\n($p	R	numbered paragraph number
.$p	M‡	output section heading
\n($s	R	column spacing (indentation)
.$s	M‡	output footnote area separator
\%	F§	control hyphenation
\n%	R§	current page number
\&	F§	dummy character
\(*xx*	F§	interpolate special character *xx*
.(b	M	begin block
.(c	M	begin centered block
.(d	M	begin delayed text
.(f	M	begin footnote
.(l	M	begin list
.(q	M	begin long quotation
.(x	M	begin index entry
.(z	M	begin floating keep
\'	F§	acute accent
.)b	M	end block
.)c	M	end centered block
.)d	M	end delayed text
.)f	M	end footnote
.)l	M	end list
.)q	M	end long quotation
.)x	M	end index entry
.)z	M	end floating keep
**x*	F§	interpolate string *x*
*(*xx*	F§	interpolate string *xx*
**	S	footnote marker
.++	M	set document segment type
.+c	M	begin chapter or appendix
\–	F§	minus sign
*–	S	em dash
\n(.$	R§	macro parameter count
\n(.i	R§	current indentation
\n(.l	R§	current line length
\n(.p	R§	current page length
\n(.s	R§	current type size in points
\0	F§	unadjustable digit-width space
.1c	M	end multi-column layout
.2c	M	begin multi-column layout

Name	**Type**	**Description**
*<	S	begin subscripting
*>	S	end subscripting
.EN	M	end *eqn* equation
.EQ	M	begin *eqn* equation
.GE	M°	end *grn* picture with position at bottom
.GF	M°	end *grn* picture with position at top
.GS	M°	start *grn* picture
.IE	M°	end *ideal* picture with position at bottom
.IF	M°	end *ideal* picture with position at top
.IS	M°	start *ideal* picture
\L'*d*'	F§	draw vertical line of length *d*
.PE	M°	end *pic* picture with position at bottom
.PF	M°	end *pic* picture with position at top
.PS	M°	start *pic* picture
.TE	M	end *tbl* table
.TH	M	end heading for multi-page *tbl* table
.TS	M	begin *tbl* table
\^	F§	1/12 em unbreakable hair space
_	F§	underrule
\`	F§	grave accent
.ad	M§	set text adjustment mode
.af	M§	assign register format
.am	M§	append to macro
.ar	M	use Arabic numerals for page numbers
.as	M§	append to string
.b	M	embolden argument
.ba	M	set base indentation
.bc	M	begin new column
.bi	M	embolden and italicize argument
\n(bi	R	display (block) indentation
.bl	M	insert space (even at page top; cf. **.sp**)
\n(bm	R	distance from text area to page bottom
.bp	M§	begin new page
.br	M§	break output line
\n(bs	R	display (block) pre/post space
\n(bt	R	block threshold for keeps
.bx	M	box argument
\c	F§	continue output line
.ce	M§	center output lines
\n(ch	R	current chapter number
.cp	M§□	toggle AT&T *troff* compatibility mode
.de	M§	define macro
\n(df	R	display font
.do	M§□	operate with compatibility mode off
.ds	M§	define string
\n(dv	R□	vertical spacing of displayed text (%)
\n(dw	R§	day of the week
*(dw	S	weekday name
\n(dy	R§	day of the month
\e	F§	printable version of escape character ****
.ef	M	set even-numbered page footer
.eh	M	set even-numbered page header
.el	M§	else part of **.ie** conditional
.ep	M	end page
\n(es	R	equation pre/post space
\f*f*	F§	inline font change to font *f*
\f(*ff*	F§	inline font change to font *ff*
.fc	M§	set field characters
\n(ff	R	footnote font
.fi	M§	enable filling
\n(fi	R	footnote indentation (first line only)
\n(fm	R	footer margin
.fo	M	set footer
.fp	M	mount font at position
\n(fp	R	footnote type size in points
\n(fs	R	footnote prespace
\n(fu	R	footnote undent (right indentation)
\h'*d*'	F§	move distance *d* horizontally
.hc	M§	set hyphenation character
.he	M	set header
.hl	M	draw horizontal line
\n(hm	R	header margin
.hx	M	suppress next page's headers/footers
.hy	M§	set automatic hyphenation mode
.i	M	italicize argument
.ie	M§	conditional with **.el**
.if	M§	conditional
\n(ii	R	indented paragraph indentation
.in	M§	set indentation (transient; cf. **.ba**)
.ip	M	begin indented paragraph
.ix	M	set indentation without break
\l'*d*'	F§	draw horizontal line of length *d*
.lc	M§	set leader repetition character
.ld	M□	reset localization and date info
.ll	M	set line length in *me* environments
.lp	M	begin fully left-aligned paragraph
*(lq	S	left double quotation mark
.ls	M§	set multi-line spacing
.m1	M	set page top to header distance
.m2	M	set header to text distance
.m3	M	set text to footer distance
.m4	M	set footer to page bottom distance
.mc	M§	set margin character
.mk	M§	mark vertical position (see **.rt**)
\n(mo	R§	month of the year
*(mo	S	month name
\n*x*	F§	interpolate register *x*
\n(*xx*	F§	interpolate register *xx*
.n1	M	begin output line numbering
.n2	M	end or alter output line numbering
.na	M§	disable adjustment
.ne	M§	need vertical space
.nf	M§	disable filling
.nh	M§	disable automatic hyphenation

Name	Type	Description
\n(no	R□	line numbering offset
.np	M	begin numbered paragraph
.nr	M§	set register
.ns	M§	begin no-space mode
.of	M	set odd-numbered page footer
.oh	M	set odd-numbered page header
.pa	M	begin page
.pd	M	output delayed text
\n(pf	R	paragraph font
\n(pi	R	paragraph indentation
.pl	M§	set page length
.pn	M§	set next page number
.po	M§	set page offset
\n(po	R	paragraph offset
.pp	M	begin first-line indented paragraph
\n(pp	R	paragraph type size in points
\n(ps	R	paragraph prespace
.q	M	quote argument
\n(qi	R	long quotation left/right indentation
\n(qp	R	long quotation type size in points
\n(qs	R	long quotation pre/post space
.r	M	set argument in roman
.re	M	reset tab stops
.rj	M□	right-align output lines
.rm	M§	remove macro, string, or request
.rn	M§	rename macro, string, or request
.ro	M	use Roman numerals for page numbers
*(rq	S	right double quotation mark
.rr	M§	remove register
.rs	M§	restore spacing (end no-space mode)
.rt	M§	return to vertical position set by **.mk**
\s'±*n*'	F§°	inline type size change to/by *n* points
\n(sf	R	section title font
.sh	M	begin numbered section
\n(si	R	section indentation per level of depth
.sk	M	skip next page
.sm	M	set argument at smaller type size
.so	M§	source (interpolate) input file
\n(so	R	additional section title offset
.sp	M§	insert space (except at page top; cf. **.bl**)
\n(sp	R	section title type size in points
\n(ss	R	section prespace
.sx	M	change section depth
\n(sx	R□	super/subscript line height increase
.sz	M	set type size and vertical spacing
.ta	M§	set tab stops
.tc	M§	set tab repetition character
*(td	S	today's date
\n(tf	R	title font
.ti	M§	temporary indentation (next line only)
.tl	M§	output three-part title
\n(tm	R	distance from page top to text area
.tp	M	begin title page
\n(tp	R	title type size in points
.tr	M§	translate characters
\n(tv	R□	vertical spacing of text (%)
.u	M	underline argument
.uh	M	begin unnumbered section
.ul	M§	underline output lines
\v'*d*'	F§	move distance *d* vertically
\w'*S*'	F§	interpolate formatted width of *S*
*(wa	S□	term for "appendix" used by **.$c**
*(wc	S□	term for "chapter" used by **.$c**
.xl	M	set line length (current environment)
.xp	M	output index
\n(xs	R	index entry prespace
\n(xu	R	index undent (right indentation)
\n(y2	R□	year of the century
\n(y4	R□	year
\n(yr	R§	year minus 1900
\n(zs	R	floating keep pre/post space
\{	F§	begin conditional group
*{	S	begin superscripting
\\|	F§	1/6 em unbreakable thin space
\}	F§	end conditional group
*}	S	end superscripting
\~	F§□	unbreakable word space