me Macro Package Reference Manual

*for groff me version 1.23.0.rc1.1628-515a8**

Eric P. Allman

Project INGRES
Electronics Research Laboratory
University of California, Berkeley
Berkeley, California 94720

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; fonts; type sizes; the use and definition of registers, strings, and macros; and units of measurement like ens, points, and vees. For a more casual introduction to text processing using *groff*, refer to the document "Writing Papers with *groff* using −me".

Many of the package's rendering parameters can be adjusted through registers, strings, and sometimes macro calls. Default parameter values are shown in square brackets in this document. Fonts may be set to a font mounting position only. Font zero tells *me* to perform no font change; the font of the preceding text is used instead. Notice that font 0 is a "pseudo-font"; that is, it is interpreted and handled by the macros. This means that although it is valid to set a font register to zero, doing so may produce unexpected results in a font style escape sequence such as **\f0**, because those are interpeted directly by *troff*.

Changes to parameters that have a global effect on the format of the page (notably page length and header and footer margins) should be done before calling any paragraphing or sectioning macros. After that, such changes are not well defined and should be avoided.

All distances are in device-specific basic units, so it is nearly always necessary to use a scaling unit. For example, the request to set the paragraph indent to eight ens is

```
.nr pi 8n
```

and not ".nr pi 8", which would make the paragraph indent eight basic units, or 8/72,000 inches on *grops, groff*'s PostScript output driver, an almost imperceptible thousandth of an inch.

Registers and strings with names of the form **$*x*** may be used in expressions but should not be changed. Macros with names of the form **$*x*** 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.

Objects in *me* follow a rigid naming convention. The user may define registers, strings, and macros, provided that s/he uses single-character uppercase names or double-character names consisting of letters and digits, with at least one uppercase letter. In no case should special characters be used in user-defined names. Locally defined macros should all be of the form **.*** *X*, where *X* is any letter (upper or lowercase) or digit.

---

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

1. Paragraphing

.lp — Begin left-aligned paragraph. Centering and underlining are turned off if they were on, the font is set to **\n(pf** [1], the type size to **\n(pp** [10p], and vertical space of **\n(ps** is inserted before the paragraph [0.35v]. 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 lines of the paragraph are kept together on a page.

.pp — Like **.lp**, but apply an indentation of **\n(pi** [5n] to the first line only.

.ip *T I* — Begin indented paragraph with hanging tag *T*. The paragraph text is indented by *I* (if not specified, then by **\n(ii** [5n]) more than an **.lp** paragraph. The tag *T* is "exdented" (the opposite of *in*dented). Any spaces in *T* must be unbreakable. If *T* will not fit in the space *I,* **.ip** will start the paragraph text on a new line.

.np — Like **.ip**, but automatically number the paragraph, using the number (in parentheses) as the tag. Numbering is reset after an **.lp**, **.pp**, **.bu**, **.sh**, or **.sx** call. The current paragraph number is in **\n($p**.

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

2. Section Headings

Numbered sections are similar to paragraphs except that a section number of the form "1.2.3" is automatically generated for each. The *depth* of the section is the count of numbers (separated by decimal points) in the section number. Unnumbered section headings are similar, except that no number is attached to the heading.

.sh ±*N T a b c d e f* — Begin numbered section of depth *N* with optional title *T.* *N* can be signed, which changes the current depth by ±*N.* If *N* is unspecified (or an an empty string if further arguments are present) the current depth, stored in the register **\n($0**, is used. The components of the section number are maintained in **\n($1** through **\n($6**; combined, they are available in ***($n**. Vertical space of **\n(ss** [1v] precedes the heading. *T* in is rendered in font **\n(sf** [8] at size **\n(sp** [10p]. If **\n(si** is non-zero, the base indentation is set to **\n(si** times the section depth, and the section title is exdented (see **.ba**). Also, an additional indentation of **\n(so** [0] is added to the section title (but not to the body of the section). The font is then set to the paragraph font **\n(pf** so that more information may occur on the line with the section number and title. **.sh** insures 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 *a* through *f* are specified, each component of the section number is assigned the corresponding argument rather than incremented automatically. A hyphen ("−") for a component argument prevents its alteration. If *T* is a single underscore ("_"), the section depth and numbering are reset, but the base indent is not, and nothing is output—this is useful to automatically coordinate section numbers with chapter numbers.

.sx ±*N* — Go to section depth ±*N* [−1], but do not print the number and title, and do not increment the section number at level *N.* This has the effect of starting a new paragraph at level *N.*

.uh *T* — Begin unnumbered section heading. Like **.sh**, ignoring the section numbering features.

.$p *T B N* — Output a section heading. *T* is the title, *B* is the number, and *N* is the depth. These parameters are not always present; in particular, **.sh** passes all three, **.uh** passes only the first, and **.sx** passes three, but the first two are empty strings. Care should be taken when redefining this macro; it is complex and subtle.

.$0 *T B N* — This *hook* macro, normally undefined, is called automatically by **.sh** and **.uh** after they call **.$p**, and is passed the same arguments. **.$0** is normally undefined; you can define it to, for instance, automatically put every section title into a table of contents.

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

3. Headers and Footers

Headers and footers known as "three-part titles" for their left, middle, and right-hand components *l, m,* and *r,* are put at the top and bottom of every page automatically. They are set in font **\n(tf** [3] at size **\n(tp** [10p]. Each of the definitions apply as of the *next* page. Three-part titles must be quoted if there are two blanks adjacent anywhere in the title or more than eight blanks total.

The spacing of headers and footers is controlled by four registers. **\n(hm** [4v] is the distance from the top of the page to the top of the header, **\n(fm** [3v] is the distance from the bottom of the page to the bottom of the footer, **\n(tm** [7v] is the distance from the top of the page to the top of the text, and **\n(bm** [6v] is the distance from the bottom of the page to the bottom of the text (nominal). The macros **.m1**, **.m2**, **.m3**, and **.m4** are also supplied for compatibility with old *roff* documents.

.he '*l*'*m*'*r*' Define header to be printed at the top of every page.

.fo '*l*'*m*'*r*' Define footer to be printed at the bottom of every page.

.eh '*l*'*m*'*r*' Define header to be printed at the top of every even-numbered page.

.oh '*l*'*m*'*r*' Define header to be printed at the top of every odd-numbered page.

.ef '*l*'*m*'*r*' Define footer to be printed at the bottom of every even-numbered page.

.of '*l*'*m*'*r*' Define footer to be printed at the bottom of every odd-numbered page.

.hx Suppress headers and footers on the next page.

.m1 +*N* Set the space between the top of the page and the header [4v].

.m2 +*N* Set the space between the header and the first line of text [2v].

.m3 +*N* Set the space between the bottom of the text and the footer [2v].

.m4 +*N* Set the space between the footer and the bottom of the page [4v].

.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** or the end of input.

.$h Called at every page to print the header. May be redefined to provide fancy (e.g., multi-line) headers, but doing so loses the function of the **.he**, **.fo**, **.eh**, **.oh**, **.ef**, and **.of** macros, as well as the chapter-style title feature of **.+c**.

.$f Print footer; same comments apply as in **.$h**.

.$H A normally undefined macro which is called at the top of each page (after putting out the header, initial saved floating keeps, etc.), or of each column when in two-column mode; in other words, this macro is called immediately before printing text on a page or in a column. It can be used for column headings and the like.

4. Displays

All displays except centered blocks and block quotes are preceded and followed by an extra **\n(bs** [same as **\n(ps**] space. Quote spacing is stored in a separate register; centered blocks have no default initial or trailing space. The vertical spacing of all displays except quotes and centered blocks is stored in register **\n($V** instead of **\n($v**.

.(l *m f* Begin list. Lists are single-spaced text with filling disabled. If *f* is **F**, filling is enabled. If *m* [**I**] is **I** the list is indented by **\n(bi** [4m]; if **M** the list is indented to the left margin; if **L** the list is left-aligned with respect to the text (different from **M** only if the base indent (stored in **\n($i** and set with **.ba**) is not zero); and if **C** the list is centered on a line-by-line basis. The list is set in font **\n(df** [0]. Must be matched by a **.)l**. This macro is almost like **.(b** except that no attempt is made to keep the display on one page.

.)l End list.

.(q Begin major quote. These are single spaced, filled, moved in from the text on both sides by **\n(qi** [4n], preceded and followed by **\n(qs** [same as **\n(bs**] space, and are set at type size **\n(qp** [one point smaller than surrounding text].

.)q End major quote.

.(b *m f* Begin block. Blocks are a form of *keep*, where the text of a keep is kept together on one page if possible (keeps are useful for tables and figures which should not be broken over a page). If the block will not fit on the current page a new page is begun, *unless* that would leave more than **\n(bt** [0] white space at the bottom of the text. If **\n(bt** is zero, the threshold feature is turned off. Blocks are not filled unless *f* is **F**, when they are filled. The block will be left-aligned if *m* is **L**, indented by **\n(bi** [4m] if *m* is **I** or absent, centered (line-for-line) if *m* is **C**, and aligned to the left margin (not to the base indent) if *m* is **M**. The block is set in font **\n(df** [0].

.)b End block.

.(z *m f* Begin floating keep. Like **.(b** except that the keep is *floated* to the bottom of the page or the top of the next page. Therefore, its position relative to the text changes. The floating keep is preceded and followed by **\n(zs** [1v] space. Also, it defaults to mode **M**.

.)z End floating keep.

.(c Begin centered block. The next keep is centered as a block, rather than on a line-by-line basis as with **.(b C**. This call may be nested inside keeps.

.)c End centered block.

5. Annotations

.(d Begin delayed text. Everything in the next keep is saved for output later with **.pd**, in a manner similar to footnotes.

.)d *n* End delayed text. The delayed text register **\n($d** and the associated string ***#** are incremented if ***#** has been referenced.

.pd Print delayed text. Everything diverted via **.(d** is printed and truncated. This might be used at the end of each chapter.

.(f Begin footnote. The text of the footnote is floated to the bottom of the page and set in font **\n(ff** [1] and size **\n(fp** [8p]. Each entry is preceded by **\n(fs** [0.2v] space, is indented **\n(fi** [3n] on the first line, and is indented **\n(fu** [0] from the right margin. Footnotes line up underneath two-column output. If the text of the footnote will not all fit on one page it will be carried over to the next page.

.)f *n* End footnote. The register **\n($f** and the associated string ****** are incremented if they have been referenced.

.$s Draw a 2i horizontal line. This macro is called by **.(f** to output the footnote separator, and may be redefined to produce another separator (no more than one vee in height).

.(x *x* Begin index entry. Index entries are saved in the index *x* [**x**] until called up with **.xp.** Each entry is preceded by a **\n(xs** [0.2v] space. Each entry is “undented” by **\n(xu** [0.5i]; this register tells how far the page number extends into the right margin.

.)x *P A* End index entry. The index entry is rendered with a *leader* (row of dots) and *A* [null] right-aligned on the last line (such as for an author’s name), followed by P [**\n%**]. If *A* is specified, *P* must be as well; **\n%** can be used to print the current page number. If *P* is an underscore, no page number or leader are printed.

.xp *x* Print index *x* [**x**]. The index is formatted in the font, size, and so forth in effect at the time it is printed, rather than at the time it is collected.

6. Columnated Output

.2c +*S N*	Enter two-column mode. The column separation is set to +*S* [4n, 0.5i in ACM mode] (saved in **\n($s**). The column width, calculated to fill the single-column line length with both columns, is stored in **\n($l**. The current column is in **\n($c**. You can test register **\n($m** [1] to see if you are in single-column or double-column mode. Actually, the macro enters *N*- [2-]column output.
.1c	Revert to single-column mode.
.bc	Begin column. This is like **.bp** except that it begins a new column on a new page only if necessary, rather than forcing a whole new page if there is another column left on the current page.

7. Type Size and Font Styles

.sz +*P*	Set the type size to *P* [10p], and the line spacing proportionally. The line spacing as a percentage of the type size is stored in **\n($v** [120]. The percentage used internally by displays and annotations is stored in **\n($V** [120] (although this is not used by **.sz**). This size is *not* sticky beyond many macros: in particular, **\n(pp** (paragraph point size) modifies the type size every time a new paragraph is begun using the **.pp**, **.lp**, **.ip**, **.np**, or **.bu** macros. Also, **\n(fp** (footnote point size), **\n(qp** (quote point size), **\n(sp** (section header point size), and **\n(tp** (title point size) may modify the type size.

The remaining macros in this section style or decorate an argument *W* and accept an optional second argument *X*, which is set in the previous style without intervening space. To concatenate several font styles within a word, 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.
.u *W X*	Underline *W*. This is a true underlining, as opposed to the **.ul** request, which changes to the "underline font" (usually italics in *groff*). It won't work right if *W* is adjusted or broken (including hyphenation breaks); it is reliable only when filling is disabled.
.q *W X*	Quote *W*. In *groff* this surrounds *W* with **"** and **"**.
.bx *W X*	Set *W* in a box. It won't work right if *W* is adjusted or broken (including hyphenation breaks); it is reliable only when filling is disabled.
.sm *W X*	Set *W* one point smaller than the surrounding type.

8. *roff* Support

.ix +*N*	Indent, no break. Equivalent to **'in** *N*.
.bl *N*	Leave *N* contiguous white space, on the next page if not enough room on this page. Equivalent to putting a **.sp** *N* inside a block.
.pa +*N*	Equivalent to **.bp**.
.ro	Set page number using Roman numerals. Equivalent to **.af % i**.
.ar	Set page number using Arabic numerals. Equivalent to **.af % 1**.
.n1	Number lines from one, in margin, on each page.
.n2 *N c*	Number lines from *N*. Stop numbering if *N* missing; resume where stopped if *N* unsigned, or resume with number incremented by ±*N*. If *c* is **c**, maintain compatibility with original **.n2** by narrowing line width to accommodate numbers.

.sk Leave the next output page blank, except for headers and footers. This is used to leave space for a full-page diagram which is produced externally and pasted in later. To get a partial-page paste-in display, say **.sv** *N*, where *N* is the amount of space to leave; this space will be output immediately if there is room, and will otherwise be output at the top of the next page. However, be warned: if *N* is greater than the amount of available space on an empty page, no space will ever be output.

9. Preprocessor Support

.EQ *m T* Begin equation. The equation is centered if *m* is **C** or omitted, indented **\n(bi** [4m] if *m* is **I**, and left-aligned if *m* is **L**. *T* 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 equation. If *c* is **C** the equation must be continued by immediately following it with another **.EQ**, the text of which can be centered along with this one. Otherwise, the equation is printed, always on one page, with **\n(es** [0.5v] space above and below it.

.TS *h* Table start. Tables are single spaced and kept on one page if possible. If you have a large table which will not fit on one page, specify **H** for *h* and follow the table header (to be printed on every page of the table) with a **.TH** call. See the *tbl*(1) man page.

.TH With **.TS H**, end the table header.

.TE Table end. This table does not float; in fact, it is not even guaranteed to stay on one page if you use requests such as **.sp** intermixed with the text of the table. If you want it to float (or if you use requests inside the table), surround the entire table (including the **.TS** and **.TE** macros) with the macros **.(z** and **.)z**.

.PS *h w* Begin *pic* picture. *h* is the height and *w* is the width, both in basic units.

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

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

.IS Begin *ideal* picture.

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

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

.GS *x* Begin *grn* picture. *x* can be either **C**, **L**, or **R** to center, left-, or right-align the whole picture. The default is **C**.

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

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

10. Miscellaneous

.re Reset tabs every 0.5i.

.ba +*N* Set the base indentation to +*N* [0] (saved in **\n($i**). All paragraphs, sections, and displays come out indented by this amount. Titles and footnotes are unaffected. The **.sh** macro performs a **.ba** if **\n(si** [0] is not zero, and sets the base indent to **\n(si*\n($0**.

.xl +*N* Set the line length to *N* [6.0i]. This differs from **.ll** because it only affects the current environment.

.ll +*N* Set line length in all environments used by *me* to *N* [6.0i].[1] This should not be used after output has begun, and particularly not in two-column output. The current line length is

[1] *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.

stored in **\n($l**.

.hl Break and draw a horizontal line of the current line length. This is useful inside floating keeps to differentiate between the text and the figure.

11. Standard Papers

.tp Begin title page. Spacing at the top of the page can occur, and headers and footers are suppressed. Also, the page number is not incremented for this page.

.++ *m H* This macro defines the section of the paper which we are entering. The section type is defined by *m*. **C** means that we are entering the chapter portion of the paper, **A** means that we are entering the appendix portion of the paper, **P** means that the material following should be the preliminary portion (abstract, table of contents, etc.) portion of the paper, **AB** means that we are entering the abstract (numbered independently from 1 in Arabic numerals), and **B** means that we are entering the bibliographic portion at the end of the paper. Also, the variants **RC** and **RA** are allowed, which specify renumbering of pages from one at the beginning of each chapter or appendix, respectively. The section type is available in register **\n(_M** [1]; value 1 is equivalent to type **C** or **RC**, value 2 represents type **A** or **RA**, and values 3 to 5 are type **P**, **B**, and **AB**, respectively. The *H* parameter defines the new header. If there are any spaces in it, the entire header must be quoted. If you want the header to have the chapter number in it, use **\\\\n(ch**. For example, to number appendices **A.1** etc., type **.++ RA '""\\\\n(ch.%'**. Each section (chapter, appendix, etc.) should be preceded by the **.+c** macro. It is easier when using *troff* to put the front material at the end of the paper, so that the table of contents can be collected and put out; this material can then be physically moved to the beginning of the paper.

.+c *T* Begin chapter with title *T*. The chapter number is maintained in **\n(ch**. This register is incremented every time **.+c** is called with a parameter. The title and chapter number are printed by **.$c**. The header is moved to the footer on the first page of each chapter. If *T* is omitted, **.$c** is not called; this is useful for doing your own "title page" at the beginning of papers without a title page proper. **.$c** calls **.$C** as a hook so that chapter titles can be inserted into a table of contents automatically. The footnote numbering is reset to one.

.$c** *T* Print chapter number (from **\n(ch**) and *T*. This macro can be redefined to your liking. It is defined by default to be acceptable for a PhD thesis at Berkeley. This macro calls **$C, which can be defined to make index entries, or whatever.

.$C *K N T* This macro is called by **.$c**. It is normally undefined, but can be used to automatically insert index entries, or whatever. *K* is a keyword, either "Chapter" or "Appendix" (depending on the **.++** mode); *N* is the chapter or appendix number, and *T* is the chapter or appendix title.

12. Predefined Strings

****** Footnote number bracketed by superscripting strings: ***[\n($f*]**. The register in this string is incremented after each call to **.)f**.

***#** Delayed text number surrounded by square brackets: **[\n($d]**. The register in this string is incremented after each call to **.)d**.

***{** Superscript. This string produces an upward movement and a change to a smaller type size. Extra space is left above the line to allow room for the superscript. In *nroff* mode, it produces a left square bracket.

***}** Counterpart of ***{**. In *nroff* mode, it produces a right square bracket. For example, to produce a superscript you might type **x*{2*}**, which will render as $\mathbf{x^2}$ on typesetters and as **x[2]** on terminals.

***<** Subscript. Extra space is left below the line to allow for the subscript. In *nroff* mode, it produces a left angle bracket.

***>** Counterpart of ***<**. In *nroff* mode, it produces a right angle bracket.

***(dw** The weekday name.

***(mo** The month name.

***(td** Today's date, of the form "February 15, 2003" (not in quotation marks). Other date formats can be obtained by using **\n(dy** (the day of the month; for example, 15), ***(mo** (as noted above) or **\n(mo** (the same, but as an ordinal number; for example, February is 2), **\n(y4** (the current year), and **\n(y2** (the last two digits of the current year).

13. Special Characters and Marks

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 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 4.3BSD and 4.4BSD Unix systems. (The last lacks even that virtue, since Version 7 Unix *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 and Marks” 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 identifer: 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; the name’s type (one of the foregoing); and its description. The types 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 can carry a variety of annotations. A section sign § indicates features interpreted by *troff* rather than the *me* package. The dagger † and double dagger ‡ indicate user-definable objects; those marked with ‡ are defined by default and changing them may have unexpected side effects. The degree sign ° notes a feature introduced in AT&T device-independent *troff*, and a square □ marks *groff* extensions.

Name	Type	Description
\space	F§	unbreakable, unadjustable space
\"	F§	comment (to end of line)
*#	S	optional delayed text tag string
\$N	F§	interpolate argument N
\n($0	R	section depth
.$0	M†	invoked after section title printed
\n($1	R	first section number
.$1	M†	invoked before printing depth 1 section
\n($2	R	second section number
.$2	M†	invoked before printing depth 2 section
\n($3	R	third section number
.$3	M†	invoked before printing depth 3 section
\n($4	R	fourth section number
.$4	M†	invoked before printing depth 4 section
\n($5	R	fifth section number
.$5	M†	invoked before printing depth 5 section
\n($6	R	sixth section number
.$6	M†	invoked before printing depth 6 section
.$C	M†	called at beginning of chapter
.$H	M†	text header
\n($V	R‡	relative vertical spacing in displays
\n($c	R	current column number
.$c	M‡	print chapter title
\n($d	R	delayed text number
\n($f	R	footnote number
.$f	M‡	print footer
.$h	M‡	print header
\n($i	R	paragraph base indent
\n($l	R	column width
\n($m	R	number of columns in effect
*($n	S	concatenated section number
\n($p	R	numbered paragraph number
.$p	M‡	print section heading (internal macro)
\n($s	R	column indent
.$s	M‡	footnote separator (from text)
\n($v	R‡	relative vertical spacing in text

Name	**Type**	**Description**
\%	F§	control hyphenation
\n%	R§	current page number
\&	F§	non-printing input break
\(*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 quote
.(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 quote
.)x	M	end index entry
.)z	M	end floating keep
**x*	F§	interpolate string *x*
*(*xx*	F§	interpolate string *xx*
**	S	optional footnote tag string
.++	M	set paper section type
.+c	M	begin chapter
\−	F§	minus sign
*−	S	em dash
\n(.$	R§	number of parameters supplied to macro
\n(.i	R§	current indent
\n(.l	R§	current line length
\n(.s	R§	current type size in points
\0	F§	unadjustable digit-width space
.1c	M	revert to single-column output
.2c	M	begin two-column output
*<	S	begin subscript
*>	S	end subscript
.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

Name	**Type**	**Description**
\^	F§	1/12 em unbreakable hair space
_	F§	underscore
\n(_M	R	section type (as set with **.++**)
\`	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 indent
.bc	M	begin new column
.bi	M	embolden and italicize argument
\n(bi	R	display (block) indent
.bl	M	blank line (even at page top; contrast with **.sp**)
\n(bm	R	bottom title margin
.bp	M§	begin new page
.br	M§	break (begin new line)
\n(bs	R	display (block) pre/post spacing
\n(bt	R	block keep threshold
.bx	M	box argument
\c	F§	continue input
.ce	M§	center lines
\n(ch	R	current chapter number
.de	M§	define macro
\n(df	R	display font
.ds	M§	define string
\n(dw	R§	current day of week
*(dw	S	current day of week
\n(dy	R§	day of month
\e	F§	printable version of \
.ef	M	set footer (even-numbered pages only)
.eh	M	set header (even-numbered pages only)
.el	M§	else part of 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 indent (first line only)
\n(fm	R	footer margin
.fo	M	set footer
\n(fp	R	footnote type size in points
\n(fs	R	footnote prespace
\n(fu	R	footnote undent (from right margin)
\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

Name	Type	Description
.hx	M	suppress headers and footers on next page
.hy	M§	set automatic hyphenation mode
.i	M	italicize argument
.ie	M§	conditional with else
.if	M§	conditional
\n(ii	R	indented paragraph indent
.in	M§	indent (transient; use **.ba** for pervasive)
.ip	M	begin indented paragraph
.ix	M	indent, no break
\l'*d*'	F§	draw horizontal line of length *d*
.lc	M§	set leader repetition character
.ll	M	set line length
.lp	M	begin left-aligned paragraph
*(lq	S	left double quotation mark
.ls	M§	set multi-line spacing
.m1	M	set space from top of page to header
.m2	M	set space from header to text
.m3	M	set space from text to footer
.m4	M	set space from footer to bottom of page
.mc	M§	insert margin character
.mk	M§	mark vertical position
\n(mo	R§	month of year
*(mo	S	current month
\n*x*	F§	interpolate register *x*
\n(*xx*	F§	interpolate register *xx*
.n1	M	number lines in margin
.n2	M	number lines in margin
.na	M§	turn off text adjustment
.ne	M§	need vertical space
.nf	M§	disable filling
.nh	M§	turn off automatic hyphenation
.np	M	begin numbered paragraph
.nr	M§	set register
.ns	M§	no space mode
.of	M	set footer (odd-numbered pages only)
.oh	M	set header (odd-numbered pages only)
.pa	M	begin page
.pd	M	print delayed text
\n(pf	R	paragraph font
\n(pi	R	paragraph indent
.pl	M§	set page length
.pn	M§	set next page number
.po	M§	page offset
\n(po	R	simulated page offset
.pp	M	begin paragraph
\n(pp	R	paragraph type size in points
\n(ps	R	paragraph prespace
.q	M	quote argument
\n(qi	R	quote indent (also shortens line)
\n(qp	R	quote type size in points
\n(qs	R	quote pre/post space
.r	M	set argument in roman style

Name	**Type**	**Description**
.rb	M	real bold font
.re	M	reset tabs
.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
.rt	M§	return to vertical position
\s'±*S*'	F§°	inline type size change to (±: by) *S* points
\n(sf	R	section title font
.sh	M	begin numbered section
\n(si	R	relative base indent per section depth
.sk	M	skip next page
.sm	M	set argument at a smaller type size
.so	M§	source input file
\n(so	R	additional section title offset
.sp	M§	vertical space (except at page top; contrast with **.bl**)
\n(sp	R	section title type size in points
\n(ss	R	section prespace
.sx	M	change section depth
.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 indent (next line only)
.tl	M§	three part title
\n(tm	R	top title margin
.tp	M	begin title page
\n(tp	R	title type size
.tr	M§	translate characters
.u	M	underline argument
.uh	M	unnumbered section
.ul	M§	underline next line
\v'*d*'	F§	move distance *d* vertically
\w'*S*'	F§	interpolate width of string *S*
.xl	M	set line length (local)
.xp	M	print index
\n(xs	R	index entry prespace
\n(xu	R	index undent (from right margin)
\n(y2	R□	year (last two digits only)
\n(y4	R□	year (all digits)
\n(yr	R§	year minus 1900
\n(zs	R	floating keep pre/post space
\{	F§	begin conditional group
*{	S	begin superscript
\\|	F§	1/6 em unbreakable thin space
\}	F§	end conditional group
*}	S	end superscript
\~	F§□	unbreakable space