–ME REFERENCE MANUAL

GROFF Version 1.20.1†

*Eric P. Allman**

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

Modified for GROFF by James Clark

This document describes in extremely terse form the features of the **–me** macro package for GROFF. Some familiarity is assumed with GROFF. Specifically, the reader should understand breaks, fonts, pointsizes, the use and definition of number registers and strings, how to define macros, and scaling factors for ens, points, **v**'s (vertical line spaces), etc.

For a more casual introduction to text processing using GROFF, refer to the document *Writing Papers with GROFF using –me.*

There are a number of macro parameters that may be adjusted. Fonts may be set to a font number only. Font 0 is no font change; the font of the surrounding text is used instead. Notice that font 0 is a "pseudo-font"; that is, it is simulated by the macros. This means that although it is valid to set a font register to zero, it is not valid to use the escape character form, such as:

```
\f0
```

All distances are in basic units, so it is nearly always necessary to use a scaling factor. For example, the request to set the paragraph indent to eight one-en spaces is:

```
.nr pi 8n
```

and not

```
.nr pi 8
```

which would set the paragraph indent to eight basic units, or about 0.02 inch. Default parameter values are given in brackets in the remainder of this document.

Registers and strings of the form **$***x* may be used in expressions but should not be changed. Macros of the form **$***x* perform some function (as described) and may be redefined to change this function. This may be a sensitive operation; look at the body of the original macro before changing it.

All names in –me follow a rigid naming convention. The user may define number registers, strings, and macros, provided that s/he uses single character upper case names or double character names consisting of letters

†Based on Berkeley Release 2.31.

*Author's current address: Britton Lee, Inc., 1919 Addison Suite 105, Berkeley, California 94704.

and digits, with at least one upper case 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 lower case) or digit.

This documentation applies to GROFF version 1.20.1 of the −me macros.

1. Paragraphing

These macros are used to begin paragraphs. The standard paragraph macro is **.pp**; the others are all variants to be used for special purposes.

After the first call to one of the paragraphing macros defined in this section or the **.sh** macro (defined in the next session), the effects of changing parameters which will have a global effect on the format of the page (notably page length and header and footer margins) are not well defined and should be avoided.

.lp
Begin left-justified paragraph. Centering and underlining are turned off if they were on, the font is set to **\n(pf** [1] the type size is set to **\n(pp** [10p], and a **\n(ps** space is inserted before the paragraph [0.35v] The indent 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**, except that it puts **\n(pi** [5n] units of indent. This is the standard paragraph macro.

.ip *T I*
Indented paragraph with hanging tag. The body of the following paragraph is indented *I* spaces (or **\n(ii** [5n] spaces if *I* is not specified) more than a non-indented paragraph (such as with **.pp**) is. The title *T* is exdented (opposite of indented). The result is a paragraph with an even left edge and *T* printed in the margin. Any spaces in *T* must be unpaddable. If *T* will not fit in the space provided, **.ip** will start a new line.

.np
A variant of .ip which numbers paragraphs. Numbering is reset after a **.lp**, **.pp**, or **.sh**. The current paragraph number is in **\n($p**.

.bu
Like **.np** except that paragraphs are marked with bullets (•). Leading space is eliminated to create compact lists.

2. Section Headings

Numbered sections are similar to paragraphs except that a section number is automatically generated for each one. The section numbers are of the form **1.2.3**. 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*. If *N* is missing the current depth (maintained in the number register **\n($0**) is used. The values of the individual parts of the section number are maintained in **\n($1** through **\n($6**. There is a **\n(ss** [1v] space before the section. *T* is printed as a section title in font **\n(sf** [8] and size **\n(sp** [10p]. The “name” of the section may be accessed via ***($n**. If **\n(si** is non-zero, the base indent is set to **\n(si** times the section depth, and the section title is exdented. (See **.ba**.) Also, an additional indent 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, so that more information may occur on the line with the section number and title. **.sh** insures that there is enough room to print the section head plus the beginning of a paragraph (about 3 lines total). If *a* through *f* are specified, the section number is set to that number rather than incremented automatically. If any of *a* through *f* are a hyphen that number is not reset. If *T* is a single underscore (“_”) then the section depth and numbering is reset, but the base indent is not reset and nothing is printed out. 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*
: Unnumbered section heading. The title *T* is printed with the same rules for spacing, font, etc., as for **.sh**.

.$p *T B N*
: Print section heading. May be redefined to get fancier headings. *T* is the title passed on the **.sh** or **.uh** line; *B* is the section number for this section, and *N* is the depth of this section. 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 null strings. Care should be taken if this macro is redefined; it is quite complex and subtle.

.$0 *T B N*
: This macro is called automatically after every call to **.$p**. It is normally undefined, but may be used to automatically put every section title into the table of contents or for some similar function. *T* is the section title for the section title which was just printed, *B* is the section number, and *N* is the section depth.

.$1 – .$6
: Traps called just before printing that depth section. May be defined to (for example) give variable spacing before sections. These macros are called from **.$p**, so if you redefine that macro you may lose this feature.

3. Headers and Footers

Headers and footers are put at the top and bottom of every page automatically. They are set in font **\n(tf** [3] and 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 are controlled by three number 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 ROFF documents.

.he *´l´m´r´*
: Define three-part header, to be printed on 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 every 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., multiline) headers, but doing so loses the function of the **.he**, **.fo**, **.eh**, **.oh**, **.ef**, and **.of** requests, 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.); in other words, this macro is called immediately before printing text on a page. 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, unfilled text. If *f* is **F**, the list will be filled. 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 justified 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 in point 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-justified if *m* is **L**, indented by **\n(bi** [4m] if *m* is **I** or absent, centered (line-for-line) if *m* is **C**, and left justified to the 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 number register **\n($d** and the associated string ***#** are incremented if ***#** has been referenced.
.pd	Print delayed text. Everything diverted via **.(d** is printed and truncated. This might be used at the end of each chapter.
.(f	Begin footnote. The text of the footnote is floated to the bottom of the page and set in font **\n(ff** [1] and size **\n(fp** [8p]. Each entry is preceded by **\n(fs** [0.2v] space, is indented **\n(fi** [3n] on the first line, and is indented **\n(fu** [0] from the right margin. Footnotes line up underneath two column output. If the text of the footnote will not all fit on one page it will be carried over to the next page.
.)f *n*	End footnote. The number register **\n($f** and the associated string ****** are incremented if they have been referenced.

.$s The macro to output the footnote separator. This macro may be redefined to give other size lines or other types of separators. Currently it draws a 1.5i line.

.(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 finished with a row of dots with *A* [null] right justified on the last line (such as for an author’s name), followed by P [**\n%**]. If *A* is specified, *P* must be specified; **\n%** can be used to print the current page number. If *P* is an underscore, no page number and no row of dots 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. Columned 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 request 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. Fonts and Sizes

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

.r *W X* Set *W* in roman font, appending *X* in the previous font. To append different font requests, use *X* = **\c**. If no parameters, change to roman font.

.i *W X* Set *W* in italics, appending *X* in the previous font. If no parameters, change to italic font.

.b *W X* Set *W* in bold font and append *X* in the previous font. If no parameters, switch to bold font.

.u *W X* Underline *W* and append *X*. This is a true underlining, as opposed to the **.ul** request, which changes to “underline font” (usually italics in GROFF). It won’t work right if *W* is spread or broken (including hyphenated). In other words, it is safe in nofill mode only.

.q *W X* Quote *W* and append *X*. In GROFF this surrounds *W* with **“**, and **”**.

.bi *W X* Set *W* in bold italics and append *X*.

.bx *W X* Sets *W* in a box, with *X* appended. It won’t work right if *W* is spread or broken (including hyphenated). In other words, it is safe in nofill mode only.

.sm *W X* Sets *W* in a smaller pointsize, with *X* appended.

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 a **.sp** *N* inside a block.
.pa +*N*	Equivalent to **.bp**.
.ro	Set page number in roman numerals. Equivalent to **.af % i**.
.ar	Set page number in Arabic. Equivalent to **.af % 1**.
.n1	Number lines in margin from one on each page.
.n2 *N*	Number lines from *N*, stop if *N* = 0.
.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 justified if *m* is **L**. *T* is a title printed on 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 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, use *h* = **H** and follow the header part (to be printed on every page of the table) with a **.TH**. See *Tbl – A Program to Format Tables* by M. E. Lesk.
.TH	With **.TS H**, ends the header portion of the table.
.TE	Table end. Note that 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** requests) with the requests **.(z** and **.)z**.
.PS *h w*	Begin *pic* picture. *H* is the height and *w* is the width, both in basic units.
.PE	End picture.
.IS	Begin *ideal* picture.
.IE	End *ideal* picture.
.IF	End *ideal* picture (alternate form).
.GS *x*	Begin *gremlin* picture. *X* can be either **C**, **L**, or **R** to center, left, or right justify the whole picture. Default is centering the image.
.GE	End *gremlin* picture.
.GF	End *gremlin* picture (alternate form).

10. Miscellaneous

.re — Reset tabs every 0.5i.

.ba *+N* — Set the base indent 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** request performs a **.ba** request 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 to *N* [6.0i]. This should not be used after output has begun, and particularly not in two-column output. The current line length is stored in **\n($l**.

.hl — Draws a horizontal line the length of the page. 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 request 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 the string **\\\\n(ch**. For example, to number appendixes **A.1** etc., type **.++ RA ´´\\\\n(ch.%´**. Each section (chapter, appendix, etc.) should be preceded by the **.+c** request. It should be mentioned that 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, actually ***[\n($f*]**. This macro is incremented after each call to **.)f**.
***#**	Delayed text number. Actually [**\n($d**].
***{**	Superscript. This string gives upward movement and a change to a smaller point size. Extra space is left above the line to allow room for the superscript.
***}**	Unsuperscript. Inverse to ***{**. For example, to produce a superscript you might type **x*{2*}**, which will produce $\mathbf{x^2}$.
***<**	Subscript. Extra space is left below the line to allow for the subscript.
***>**	Inverse to ***<**.
***(dw**	The day of the week, as a word.
***(mo**	The month, as a word.
***(td**	Today's date, directly printable. The date is of the form February 15, 2003. Other forms of the date can be used 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).
***(lq**	Left quote marks.
***(rq**	Right quote.
***−**	¾ em dash.

13. Special Characters and Marks

There are a number of special characters and diacritical marks (such as accents) available through −me.

Name	Usage	Example	
Acute accent	*´	a*´	á
Grave accent	*`	e*`	è
Umlaut	*:	u*:	ü
Tilde	*~	n*~	ñ
Caret	*^	e*^	ê
Cedilla	*,	c*,	ç
Czech	*v	e*v	ě
Circle	*o	A*o	Å
There exists	*(qe		∃
For all	*(qa		∀

Acknowledgments

I 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

This alphabetical list summarizes all macros, strings, and number registers available in the −me macros. Selected *troff* commands, registers, and functions are included as well; those listed can generally be used with impunity.

The columns are the name of the command, macro, register, or string; the type of the object, and the description. Types are **M** for macro or builtin command (invoked with **.** or **´** in the first input column), **S** for a string (invoked with ***** or ***(**), **R** for a number register (invoked with **\n** or **\n(**), and **F** for a *troff* builtin function (invoked by preceding it with a single backslash).

Lines marked with § are *troff* internal codes. Lines marked with † or ‡ may be defined by the user to get special functions; ‡ indicates that these are defined by default and changing them may have unexpected side effects. Lines marked with ° are specific to *ditroff* (device-independent *troff*).

NAME	TYPE	DESCRIPTION
\(space)	F§	unpaddable 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	section name
\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
\n%	R§	current page number
\&	F§	zero width character, useful for hiding controls
\(*xx*	F§	interpolate special character *xx*
.(b	M	begin block

NAME	TYPE	DESCRIPTION
.(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
.)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
*,	S	cedilla
\–	F§	minus sign
*–	S	3/4 em dash
\0	F§	unpaddable digit-width space
.1c	M	revert to single column output
.2c	M	begin two column output
*:	S	umlaut
*<	S	begin subscript
*>	S	end subscript
.EN	M	end equation
.EQ	M	begin equation
\L´d´	F§	vertical line drawing function for distance *d*
.GE	M°	end *gremlin* picture
.GF	M°	end *gremlin* picture (with flyback)
.GS	M°	start *gremlin* picture
.IE	M°	end *ideal* picture
.IF	M°	end *ideal* picture (with flyback)
.IS	M°	start *ideal* picture
.PE	M°	end *pic* picture
.PF	M°	end *pic* picture (with flyback)
.PS	M°	start *pic* picture
.TE	M	end table
.TH	M	end header of table
.TS	M	begin table
*{	S	begin superscript
\n(.$	R§	number of arguments to macro
\n(.i	R§	current indent
\n(.l	R§	current line length
\n(.s	R§	current point size
*(´	S	acute accent
*(`	S	grave accent
\(´	F§	acute accent

NAME	TYPE	DESCRIPTION
\(`	F§	grave accent
*}	S	end superscript
\^	F§	1/12 em narrow space
*^	S	caret
.ad	M§	set text adjustment
.af	M§	assign format to register
.am	M§	append to macro
.ar	M	set page numbers in Arabic
.as	M§	append to string
.b	M	bold font
.ba	M	set base indent
.bc	M	begin new column
.bi	M	bold italic
\n(bi	R	display (block) indent
.bl	M	blank lines (even at top of page)
\n(bm	R	bottom title margin
.bp	M§	begin page
.br	M§	break (start new line)
\n(bs	R	display (block) pre/post spacing
\n(bt	R	block keep threshold
.bx	M	boxed
\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§	fill output lines
\n(fi	R	footnote indent (first line only)
\n(fm	R	footer margin
.fo	M	set footer
\n(fp	R	footnote pointsize
\n(fs	R	footnote prespace
\n(fu	R	footnote undent (from right margin)
\h´*d*´	F§	local horizontal motion for distance *d*
.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 hyphenation mode
.i	M	italic font
.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§	horizontal line drawing function for distance *d*
.lc	M§	set leader repetition character
.ll	M	set line length
.lp	M	begin left justified paragraph
*(lq	S	left quote marks
.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 number register *x*
\n(*xx*	F§	interpolate number 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§	don’t fill output lines
.nh	M§	turn off hyphenation
.np	M	begin numbered paragraph
.nr	M§	set number register
.ns	M§	no space mode
*o	S	circle (e.g., for Norse Å)
.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 pointsize
\n(ps	R	paragraph prespace
.q	M	quoted
*(qa	S	for all
*(qe	S	there exists
\n(qi	R	quote indent (also shortens line)

NAME	TYPE	DESCRIPTION
\n(qp	R	quote pointsize
\n(qs	R	quote pre/post space
.r	M	roman font
.rb	M	real bold font
.re	M	reset tabs
.rm	M§	remove macro or string
.rn	M§	rename macro or string
.ro	M	set page numbers in roman
*(rq	S	right quote marks
.rr	M§	remove register
.rs	M§	restore spacing
.rt	M§	return to vertical position
\s*S*	F§	inline size change to size *S*
\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 in a smaller pointsize
.so	M§	source input file
\n(so	R	additional section title offset
.sp	M§	vertical space
\n(sp	R	section title pointsize
\n(ss	R	section prespace
.sx	M	change section depth
.sz	M	set pointsize 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 pointsize
.tr	M§	translate
.u	M	underlined
.uh	M	unnumbered section
.ul	M§	underline next line
\v´*d*´	F§	local vertical motion for distance *d*
*v	S	inverted 'v' for czeck "ě"
\w´*S*´	F§	return 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
\\|	F§	1/6 em narrow space
\}	F§	end conditional group

NAME	TYPE	DESCRIPTION
*~	S	tilde
\n(_M	R	section type (as set with .++ macro)