NAME

groff_man − GNU roff macro package for formatting man pages

SYNOPSIS

groff −man [*option* . . .] [*input-file* . . .]
groff −m man [*option* . . .] [*input-file* . . .]

DESCRIPTION

The *man* macro package for *groff* is used to produce manual pages (“man pages”) like the one you are reading. GNU *roff*’s implementation was written by James Clark.

This document presents the macros thematically to aid learners; for those needing only a quick reference, the following table lists them alphabetically, with cross-references to appropriate subsections below.

Macro	Meaning	Subsection
.B	Bold	Font style macros
.BI	Bold, italic alternating	Font style macros
.BR	Bold, roman alternating	Font style macros
.EE	Example end	Document structure macros
.EX	Example begin	Document structure macros
.I	Italic	Font style macros
.IB	Italic, bold alternating	Font style macros
.IP	Indented paragraph	Paragraph macros
.IR	Italic, roman alternating	Font style macros
.LP	(Left) paragraph	Paragraph macros
.ME	Mail-to end	Hyperlink and email macros
.MT	Mail-to start	Hyperlink and email macros
.OP	(Command-line) option	Command synopsis macros
.P	Paragraph	Paragraph macros
.PP	Paragraph	Paragraph macros
.RB	Roman, bold alternating	Font style macros
.RE	Relative-indent end	Document structure macros
.RI	Roman, italic alternating	Font style macros
.RS	Relative-indent start	Document structure macros
.SB	Small bold	Font style macros
.SH	Section heading	Document structure macros
.SM	Small	Font style macros
.SS	Subection heading	Document structure macros
.SY	Synopsis start	Command synopsis macros
.TH	Title heading	Document structure macros
.TP	Tagged paragraph	Paragraph macros
.TQ	Tagged paragraph continuation	Paragraph macros
.UE	URL end	Hyperlink and email macros
.UR	URL start	Hyperlink and email macros
.YS	Synopsis end	Command synopsis macros

Macros whose use we discourage (**.AT**, **.BT**, **.DT**, **.HP**, **.PD**, **.PT**, and **.UC**) are described in subsection “Deprecated features”, below.

Macro reference preliminaries

Each macro is described in a tagged paragraph. Closely related macros, such as **.EX** and **.EE**, are grouped together.

Optional macro arguments are indicated by surrounding them with square brackets. If a macro accepts multiple arguments, arguments containing whitespace must be double-quoted ("one two"), to be interpreted correctly. Most macro arguments are strings that will be output as text; exceptions are noted.

Bear in mind that *groff* is fundamentally a programming system for typesetting. Consequently, the verb “to set” is frequently used below in the sense “to typeset”.

Document structure macros

The highest level of organization of a man page is determined by this group of macros. **.TH** (title heading) identifies the document as a man page and defines information enabling its indexing by *mandb*(8) or a similar tool. Sections (**.SH**), one of which is mandatory and many of which are standardized, facilitate quick location of relevant material by the reader and aid the man page writer to discuss all essential aspects of the topic. Subsections (**.SS**) are optional and permit sections that grow long to develop in a controlled way. Many technical discussions require examples; lengthy ones, especially those reflecting multiple lines of input to or output from the system, are usefully bracketed by **.EX** and **.EE**. When none of the foregoing meets a structural demand, a section of the discussion can be manually indented within **.RS** and **.RE** macros.

.TH *title section* [*footer-middle*] [*footer-outside*] [*header-middle*]

Define the title of the man page as *title* and the section as *section*. See *man*(1) for details on the section numbers and suffixes applicable to your system. *title* and *section* are positioned together at the left and right in the header line (with *section* in parentheses immediately appended to *title*). *footer-middle* is centered in the footer line. *footer-outside* is positioned at the left in the footer line (or at the left on even pages and at the right on odd pages if double-sided printing is active). *header-middle* is centered in the header line. If *section* is a simple integer between 1 and 9 (inclusive), or is exactly "3p", there is no need to specify *header-middle*; the macro package will supply text for it.

For HTML output, headers and footers are completely suppressed.

Additionally, this macro starts a new page; the page number is reset to 1 (unless the **−rC1** option is given on the command line). This feature is intended only for formatting multiple man pages.

A man page should contain exactly one **.TH** call at or near the beginning of the file, prior to any other macro calls.

By convention, *footer-middle* is the most recent modification date of the man page source document, and *footer-outside* is the name and version or release of the project providing it.

.SH [*heading-text*]

Set *heading-text* as a section heading flush left. The text following **.SH** up to the end of the line, or the text on the next input line if **.SH** is given no arguments, is set in bold (or the font specified by the string register **HF**) slightly larger than the base font size. Additionally, the left margin and indentation affecting subsequent text are reset to their default values. Text on input lines after *heading-text* is set as a normal paragraph (**.PP**).

The content of *heading-text* and ordering of sections has been standardized by common practice, as has much of the layout of material within sections. For example, a section called "Name" or "NAME" must exist, must be the first section after the **.TH** call, and must contain only a line of the form

page-topic[**,** . . .] **\−** *summary-description*

for a man page to be properly indexed. See *man*(7) for the conventions prevailing on your system.

.SS [*subheading-text*]

Set *subheading-text* as a subsection heading indented (by default) partway between a section heading and a normally-indented paragraph (**.PP**). The text following **.SS** up to the end of the line, or the text on the next input line if **.SS** is given no arguments, is set in bold (or the font specified by the string register **HF**) at the base font size. Additionally, the left margin and indentation affecting subsequent text are reset to their default values. Text on input lines after *subheading-text* is set as a normal paragraph (**.PP**).

.EX

.EE Begin and end example. After **.EX**, filling and hyphenation are disabled and a constant-width (monospaced) font is selected. Calling **.EE** enables filling and restores the previous hyphenation setting and font.

Example regions are useful for formatting code, shell sessions, and text file contents.

These macros are defined on many (but not all) legacy Unix systems running classic *troff*. To be certain your page will be portable to those systems, copy their definitions from the *an−ext.tmac* file of a *groff* installation.

.RS [*indent*]
Move the left margin to the right by the value *indent*, if specified, and by a default amount otherwise; see subsection "Horizontal and vertical spacing" below. Calls to **.RS** can be nested; each call increments by 1 the indentation level used by **.RE**. The indentation level prior to any **.RS** calls is 1.

.RE [*level*]
Move the left margin back to that corresponding to indentation level *level*. If no argument is given, move the left margin one level back.

Paragraph macros

A typical paragraph (**.PP**) is set at the current left margin, which by default is indented from the left margin of the output device. In man pages and other technical literature, definition lists are frequently encountered; these can be set as "tagged paragraphs" (**.TP** and **.TQ**), which have one or more leading tags followed by a paragraph that has an additional left indent. The indented paragraph (**.IP**) macro is useful to continue the indented content of a narrative started with **.TP**, or to present an itemized or ordered list.

.LP
.PP
.P Begin a new paragraph; these macros are synonymous. They break the output line at the current position, followed by a vertical space downward by a default amount (which can be changed by the deprecated **.PD** macro). The font size and style are reset to defaults; see subsection "Font style macros" below. Finally, the left margin and indentation are reset to default values.

.TP [*indent*]
Set a tagged, indented paragraph. The input line following this macro, known as the *tag*, is printed at the current left margin. Subsequent text is indented by *indent*, if specified, and by a default amount otherwise; see subsection "Horizontal and vertical spacing" below.

If the tag is not as wide as the indentation, the paragraph starts on the same line as the tag, at the applicable indentation, and continues on the following lines. Otherwise, the descriptive part of the paragraph begins on the line following the tag, entirely indented. The line containing the tag can include a macro call, for instance to set the tag in bold with **.B**.

.TP was used to write the first paragraph of this description of **.TP**, and **.IP** the subsequent ones.

.TQ Set an additional tag for a paragraph tagged with **.TP**. The pending output line is broken. The tag on the input line following this macro and subsequent lines are handled as with **.TP**.

This macro is not defined on legacy Unix systems running classic *troff*. To be certain your page will be portable to those systems, copy its definition from the *an−ext.tmac* file of a *groff* installation.

The descriptions of **.LP**, **.PP**, and **.P** above were written using **.TP** and **.TQ**.

.IP [*tag*] [*indent*]
Set an indented paragraph with an optional tag. The *tag* and *indent* arguments, if present, are handled as with **.TP**, with the exception that the *tag* argument to **.IP** cannot include a macro call.

Two convenient use cases for **.IP** are

(1) to start a new paragraph with the same indentation as the previous **.IP** or **.TP** paragraph, if no *indent* argument is given; and

(2) to set a paragraph with a short *tag* that is not semantically important, such as a bullet (•)—obtained with the '\(bu' character escape—or list enumerator, as seen in this very paragraph.

Command synopsis macros

Command synopses are a staple of section 1 and 8 man pages. These macros aid you to construct one that has the classical Unix appearance. Furthermore, some tools are able to interpret these macros semantically and treat them appropriately for localization and/or presentation. A command synopsis is wrapped in **.SY/.YS** calls, with command-line options of some formats indicated by **.OP**.

These macros are not defined on legacy Unix systems running classic *troff*. To be certain your page will be portable to those systems, copy their definitions from the *an−ext.tmac* file of a *groff* installation.

.SY *command*
: Begin synopsis. Hyphenation is turned off. The *command* argument is set in bold. The output line is filled as normal, but if a break is required, subsequent output lines are indented by the width of *command* plus a space.

.OP *option-name* [*option-argument*]
: Indicate an optional command parameter called *option-name*, which is set in bold. If the option takes an argument, specify *option-argument* using a noun, abbreviation, or hyphenated noun phrase. If present, *option-argument* is preceded by a space and set in italics. Square brackets (in roman) surround both arguments.

.YS
: End synopsis. Restore indentation and hyphenation to previous values.

Multiple **.SY/.YS** blocks can be specified, for instance to distinguish differing modes of operation of a complex command like *tar*(1); each will be separated by a paragraph space.

.SY can also be repeated multiple times before a closing **.YS**, which is useful to indicate synonymous ways of invoking a particular mode of operation.

For example,

```
.SY groff
.OP \-abcegiklpstzCEGNRSUVXZ
.OP \-d cs
.OP \-f fam
.OP \-F dir
.OP \-I dir
.OP \-K arg
.OP \-L arg
.OP \-m name
.OP \-M dir
.OP \-n num
.OP \-o list
.OP \-P arg
.OP \-r cn
.OP \-T dev
.OP \-w name
.OP \-W name
.RI [ file
\&.\|.\|.\&]
.YS
.
.SY groff
.B \-h
.SY groff
.B \-\-help
.YS
```

produces the following output.

groff [**−abcegiklpstzCEGNRSUVXZ**] [**−d** *cs*] [**−f** *fam*] [**−F** *dir*] [**−I** *dir*] [**−K** *arg*] [**−L** *arg*] [**−m** *name*] [**−M** *dir*] [**−n** *num*] [**−o** *list*] [**−P** *arg*] [**−r** *cn*] [**−T** *dev*] [**−w** *name*] [**−W** *name*] [*file* . . .]

groff −h
groff −−help

Several features of the above example are of note.

- The empty request (.), which does nothing, is used for vertical spacing in the input file for readability by the document maintainer. Do not put empty lines in a *roff* source document.
- The command and option names are presented in **bold** to cue the user that they should be input literally.
- Option dashes are specified with the '\−' escape sequence; this is an important practice to make them clearly visible and to facilitate cut-and-paste from the rendered man page to a shell prompt or text file.
- Option arguments and command operands are presented in *italics* (underlined on some output devices, such as terminals and emulators), to cue the user that they must be replaced with appropriate text.
- Symbols that are neither to be typed literally nor simply replaced appear in the roman style; brackets surround optional arguments, and an ellipsis indicates that the previous syntactical element may be repeated arbitrarily.

  Some man pages use a brace-and-pipe notation such as "{**−−diff**|**−−compare**}" to indicate that one and only one of the '|'-separated items within the braces must be input. If this braced construct is furthermore surrounded by square brackets, it means that at most one of the items is accepted.

  Authors of man pages should note the use of the zero-width space escape sequence '\&' on both sides of the ellipsis; this is a good practice to avoid surprises in the event the ellipsis gets refilled in your text editor. See "Portability", below. The morbidly curious may consult *groff*(7) regarding the narrow-space escape sequence '\|'.

Hyperlink and email macros

Email addresses are bracketed with **.MT/.ME** and URL hyperlinks with **.UR/.UE**.

These macros are not defined on legacy Unix systems running classic *troff*. To be certain your page will be portable to those systems, copy their definitions from the *an−ext.tmac* file of a *groff* installation.

.MT *address*
.ME [*punctuation*]

Identify *address* as an RFC 6068 *addr-spec* for a "mailto:" URI with the text between the two macro calls as the link text. A *punctuation* argument to **.ME** is placed at the end of the link text without intervening space. Note that *address* may not be visible in the output text, particularly if the man page is being viewed as HTML. On a device that is not a browser, *address* is set in angle brackets after the link text and before *punctuation*.

When rendered by *groff* to a TTY or PostScript output device,

```
Contact
.MT fred.foonly@\:fubar.net
Fred Foonly
.ME
for more information.
```

displays as: "Contact Fred Foonly ⟨fred.foonly@fubar.net⟩ for more information.".

The use of '\:' to insert hyphenless discretionary breaks is a *groff* extension and can be omitted.

.UR *URL*
.UE [*punctuation*]

Identify *URL* as an RFC 3986 URI hyperlink with the text between the two macro calls as the link text. A *punctuation* argument to **.UE** is placed at the end of the link text without intervening space. Note that *URL* may not be visible in the output text, particularly if the man page is being viewed as HTML. On a device that is not a browser, *URL* is set in angle brackets after the link text and before *punctuation*.

When rendered by *groff* to a TTY or PostScript output device,

```
The GNU Project of the Free Software Foundation hosts the
.UR https://\:www.gnu.org/\:software/\:groff/
Groff home page
.UE .
```

displays as: "The GNU Project of the Free Software Foundation hosts the Groff home page ⟨https://www.gnu.org/software/groff/⟩.".

The use of '\:' to insert hyphenless discretionary breaks is a *groff* extension and can be omitted.

Font style macros

The *man* macro package is limited in its font styling options, offering only **bold** (**.B**), *italic* (**.I**), and roman (the default). Italic text is usually set underscored instead on terminals and other classical *nroff*-style output devices. The **.SM** and **.SB** macros set text in roman or bold, respectively, at a smaller point size; these differ visually from regular-sized roman or bold text only on *troff*-style output devices. The foregoing macros cause word breaks before and after their arguments, but it is often necessary to set text in different styles without intervening whitespace. The macros **.BI**, **.BR**, **.IB**, **.IR**, **.RB**, and **.RI**, where 'B', 'I', and 'R' indicate bold, italic, and roman, respectively, set their odd- and even-numbered arguments in alternating styles, with no whitespace separating them.

Because font styles are presentational rather than semantic, conflicting traditions have arisen regarding which font styles should be used to mark file or path names, environment variables, in-line literals, and even man page cross-references.

The default font size and family (for *troff* output devices) is 10-point Times. The default style is roman.

.B [*text*]

Set *text* in bold. If the macro is given no arguments, the text of the next input line is set in bold.

Use bold for literal portions of syntax synopses, for command-line options in running text, and for literals that are major topics of the subject under discussion; for example, this page uses bold for macro and register names. In **.EX/.EE** examples of interactive I/O (such as a shell session), set only the user-typed input in bold.

.I [*text*] Set *text* in italics. If the macro is given no arguments, the text of the next input line is set in italics.

Use italics for file and path names, for environment variables, for enumeration or preprocessor constants in C, for variable (user-determined) portions of syntax synopses, for the first occurrence only of a technical concept being introduced, for names of works of software (including commands and functions, but excluding names of operating systems or their kernels), and anywhere a parameter requiring replacement by the user is encountered. An exception involves variable text in a context that is already marked up in italics, such as file or path names with variable components; in such cases, follow the convention of mathematical typography: set the file or path name in italics as usual (see **.IR** below), but use roman for the variable part, and italics again in running roman text when referring to the variable material.

.SM [*text*]

Set *text* one point size smaller than the default size. If the macro is given no arguments, the text of the next input line is set smaller.

Note: *nroff*-style output devices, such as terminals, will render *text* at the normal font size instead. Do not rely upon **.SM** to communicate semantic information distinct from using roman style at the

normal size; it will be hidden from readers using such devices.

.SB [*text*]
Set *text* in bold, one point size smaller than the default size. If the macro is given no arguments, the text of the next input line is set smaller and in bold.

Note: *nroff*-style output devices, such as terminals, will render *text* in bold at the normal font size instead. Do not rely upon **.SB** to communicate semantic information distinct from using bold style at the normal size; it will be hidden from readers using such devices.

Note what is *not* prescribed for setting in bold or italics above: elements of "synopsis language" such as ellipses and brackets around options; proper names and adjectives; titles of anything other than works of literature or software; identifiers for standards documents or technical reports such as CSTR #54, RFC 1918, Unicode 11.0, or POSIX.1-2017; acronyms; and occurrences after the first of a technical term or piece of jargon. Again, the names of operating systems and their kernels are, by practically universal convention, set in roman.

Be frugal with the use of italics for emphasis, and particularly with the use of bold. Brief runs of literal text, such as references to individual characters or short strings, including section and subsection headings of man pages, are suitable objects for quotation; see the '\(lq', '\(rq', '\(oq', and '\(cq' escapes in subsection "Portability" below.

Unlike the above font style macros, the font alternation macros below accept only arguments on the same line as the macro call. If whitespace is required within one of the arguments, first consider whether the same result could be achieved with as much clarity by using the single-style macros on separate input lines. When it cannot, double-quote an argument with one or more embedded space characters. Setting all three different styles within one whitespace-delimited word presents challenges; it is possible with the '\c' and/or '\f' escapes, but see subsection "Portability" below for caveats.

.BI *bold-text italic-text* ...
Set each argument in bold and italics, alternately.

```
.BI \-r name = n
```

.BR *bold-text roman-text* ...
Set each argument in bold and roman, alternately.

```
Any such change becomes effective with the first use of
.BR .NH ,
.I after
the new alias is defined.
```

.IB *italic-text bold-text* ...
Set each argument in italics and bold, alternately.

```
All macro package files must be named
.IB name .tmac
to fully use the
.I tmac
mechanism.
```

.IR *italic-text roman-text* ...
Set each argument in italics and roman, alternately.

```
This is the first command of the
.IR prologue .
```

.RB *roman-text bold-text* ...
Set each argument in roman and bold, alternately.

```
Also, the statement
.RB \(oq "delim on" \(cq
is not handled specially.
```

.RI *roman-text italic-text* ...

Set each argument in roman and italics, alternately.

```
.RI [ file
\&.\|.\|.\&]
```

Horizontal and vertical spacing

The *indent* argument accepted by **.RS**, **.IP**, **.TP**, and the deprecated **.HP** is a number plus an optional scaling indicator. If no scaling indicator is given, the *man* package assumes 'n'; that is, the width of a letter "n" in the font current when the macro is called. See section "Numerical Expressions" in *groff*(7) for further details. An indent specified in a call to **.IP**, **.TP**, or the deprecated **.HP** persists until (1) another of these macros is called with an explicit indent argument, or (2) **.SH**, **.SS**, or **.PP** or its synonyms is called; these clear the indent entirely.

Indents set by **.RS** move the left margin and persist until **.RS**, **.RE**, **.SH**, or **.SS** is called. The default indentation, exhibited by ordinary **.PP** paragraphs not within an **.RS/.RE** relative indent, is 7.2n in *troff* mode and 7n in *nroff* mode. The HTML output device is an exception; it ignores indentation completely. This same indentation is used again (additively) for the defaults of **.IP**, **.TP**, **.RS**, and the deprecated **.HP**. Section headings (**.SH**) are set flush with the left margin of the output device, and subsection headings (**.SS**) are indented 3n.

Resist the temptation to mock up tabular or multi-column output with ASCII tab characters or the indentation arguments to **.IP**, **.TP**, **.RS**, or the deprecated **.HP**; the result may not render comprehensibly on an output device you fail to check, or which is developed in the future. The table preprocessor *tbl*(1) can likely meet your needs.

The following macros cause a line break with the insertion of vertical space: **.SH**, **.SS**, **.TP**, **.TQ**, **.PP** (and its synonyms), **.IP**, and the deprecated **.HP**. The default inter-section and inter-paragraph spacing is 1 line in *nroff* mode, and 0.4v in *troff* mode. (The deprecated macro **.PD** can change this vertical spacing, but its use is discouraged.) The macros **.RS**, **.RE**, **.EX**, and **.EE** also cause a break but no insertion of vertical space.

Number registers

Number registers are described in section "Options" below.

String registers

The following strings are defined.

***R** expands to the character escape for the "registered sign" glyph, '\(rg', if available, and "(Reg.)" otherwise.

***S** expands to an escape setting the font size to the document default.

***(HF** expands to the font identifier used to print headings and subheadings. The default is 'B'.

***(lq**
***(rq** expand to the character escapes for left and right double-quotation marks, '\(lq' and '\(rq', respectively.

***(Tm** expands to the character escape for the "trade mark sign" glyph, '\(tm', if available, and "(TM)" otherwise.

Interaction with preprocessors

When a preprocessor like *tbl* or *eqn* is needed, a hint can be given to the man page formatter by making the first line of a man page look like this:

'\" *word*

Note that the line starts with an apostrophe ('), not a dot, and that a single space character follows the double quote. The *word* consists of one letter for each needed preprocessor: 'e' for *eqn*, 'r' for *refer*, and 't' for *tbl*. Modern implementations of the *man* program interpret this first line and automatically call the right preprocessor(s).

The usual *tbl* and *eqn* macros for table and equation inclusion, **.TS**, **.T&**, **.TE**, **.EQ**, and **.EN**, may be used

freely. Note that *nroff* output devices are extremely limited in presentation of mathematical equations.

Portability

The two major syntactical categories of *roff* languages are requests and escapes. Since the *man* macros are implemented in terms of *groff* requests and escapes, one can, in principle, supplement the functionality of *man* with these lower-level elements where necessary.

Note, however, that using raw *groff* requests is likely to make your page render poorly on the class of viewers that transform it to HTML. Some requests make implicit assumptions about things like character and page sizes that may not hold in an HTML environment; also, many of these viewers don't interpret the full *groff* vocabulary, a problem that can lead to portions of your text being silently dropped.

For portability to modern viewers, it is best to write your page entirely with the macros described in this page (except for the ones identified as deprecated, which should be avoided). The macros we have described as extensions (**.EX/.EE**, **.SY/.OP/.YS**, **.UR/.UE**, and **.MT/.ME**) should be used with caution, as they may not yet be built in to some viewer that is important to your audience. If in doubt, copy the implementation into your page—after the **.TH** call and the "Name" section, to accommodate timid *mandb* implementations.

Similar caveats apply to escapes. Some escape sequences are however required for correct typesetting even in man pages and usually do not cause portability problems:

\" Comment. Everything after the double-quote to the end of the input line is ignored. Whole-line comments are frequently placed immediately after the empty request '.'.

*****newline*

Join the next input line to the current one. Except for the update of the input line counter (used for diagnostic messages and related purposes), a series of lines ending in backslash-newline is transparent to *groff*. Use this escape to break excessively input long lines for document maintenance.

\~ Adjustable, non-breaking space character. Use this escape to prevent a break inside a short phrase or between a numerical quantity and its corresponding unit(s).

```
Before starting the motor, set the output speed to\~1.
There are 1,024\~bytes in 1\~kiB.
CSTR\~#8 documents the B language.
```

\& Zero-width space. Append to an input line to prevent an end-of-sentence punctuation sequence from being recognized as such, or insert at the beginning of an input line to prevent a dot or apostrophe from being interpreted as the beginning of a *roff* request.

\(aq ASCII apostrophe. Use for syntax elements of programming languages because some output devices might replace unescaped apostrophes with right single quotation marks.

\(oq Opening single quotation mark.
\(cq Closing single quotation mark.

Use these for paired directional single quotes, 'like this'.

\(dq ASCII double-quote. Sometimes needed after macro calls to prevent the interpretation of the ASCII quotation mark character '"' as the beginning or end of a macro argument.

\(lq Left double quotation mark.
\(rq Right double quotation mark.

Use these for paired directional double quotes, "like this".

\(em Em-dash. Use for an interruption in a sentence—such as this one.

\(en En-dash. Use to separate the two ends of a range, in particular between numbers, for example: the digits 1–9.

\(ga ASCII grave accent. Use for syntax elements of programming languages, for example shell command substitutions, because some output devices might replace unescaped grave accents with left single quotation marks.

\(ha ASCII circumflex accent. Use for syntax elements of programming languages because some output devices might replace unescaped circumflex accents with non-ASCII glyphs like the Unicode U+02C6 modifier letter circumflex.

\(ti ASCII tilde. Use for syntax elements of programming languages because some output devices might replace unescaped tildes with non-ASCII glyphs like the Unicode U+02DC small tilde.

\– Minus sign. Also use this to display syntax elements that require the ASCII hyphen-minus character, for example command-line options and C language operators. The unescaped '–' input character is not appropriate for these cases because it may render as a hyphen on some output devices.

\c If this escape sequence occurs at the end of an input line, no white space is inserted between the last glyph on it and the first glyph resulting from the next input line. This is occasionally useful when three different fonts are needed in a single word.

```
Normally, the final output file should be named
.IB file .pdf\c
\&.
```

Note that when using this trick with the **.BI** or **.RI** macros, you will need to manually add an italic correction escape '\/' before the '\c' due to way macros expand their arguments.

```
Files processed with
.B groff \-mom
(or
.BI "\-m " mom\/\c
) produce PostScript output by default.
```

Alternatively, and perhaps with better portability, the '\f' font escape sequence can be used; see below. Using '\c' to include the output from more than one input line into the next-line argument of a **.TP** macro will render incorrectly with *groff* 1.22.3, *mandoc* 1.14.1, older versions of these programs, and perhaps with some other formatters.

\e Widely used in man pages to represent a backslash output glyph. It works reliably as long as the **.ec** request is not used, which should never happen in man pages, and it is slightly more portable than the more exact '\(rs' ("reverse solidus") escape sequence.

\fB, **\fI**, **\fR**, **\fP**

Switch to bold, italic, roman, or back to the previous font, respectively. Either these or '\c' is needed when three different fonts are required in a single whitespace-delimited word.

```
.RB [ \-\-reference\-dictionary=\fI\,name\/\fP ]
```

```
.RB [ \-\-reference\-dictionary=\c
.IR name ]
```

Font escapes may be more portable than '\c'. As shown above, it is up to you to account for italic corrections with '\/' and '\,', which are themselves *groff* extensions, if desired and if supported by your implementation.

Note that '\fP' reliably returns to the style in use immediately preceding the previous '\f' escape only if no sectioning, paragraph, or font face macro calls have intervened.

As long as only two fonts are needed in any single whitespace-delimited word, font alternation macros like **.BI** usually result in more readable source code than '\f' escapes do.

For maximum portability, escape sequences and special characters not listed above are better avoided in man pages.

Deprecated features

Use of the following is discouraged.

.AT [*system* [*release*]]

Alter the footer for use with AT&T man pages, overriding any definition of the *footer-outside* argument to **.TH**. This macro exists only for compatibility; don't use it.

The first argument *system* can be:

3 7th edition *(default)*

4 System III

5 System V

The optional second argument *release* specifies the release number, such as in "System V Release 3".

.BT Set the page footer. Redefine this macro to get control of the footer.

.DT Set tabs every 0.5 inches. Since this macro is always called during a **.TH** macro, it makes sense to call it only if the tab positions have been changed.

Use of this presentation-level macro is deprecated. It translates poorly to HTML, under which exact whitespace control and tabbing are not readily available. Thus, information or distinctions that you use **.DT** to express are likely to be lost. If you feel tempted to use it, you should probably be composing a table using *tbl*(1) markup instead.

.HP [*indent*]

Set up a paragraph with a hanging left indentation. The *indent* argument, if present, is handled as with **.TP**.

Use of this presentation-level macro is deprecated. While it is universally portable to legacy Unix systems, a hanging indentation cannot be expressed naturally under HTML, and many HTML-based manual viewers simply interpret it as a starter for a normal paragraph. Thus, any information or distinction you tried to express with the indentation may be lost.

.PD [*vertical-space*]

Define the vertical space between paragraphs or (sub)sections. The optional argument *vertical-space* specifies the amount of space; the default scaling is 'v'). Without an argument, the spacing is reset to its default value; see "Horizontal and vertical spacing" above.

Use of this presentation-level macro is deprecated. It translates poorly to HTML, under which exact control of inter-paragraph spacing is not readily available. Thus, information or distinctions that you use **.PD** to express are likely to be lost.

.PT Set the page header. Redefine this macro to get control of the header.

.UC [*version*]

Alter the footer for use with BSD man pages, overriding any definition of the *footer-outside* argument to **.TH**. This macro exists only for compatibility; don't use it.

The argument *version* can be:

3 3rd Berkeley Distribution *(default)*

4 4th Berkeley Distribution

5 4.2 Berkeley Distribution

6 4.3 Berkeley Distribution

7 4.4 Berkeley Distribution

History

According to its own *man*(7) page, Version 7 Unix (1979) supported all of the macros described in this page not listed as GNU extensions, except **.P**, **.SB**, **.SS**, and the deprecated **.AT**, **.BT**, **.PT**, and **.UC**. The only string registers defined were **R** and **S**; no number registers were documented.

OPTIONS

The following *groff* options set number registers recognized and used by the *man* macro package.

−rcR=1

Continuous rendering. Create a single, very long page instead of multiple pages. This is the default in *nroff* mode. Use **−rcR=0** to disable it.

−rC1 Number pages continuously. If more than one man page is given on the command line, number the pages continuously, rather than starting each at 1.

−rD1 Enable double-sided printing. Footers for even and odd pages are formatted differently; see the description of **.TH** in "Document structure macros", above.

−rFT=*footer-distance*
Set distance of the footer, relative to the bottom of the page if negative or relative to the top if positive, to *footer-distance*. The default is −0.5i.

−rHY=*flags*
Set hyphenation flags. Permissible values of *flags* are documented in section "Hyphenation" of *groff* (7). The default is 4 if continuous rendering is enabled (**−rcR=1** above), and 6 otherwise.

−rIN=*indent*
Set the body text indentation (for normal paragraphs) to *indent*. See "Horizontal and vertical spacing" above for the default indentation value. For *nroff*, *indent* should always be an integer multiple of unit 'n' to get consistent indentation.

−rLL=*line-length*
Set line length. If this option is not given, the line length is set to respect any value set by a prior ".ll" request (which *must* be in effect when the **.TH** macro is invoked), if this differs from the built-in default for the formatter; otherwise it defaults to 78n in *nroff* mode and 6.5i in *troff* mode.

Note that the use of a ".ll" request to initialize the line length is supported for backward compatibility with some versions of the *man* program; direct initialization of the **LL** register should *always* be preferred to the use of such a request. In particular, note that a ".ll 65n" request does *not* preserve the normal *nroff* default line length (the *man* default initialization to 78n prevails), whereas the **−rLL=65n** option, or an equivalent ".nr LL 65n" request preceding the use of the **.TH** macro, *does* set a line length of 65n.

−rLT=*title-length*
Set title length. If this option is not given, the title length defaults to the line length.

−rP*n* Start enumeration of pages at *n* rather than 1.

−rS*point-size*
Use *point-size* as the base document font size. Acceptable values are 10, 11, or 12. See subsection "Font style macros" above for the default.

−rSN=*subsection-indent*
Set subsection indentation to *subsection-indent*. See "Horizontal and vertical spacing" above for the default indentation value.

−rX*p* After page *p*, number pages as *p*a, *p*b, *p*c, and so forth. For example, the option **−rX2** produces the following page numbers: 1, 2, 2a, 2b, 2c, and so on.

FILES

/usr/share/groff/1.22.4/tmac/man.tmac
/usr/share/groff/1.22.4/tmac/an.tmac
These are wrapper files to call *andoc.tmac*.

/usr/share/groff/1.22.4/tmac/andoc.tmac
This brief *groff* program detects whether the *man* or *mdoc* macro package is being used by a document and loads the correct macro definitions, taking advantage of the fact that pages using them must call **.TH** or **.Dd**, respectively, as their first macro. Because the wrappers above load this file, a *man* program or user typing, for example, "groff −man page.1", need not know which package the file *page.1* uses. Multiple man pages, in either format, can be handled.

/usr/share/groff/1.22.4/tmac/an−old.tmac
Most *man* macros are contained in this file. It also loads the GNU extensions from *an−ext.tmac* (see below).

/usr/share/groff/1.22.4/tmac/an−ext.tmac

The extension macro definitions for **.SY**, **.OP**, **.YS**, **.TQ**, **.EX/.EE**, **.UR/.UE**, and **.MT/.ME** are contained in this file, which is written in classic *troff* and permissively licensed—not copylefted. Man page authors concerned about portability to legacy Unix systems are encouraged to copy these definitions into their pages, and maintainers of *troff* implementations or work-alike systems that format man pages are encouraged to re-use them.

Note that the definitions for these macros are read after the call of **.TH**, so they will replace any macros of the same names preceding it in your file. If you use your own implementations of these macros, they must be defined after calling **.TH** to have any effect.

/usr/share/groff/site−tmac/man.local

Local changes and customizations should be put into this file.

NOTES

Some tips on troubleshooting your man pages follow.

- **.RS** doesn't indent relative to my indented paragraph

  The **.RS** macro sets the indentation relative to the amount of a *normal* paragraph (**.PP** and its synonyms). The same default indentation amount is used for **.RS**, **.IP**, **.TP**, and the deprecated **.HP**. If you need to start an indent relative to an indented paragraph, call **.RS** repeatedly until an acceptable indentation is achieved, or give **.RS** an indentation argument that is at least as much as the paragraph's indentation amount relative to an adjacent **.PP** paragraph. See "Horizontal and vertical spacing" above for the values.

- **.RE** doesn't reset the indent to the expected level
- warning: scale indicator invalid in this context
- warning: number register 'an−saved−margin*n*' not defined
- warning: number register 'an−saved−prevailing−indent*n*' not defined

  The **.RS** macro takes an indentation *amount* as an argument; the **.RE** macro's argument is a specific indentation *level*. **.RE 1** goes to the level before any **.RS** macros were called, **.RE 2** goes to the level of the first **.RS** call you made, and so forth. If you desire symmetry in your macro calls, simply issue one **.RE** without an argument for each **.RS** that precedes it.

  After calls to the **.SH** and **.SS** sectioning macros, all relative indents are cleared and calls to **.RE** have no effect.

AUTHORS

The GNU version of the *man* macro package was written by James Clark and contributors. The extension macros were written by Werner Lemberg ⟨wl@gnu.org⟩ and Eric S. Raymond ⟨esr@thyrsus.com⟩.

This document was originally written for the Debian GNU/Linux system by Susan G. Kleinmann ⟨sgk@debian.org⟩. It was corrected and updated by Werner Lemberg and G. Branden Robinson. The extension macros were documented by Eric S. Raymond; he also originated the portability section, to which Ingo Schwarze contributed most of the material on escape sequences.

SEE ALSO

Groff: The GNU Implementation of troff, by Trent A. Fisher and Werner Lemberg, is the main *groff* documentation. You can browse it interactively with "info groff".

tbl(1), *eqn*(1), and *refer*(1) are preprocessors used with man pages.

man(1) describes the man page formatter on your system.

groff_mdoc(7) describes the *groff* version of the BSD-originated alternative macro package for man pages.

groff(7), *groff_char*(7), *man*(7)