GROFF_MAN(7) Miscellaneous Information Manual GROFF_MAN(7)

NAME top

groff_man - GNU roff macro package for formatting man pages

SYNOPSIS top

groff -man [*options ...*] [*files ...*]
groff -m man [*options ...*] [*files ...*]

DESCRIPTION top

The **man** macros used to generate *man pages* with *groff* were written by James Clark. This document provides a brief summary of the use of each macro in that package.

OPTIONS top

The **man** macros understand the following command-line options (which define various registers).

-rcR=1 This option (the default if in nroff mode) creates a single, very long page instead of multiple pages. Say **-rcR=0** to disable it.

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

-rD1 Double-sided printing. Footers for even and odd pages are formatted differently.

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

-rHY=*flags*
Set hyphenation flags. Possible values are 1 to hyphenate without restrictions, 2 to not hyphenate the last word on a

page, 4 to not hyphenate the last two characters of a word, and 8 to not hyphenate the first two characters of a word. These values are additive; the default is 14.

-rIN=*width*
Set body text indentation to *width*. The default is 7n for *nroff*, 7.2n for *troff*. For *nroff*, this value 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*nnn* Enumeration of pages start with *nnn* rather than with 1.

-rS*xx* Base document font size is *xx* points (*xx* can be 10, 11, or 12) rather than 10 points.

-rSN=*width*
Set sub-subheading indentation to *width*. The default is 3n.

-rX*nnn* After page *nnn*, number pages as *nnn*a, *nnn*b, *nnn*c, etc. For example, the option '-rX2' produces the following page numbers: 1, 2, 2a, 2b, 2c, etc.

USAGE top

This section describes the available macros for manual pages. For further customization, put additional macros and requests into the file **man.local**, which is loaded immediately after the **man** package.

.EX
.EE Example/End Example. After **.EX**, filling is disabled and the font is set to constant-width. This is useful for formatting

code, command, and configuration-file examples. The **EE** macro restores filling and restores the previous font.

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.

.HP [*nnn*]
Set up a paragraph with hanging left indentation. The indentation is set to *nnn* if that argument is supplied (the default unit is 'n' if omitted), otherwise it is set to the previous indentation value specified with **.TP**, **.IP**, or **.HP** (or to the default value if none of them have been used yet). Font size and face are reset to its default values. The following paragraph illustrates the effect of this macro with hanging indentation set to 4 (enclosed by **.RS** and **.RE** to set the left margin temporarily to the current indentation):

This is a paragraph following an invocation of the **HP** macro.
As you can see, it produces a paragraph where all lines but the first are indented.

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.

.IP [*designator*] [*nnn*]
Set up an indented paragraph, using *designator* as a tag to mark its beginning. The indentation is set to *nnn* if that argument is supplied (the default unit is 'n' if omitted), otherwise it is set to the previous indentation value specified with **.TP**, **.IP**, or **.HP** (or to the default value if none of them have been used yet). Font size and face of the paragraph (but not the designator) are reset to its default values.

To start an indented paragraph with a particular indentation but without a designator, use '""' (two doublequotes) as the second argument.

For example, the following paragraphs were all set up with bullets as the designator, using '.IP \(bu 4'. The whole block has been enclosed with **.RS** and **.RE** to set the left margin temporarily to the current indentation value.

- **IP** is one of the three macros used in the **man** package to format lists.

- **HP** is another. This macro produces a paragraph with a left hanging indentation.

- **TP** is another. This macro produces an unindented label followed by an indented paragraph.

.LP
.PP
.P These macros are mutual aliases. Any of them causes a line break at the current position, followed by a vertical space downwards by the amount specified by the **PD** macro. The font size and shape are reset to the default value (normally 10pt Roman). Finally, the current left margin and the indentation is reset to the default values.

.RE [*nnn*]
This macro moves the left margin back to level *nnn*, restoring the previous left margin. If no argument is given, it moves one level back. The first level (i.e., no call to **.RS** yet) has number 1, and each call to **.RS** increases the level by 1.

.RS [*nnn*]
This macro moves the left margin to the right by the value *nnn* if specified (default unit is 'n'); otherwise it is set to the previous indentation value specified with **.TP**, **.IP**, or **.HP** (or to the default value if none of them have been used yet). The indentation value is then set to the default.

Calls to the **RS** macro can be nested.

.SH [*text for a heading*]
Set up an unnumbered section heading sticking out to the left. Prints out all the text following **.SH** up to the end of the line (or the text in the next input line if there is no argument to **.SH**) in bold face (or the font specified by the string **HF**), one size larger than the base document size. Additionally, the left margin and the indentation for the following text is reset to the default values.

.SS [*text for a heading*]
Set up a secondary, unnumbered section heading. Prints out all the text following **.SS** up to the end of the line (or the text in the next input line if there is no argument to **.SS**) in bold face (or the font specified by the string **HF**), at the same size as the base document size. Additionally, the left margin and the indentation for the following text is reset to the default values.

.TH *title section* [*extra1*] [*extra2*] [*extra3*]
Set the title of the *man page* to *title* and the section to *section*, which must take on a value between 1 and 8. The value *section* may also have a string appended, e.g. '.pm', to indicate a specific subsection of the *man pages*. Both *title* and *section* are positioned at the left and right in the header line (with *section* in parentheses immediately appended to *title*. *extra1* is positioned in the middle of the footer line.

extra2 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). *extra3* is centered in the header line.

For HTML output, headers and footers are completely suppressed.

Additionally, this macro starts a new page; the new line number is 1 again (except if the '-rC1' option is given on the command line) -- this feature is intended only for formatting multiple *man pages*; a single *man page* should contain exactly one **TH** macro at the beginning of the file.

.TP [*nnn*]
Set up an indented paragraph with label. The indentation is set to *nnn* if that argument is supplied (the default unit is 'n' if omitted), otherwise it is set to the previous indentation value specified with **.TP**, **.IP**, or **.HP** (or to the default value if none of them have been used yet).

The first input line of text following this macro is interpreted as a string to be printed flush-left, as it is appropriate for a label. It is not interpreted as part of a paragraph, so there is no attempt to fill the first line with text from the following input lines. Nevertheless, if the label is not as wide as the indentation the paragraph starts at the same line (but indented), continuing on the following lines. If the label is wider than the indentation the descriptive part of the paragraph begins on the line following the label, entirely indented. Note that neither font shape nor font size of the label is set to a default value; on the other hand, the rest of the text has default font settings.

The **TP** macro is the macro used for the explanations you are just reading.

.TQ The **TQ** macro sets up header continuation for a **TP** macro. With it, you can stack up any number of labels (such as in a glossary, or list of commands) before beginning the indented paragraph. For an example, look up the documentation of the **LP**, **PP**, and **P** macros.

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.

To summarize, the following macros cause a line break with the insertion of vertical space (which amount can be changed with the **PD** macro): **SH**, **SS**, **TP**, **TQ**, **LP** (**PP**, **P**), **IP**, and **HP**. The macros **RS**, **RE**, **EX**, and **EE** also cause a break but no insertion of vertical space.

MACROS TO SET FONTS top

The standard font is Roman; the default text size is 10 point.

.B [*text*]
Causes *text* to appear in bold face. If no text is present on the line where the macro is called the text of the next input line appears in bold face.

.BI *text*
Causes text on the same line to appear alternately in bold face and italic. The text must be on the same line as the macro call. Thus

```
.BI this "word and" that
```

would cause ‘this’ and ‘that’ to appear in bold face, while ‘word and’ appears in italics.

.BR *text*
Causes text on the same line to appear alternately in bold face and roman. The text must be on the same line as the macro call.

.I [*text*]
Causes *text* to appear in italic. If no text is present on the line where the macro is called the text of the next input line appears in italic.

.IB *text*
Causes text to appear alternately in italic and bold face. The text must be on the same line as the macro call.

.IR *text*
Causes text on the same line to appear alternately in italic and roman. The text must be on the same line as the macro call.

.RB *text*
Causes text on the same line to appear alternately in roman and bold face. The text must be on the same line as the macro call.

.RI *text*
Causes text on the same line to appear alternately in roman and italic. The text must be on the same line as the macro call.

.SB [*text*]
Causes the text on the same line or the text on the next input line to appear in boldface font, one point size smaller than the default font.

.SM [*text*]
Causes the text on the same line or the text on the next input line to appear in a font that is one point size smaller than the default font.

MACROS TO DESCRIBE HYPERLINKS AND EMAIL ADDRESSES top

The following 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.

Using these macros helps ensure that you get hyperlinks when your manual page is rendered in a browser or other program that is Web-enabled.

.MT *address*
.ME [*punctuation*]
Wrap an email address. The argument of **.MT** is the address; text following, until **.ME**, is a name to be associated with the address. Any argument to the **ME** macro is pasted to the end of the link text. On a device that is not a browser,

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

usually displays like this: “contact Fred Foonly <fred.foonly@fubar.net> for more information”.

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

.UR *URL*
.UE [*punctuation*]
Wrap a World Wide Web hyperlink. The argument to **.UR** is the URL; thereafter, lines until **.UE** are collected and used as the link text. Any argument to the **UE** macro is pasted to the end of the text. On a device that is not a browser,

```
this is a link to
.UR http://\:randomsite.org/\:fubar
some random site
.UE ,
given as an example
```

usually displays like this: “this is a link to some random site ⟨http://randomsite.org/fubar⟩, given as an example”.

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

MACROS TO DESCRIBE COMMAND SYNOPSES top

The following 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.

These macros are a convenience for authors. They also assist automated translation tools and help browsers in recognizing command synopses and treating them differently from running text.

.OP *key value*
: Describe an optional command argument. The arguments of this macro are set surrounded by option braces in the default Roman font; the first argument is printed with a bold face, while the second argument is typeset as italic.

.SY *command*
: Begin synopsis. Takes a single argument, the name of a command. Text following, until closed by **.YS**, is set with a hanging indentation with the width of *command* plus a space. This produces the traditional look of a Unix command synopsis.

.YS
: This macro restores normal indentation at the end of a command synopsis.

Here is a real 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
.IR .\|.\|. ]
.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 ...]
```

If necessary, you might use **br** requests to control line breaking. You can insert plain text as well; this looks like the traditional (unornamented) syntax for a required command argument or filename.

MISCELLANEOUS top

The default indentation is 7.2n in troff mode and 7n in nroff mode except for **grohtml**, which ignores indentation.

.AT [*system* [*release*]]
: Alter the footer for use with AT&T *man pages*. This command exists only for compatibility; don't use it. See the *groff* info manual for more.

.BT
: Print the footer string. 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.

.PD [*nnn*]
: Adjust the empty space before a new paragraph or section. The optional argument gives the amount of space (default unit is 'v'); without parameter, the value is reset to its default value (1 line in nroff mode, 0.4v otherwise). This affects the macros **SH**, **SS**, **TP**, **LP** (resp. **PP** and **P**), **IP**, and **HP**.

  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
: Print the header string. Redefine this macro to get control of the header.

.UC [*version*]
: Alter the footer for use with BSD *man pages*. This command exists only for compatibility; don't use it. See the *groff* info manual for more.

The following strings are defined:

***R** The ‘registered’ sign.

***S** Switch back to the default font size.

***(lq**
***(rq** Left and right quote. This is equal to ‘\(lq’ and ‘\(rq’, respectively.

***(HF** The typeface used to print headings and subheadings. The default is ‘B’.

***(Tm** The ‘trademark’ sign.

If a preprocessor like **tbl** or **eqn** is needed, it has become common to make the first line of the *man page* look like this:

'\" *word*

Note the single space character after the double quote. *word* consists of letters for the needed preprocessors: ‘e’ for **eqn**, ‘r’ for **refer**, and ‘t’ for **tbl**. Modern implementations of the **man** program read this first line and automatically call the right preprocessor(s).

PORTABILITY AND TROFF REQUESTS top

Since the **man** macros consist of groups of *groff* requests, one can, in principle, supplement the functionality of the **man** macros with individual *groff* requests where necessary. See the *groff* info pages for a complete reference of all requests.

Note, however, that using raw troff requests is likely to make your page render poorly on the class of viewers that transform it to HTML. Troff requests make implicit assumptions about things like character and page sizes that may break in an HTML environment; also, many of these viewers don't interpret the full troff 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 in the requests described on this page. Further, it is best to completely avoid those we have described as ‘presentation-level’ (**.HP**, **.PD**, and **.DT**).

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 onto your page.

In a way similar to using *groff* requests, it is possible to use the facilities documented in the ESCAPE SEQUENCES section of the groff(7) manual page and in the groff_char(7) manual page. Regarding portability, similar caveats apply as with respect to *groff* requests.

Some escape sequences are however required for correct typesetting even in manual pages and usually do not cause portability problems:

"\ " Unpaddable non-breaking space character. (The double-quotes are to make the presence of the space character clear in this document, and are not necessary in the input file.) Useful for preventing breaking between a numerical quantity and its corresponding unit(s), for instance:

```
There are 2.54\ cm in an inch, and 1,024\ bytes in 1\ kiB.
```

\& 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. Useful 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 on macro lines 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. Used as a punctuation mark for an interruption in a sentence—like in this one.

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

\(ga ASCII grave accent. Useful 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. Useful 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. Useful 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, for example:

```
.BR "dd if" =\c
.I file
```

Alternatively, and perhaps with better portability, the **\f** font escape sequence can be used; see below.

Attempting to use **\c** to include the output from more than one macro line into the head of a **.TP** macro will misrender with groff-1.22.3, mandoc-1.14.1, older versions of these programs, and perhaps with some other formatters.

\e Widely used in manual pages to represent a backslash output glyph. It works reliably as long as the **.ec** request is not used, which should never happen in manual 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. This is needed when three different fonts are required on a single input line, for example:

```
.TP
\fBif\fP=\fIfile\fP
```

It can also be used if three different fonts are needed in a single word. It may be more portable than **\c**.

As long as only two fonts are needed, using font alternation macros like **.BR** usually results in more readable source code.

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

FILES top

man.tmac
an.tmac
These are wrapper files to call **andoc.tmac**.

andoc.tmac
Use this file in case you don't know whether the **man** macros or the **mdoc** package should be used. Multiple man pages (in either format) can be handled.

an-old.tmac
Most **man** macros are contained in this file.

an-ext.tmac
The extension macro definitions for **.SY**, **.OP**, **.YS**, **.TQ**, **.EX/.EE**, **.UR/.UE**, and **.MT/.ME** are contained in this file. It is written in classic troff, and released for free re-use, and not copylefted; manual page authors concerned about portability to legacy Unix systems are encouraged to copy these definitions into their pages, and maintainers of troff or its workalikes are encouraged to re-use them.

Note that the definitions for these macros are read after the call of **TH**, so they will replace macros of the same names given at the beginning of your file. If you must use your own definitions for these macros, they must be given after calling **TH**.

man.local
Local changes and customizations should be put into this file.

AUTHORS top

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. The extension macros were documented (and partly designed) by Eric S. Raymond; he also originated the portability advice.

SEE ALSO top

tbl(1), eqn(1), refer(1), man(1), man(7), **groff_mdoc**(7)

COLOPHON top

This page is part of the *groff* (GNU troff) project. Information about the project can be found at ⟨http://www.gnu.org/software/groff/⟩. If you have a bug report for this manual page, see ⟨http://www.gnu.org/software/groff/⟩. This page was obtained from the project's upstream Git repository

(https://git.savannah.gnu.org/git/groff.git) on 2017-11-25. (At that time, the date of the most recent commit that was found in the repository was 2017-11-21.) If you discover any rendering problems in this HTML version of the page, or you believe there is a better or more up-to-date source for the page, or you have corrections or improvements to the information in this COLOPHON (which is *not* part of the original manual page), send a mail to man-pages@man7.org

Groff Version 1.22.3 24 November 2017 GROFF_MAN(7)

Pages that refer to this page: groff(1), groff_tmac(5), man(7), man-pages(7)
