



Rocky Enterprise Linux 9.2 Manual Pages on command 'man.7'

C:~>man man.7

MAN(7) Linux Programmer's Manual MAN(7)

NAME

man - macros to format man pages

SYNOPSIS

groff -Tascii -man file ...

groff -Tps -man file ...

man [section] title

DESCRIPTION

This manual page explains the groff an.tmac macro package (often called the man macro package). This macro package should be used by developers when writing or porting man pages for Linux. It is fairly compatible with other versions of this macro package, so porting man pages should not be a major problem (exceptions include the NET-2 BSD release, which uses a totally different macro package called mdoc; see mdoc(7)).

Note that NET-2 BSD mdoc man pages can be used with groff simply by specifying the -mdoc option instead of the -man option. Using the -mandoc option is, however, recommended, since this will automatically detect which macro package is in use.

For conventions that should be employed when writing man pages for the Linux man-pages package, see man-pages(7).

Title line

The first command in a man page (after comment lines, that is, lines that start with .") should be

.TH title section date source manual

For details of the arguments that should be supplied to the TH command, see man-pages(7).

Note that BSD mdoc-formatted pages begin with the Dd command, not the TH command.

Sections

Sections are started with .SH followed by the heading name.

The only mandatory heading is NAME, which should be the first section and be followed on the next line by a one-line description of the program:

```
.SH NAME
```

```
item \- description
```

It is extremely important that this format is followed, and that there is a backslash before the single dash which follows the item name. This syntax is used by the mandb(8) program to create a database of short descriptions for the whatis(1) and apropos(1) commands. (See lexgrog(1) for further details on the syntax of the NAME section.)

For a list of other sections that might appear in a manual page, see man-pages(7).

Fonts

The commands to select the type face are:

.B Bold

.BI Bold alternating with italics (especially useful for function specifications)

.BR Bold alternating with Roman (especially useful for referring to other manual pages)

.I Italics

.IB Italics alternating with bold

.IR Italics alternating with Roman

.RB Roman alternating with bold

.RI Roman alternating with italics

.SB Small alternating with bold

.SM Small (useful for acronyms)

Traditionally, each command can have up to six arguments, but the GNU implementation removes this limitation (you might still want to limit yourself to 6 arguments for portability's sake). Arguments are delimited by spaces. Double quotes can be used to specify an argument which contains spaces. All of the arguments will be

printed next to each other without intervening spaces, so that the .BR command can be used to specify a word in bold followed by a mark of punctuation in Roman. If no arguments are given, the command is applied to the following line of text.

Other macros and strings

Below are other relevant macros and predefined strings. Unless noted otherwise, all macros cause a break (end the current line of text). Many of these macros set or use the "prevailing indent." The "prevailing indent" value is set by any macro with the parameter *i* below; macros may omit *i* in which case the current prevailing indent will be used. As a result, successive indented paragraphs can use the same indent without respecifying the indent value. A normal (nonindented) paragraph resets the prevailing indent value to its default value (0.5 inches). By default, a given indent is measured in ens; try to use ens or ems as units for indents, since these will automatically adjust to font size changes. The other key macro definitions are:

Normal paragraphs

- .LP Same as .PP (begin a new paragraph).
- .P Same as .PP (begin a new paragraph).
- .PP Begin a new paragraph and reset prevailing indent.

Relative margin indent

- .RS *i* Start relative margin indent: moves the left margin *i* to the right (if *i* is omitted, the prevailing indent value is used). A new prevailing indent is set to 0.5 inches. As a result, all following paragraph(s) will be indented until the corresponding .RE.
- .RE End relative margin indent and restores the previous value of the prevailing indent.

Indented paragraph macros

- .HP *i* Begin paragraph with a hanging indent (the first line of the paragraph is at the left margin of normal paragraphs, and the rest of the paragraph's lines are indented).
- .IP *x i* Indented paragraph with optional hanging tag. If the tag *x* is omitted, the entire following paragraph is indented by *i*. If the tag *x* is provided, it is hung at the left margin before the following indented paragraph (this is just like .TP except the tag is included with the command

instead of being on the following line). If the tag is too long, the text after the tag will be moved down to the next line (text will not be lost or garbled). For bulleted lists, use this macro with `\(bu` (bullet) or `\(em` (em dash) as the tag, and for numbered lists, use the number or letter followed by a period as the tag; this simplifies translation to other formats.

`.TP i` Begin paragraph with hanging tag. The tag is given on the next line, but its results are like those of the `.IP` command.

Hypertext link macros

`.UR url`

Insert a hypertext link to the URI (URL) `url`, with all text up to the following `.UE` macro as the link text.

`.UE [trailer]`

Terminate the link text of the preceding `.UR` macro, with the optional trailer (if present, usually a closing parenthesis and/or end-of-sentence punctuation) immediately following. For non-HTML output devices (e.g., `man-Tutf8`), the link text is followed by the URL in angle brackets; if there is no link text, the URL is printed as its own link text, surrounded by angle brackets. (Angle brackets may not be available on all output devices.) For the HTML output device, the link text is hyperlinked to the URL; if there is no link text, the URL is printed as its own link text.

These macros have been supported since GNU Troff 1.20 (2009-01-05) and Heirloom Doctools Troff since 160217 (2016-02-17).

Miscellaneous macros

`.DT` Reset tabs to default tab values (every 0.5 inches); does not cause a break.

`.PD d` Set inter-paragraph vertical distance to `d` (if omitted, `d=0.4v`); does not cause a break.

`.SS t` Subheading `t` (like `.SH`, but used for a subsection inside a section).

Predefined strings

The `man` package has the following predefined strings:

`*R` Registration Symbol: ?

`*S` Change to default font size

*(Tm Trademark Symbol: ?

*(lq Left angled double quote: ?

*(rq Right angled double quote: ?

Safe subset

Although technically `man` is a troff macro package, in reality a large number of other tools process `man` page files that don't implement all of troff's abilities.

Thus, it's best to avoid some of troff's more exotic abilities where possible to permit these other tools to work correctly. Avoid using the various troff preprocessors (if you must, go ahead and use `tbl(1)`, but try to use the IP and TP commands instead for two-column tables). Avoid using computations; most other tools can't process them. Use simple commands that are easy to translate to other formats. The following troff macros are believed to be safe (though in many cases they will be ignored by translators): `\`, `..`, `ad`, `bp`, `br`, `ce`, `de`, `ds`, `el`, `ie`, `if`, `fi`, `ft`, `hy`, `ig`, `in`, `na`, `ne`, `nf`, `nh`, `ps`, `so`, `sp`, `ti`, `tr`.

You may also use many troff escape sequences (those sequences beginning with `\`).

When you need to include the backslash character as normal text, use `\e`. Other sequences you may use, where `x` or `xx` are any characters and `N` is any digit, include:

`\'`, `\``, `\-`, `\.`, `\"`, `\%`, `*x`, `*(xx`, `\(xx`, `\$N`, `\nx`, `\n(xx`, `\fx`, and `\f(xx`. Avoid using the escape sequences for drawing graphics.

Do not use the optional parameter for `bp` (break page). Use only positive values for `sp` (vertical space). Don't define a macro (`de`) with the same name as a macro in this or the `mdoc` macro package with a different meaning; it's likely that such redefinitions will be ignored. Every positive indent (`in`) should be paired with a matching negative indent (although you should be using the `RS` and `RE` macros instead). The condition test (`if,ie`) should only have 't' or 'n' as the condition.

Only translations (`tr`) that can be ignored should be used. Font changes (`ft` and the `\f` escape sequence) should only have the values 1, 2, 3, 4, R, I, B, P, or CW (the `ft` command may also have no parameters).

If you use capabilities beyond these, check the results carefully on several tools.

Once you've confirmed that the additional capability is safe, let the maintainer of this document know about the safe command or sequence that should be added to this list.

/usr/share/groff/[*/]tmac/an.tmac

/usr/man/whatis

NOTES

By all means include full URLs (or URIs) in the text itself; some tools such as `man2html(1)` can automatically turn them into hypertext links. You can also use the `UR` and `UE` macros to identify links to related information. If you include URLs, use the full URL (e.g., `?http://www.kernel.org?`) to ensure that tools can automatically find the URLs.

Tools processing these files should open the file and examine the first nonwhite space character. A period (.) or single quote (') at the beginning of a line indicates a troff-based file (such as `man` or `mdoc`). A left angle bracket (<) indicates an SGML/XML-based file (such as HTML or Docbook). Anything else suggests simple ASCII text (e.g., a "catman" result).

Many `man` pages begin with `\` followed by a space and a list of characters, indicating how the page is to be preprocessed. For portability's sake to non-troff translators we recommend that you avoid using anything other than `tbl(1)`, and Linux can detect that automatically. However, you might want to include this information so your `man` page can be handled by other (less capable) systems. Here are the definitions of the preprocessors invoked by these characters:

`e` `eqn(1)`

`g` `grap(1)`

`p` `pic(1)`

`r` `refer(1)`

`t` `tbl(1)`

`v` `vgrind(1)`

BUGS

Most of the macros describe formatting (e.g., font type and spacing) instead of marking semantic content (e.g., this text is a reference to another page), compared to formats like `mdoc` and `DocBook` (even HTML has more semantic markings). This situation makes it harder to vary the `man` format for different media, to make the formatting consistent for a given media, and to automatically insert cross-references. By sticking to the safe subset described above, it should be easier to automate transitioning to a different reference page format in the future.

The Sun macro TX is not implemented.

SEE ALSO

`apropos(1)`, `groff(1)`, `lexgrog(1)`, `man(1)`, `man2html(1)`, `groff_mdoc(7)`, `whatis(1)`,
`groff_man(7)`, `groff_www(7)`, `man-pages(7)`, `mdoc(7)`

COLOPHON

This page is part of release 5.05 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.

Linux

2019-03-06

MAN(7)