

NAME

help format – help file format conventions and syntax

INTRODUCTION

This document describes the formatting syntax and conventions used with the `spec` and C-PLOT help files. The formatting syntax uses the conventions of the traditional UNIX text formatters (*nroff*, *troff*, *ditroff*, etc.), although only a small subset of the formatting directives are supported by the `spec` and C-PLOT built-in formatting. The supported directives are described below.

The distributions of `spec` and C-PLOT include utilities for preparing both PostScript and HTML (hyper text markup language) versions of the help files using either the standard *ditroff* or the GNU *groff* packages.

FORMATTING COMMANDS

Input text that has a dot on the first line is a formatting command. Unrecognized commands are ignored in `spec` and C-PLOT.

The following commands are supported:

- .\" – Begins a comment. The rest of the text on the line is ignored.
- .bp – Begins a new page.
- .BP – Same as above.
- .ne *N* – Indicates *N* lines are needed on the current page. If there isn't enough room on the current page, a new page is started.
- .sp *N* – spaces *N* lines. Without an argument, one line is skipped. The argument may contain a decimal point and a unit indicator for the benefit of *nroff/troff*, although for `spec` and C-PLOT, only the integer part of the argument is used.
- .ta *N* – Start words every *N* letters, somewhat like using tabs.
- .br – Causes a line break. The current line is finished and a new line is started.
- .nf – Switch to no-fill mode. Each newline in the input causes a newline in the output.
- .fi – Switch to fill mode. Words are collected from the input until a full line of output is available.
- .TH *name category date SCCS_info* – Should be on the first line of a help file. The arguments are used in the page header (and footer, when printed) and are, respectively, the name of the help file, the category (such as macro, function, reference, etc.), the date the file was last revised and source-code control information.
- .SH – Does a subhead. There can't be more than six arguments, although you can use double quotes to put more than one word in an argument. Subheads are placed all the way to the left of the page.
- .HP *N* – Starts a hanging paragraph. The optional argument *N* specifies the indent level. The default is 0.
- .hp *N* – Can be used within a `.HP` block to increase or decrease the indent. The optional argument *N* specifies the indent level. This command will skip a line.
- .PP – Starts a paragraph.
- .LP – Same as above.
- .(L *F N I* – Starts a no-fill block. The optional argument *F* indicates the font to use and may be R, I, C, B or O. (See below for a description of the font code letters.) The optional argument *N* is the number of lines needed. If there is not enough room to

print that many lines on the current page, a new page will be started. The optional argument *I* is the number of spaces of extra indentation to add.

- .)L – Ends the no-fill block started by .(L.
- .Q – Put quotation marks around arguments. If no arguments, put quotation marks around text on next input line.
- .NE – Only relevant for screen displays, this command indicates the screen should not be erased before displaying the first page.
- # – Starts a comment if its the first character on a line. The rest of the text on the line is ignored. This comment notation is only for files that will not be run off using *nroff/troff*.
- .BL – Begin a list of items.
- .EL – Ends a list of items.
- .LI *item* Adds *item* to a list of items. For HTML documents, one list items appears per line of output. For non-HTML documents, list items are separated by tabs, and will fill the row according to the current tab stops.
- .DL + or .DL – – For HTML output, indicates the start or end of a definition block.
- .DD *text* – For HTML output, indicates the start of a definition. For the other formats, the optional argument is simply copied verbatim.
- .HR *text* [*C* [*URL*]] – For HTML output, creates a hypertext link. If one or two arguments are present, names the link *text.html* If the second argument is *C*, outputs *text* in Courier font. If the third argument is present, uses that as the URL link. For non-HTML formats, the text is copied verbatim in the indicated font.

FONT CONTROL

The help files use the following fonts for the purposes indicated. In the printed documentation, the New Century Schoolbook typeface is used for the Roman, italic and bold fonts. When used with *spec* and *C-PLOT*, the directives to switch to bold, italic or one of the Courier typefaces result in highlighted text on a video terminal.

Roman (R) – Used for normal text.

Italic (I) – Used for file and path names, including system commands, along with book titles, etc.

Bold (B) – Used for section heads. Occasionally used for emphasis. Sometimes used to display arguments to system commands.

Courier (C) – Used to represent literal text that might be output by the computer program or might be typed by a user at the keyboard.

Courier Bold (CB) – Occasionally used to distinguish text a user types from text output by the computer.

Courier Oblique (O) – Used to represent variable computer text, such as the argument to a function.

“Quoted Text” (Q) – Not really a font, but used to place double quotes around a bit of text.

All the fonts, except Roman, display as highlighted text on the video terminal.

The formatting commands:

```
.B .C .I .O
```

put their arguments in the associated font. If there are no arguments to the above commands, the next line of input is put in that font.

The formatting commands:

```
.RB .BR .IR .CR .OR .CB
.RI .BI .IB .CI .OC
.RC .IC .CO
.RO
```

place the first argument in the first font, the second argument in the second font and append an optional third argument in the regular text font, except for `.CB`, which uses Courier Bold and Roman for the two fonts. The space characters between the arguments are removed. Use double quotes to include space characters within arguments. A space character is added after the last argument. The formatting commands:

```
.rb .ir
.ri .ci .oc
.rc .ic .co
.ro
```

work as above, except a space characters is not appended after the last argument.

The `@` symbol is used in the help file sources to represent literal double quote characters, which makes it easier to insert literal double quote characters into formatting command arguments. To enter a literal `@` use both a backslash and the troff `.tr` directive, as in

```
.tr @@
a literal \@ appears
.tr @"
```

SPECIAL CHARACTER SEQUENCES

The following special character sequences are recognized:

<code>\"</code>	- A double quote
<code>\&</code>	- A zero-width space
<code>\0</code>	- A digit-width space
<code>\ </code>	- A sixth-em space (no space on screen)
<code>\(**</code>	- An asterisk (*)
<code>\(hy</code>	- A hyphen (-)
<code>\(mu</code>	- The multiplication sign (×)
<code>\(space)</code>	- An unpaddingable space
<code>*(CP</code>	- The string C-PLOT
<code>*(SP</code>	- The string SPEC
<code>*(UN</code>	- The string UNIX
<code>\-</code>	- A minus sign (-)
<code>\@</code>	- An @ sign
<code>\</code>	- A back slash
<code>\e</code>	- A back slash
<code>\f1</code>	- Switch to font R
<code>\fC</code>	- Switch to font C
<code>\fP</code>	- Switch to previous font
<code>\f*(#C</code>	- Switch to font C
<code>\f*(#I</code>	- Switch to font I
<code>\f*(#O</code>	- Switch to font O

HINTS

Since the `.if` command is not recognized by the `spec` and `C-PLOT` built-in formatter, use the command `.if t ...` to insert formatting directives to be executed only when the help files are being printed with *troff*.