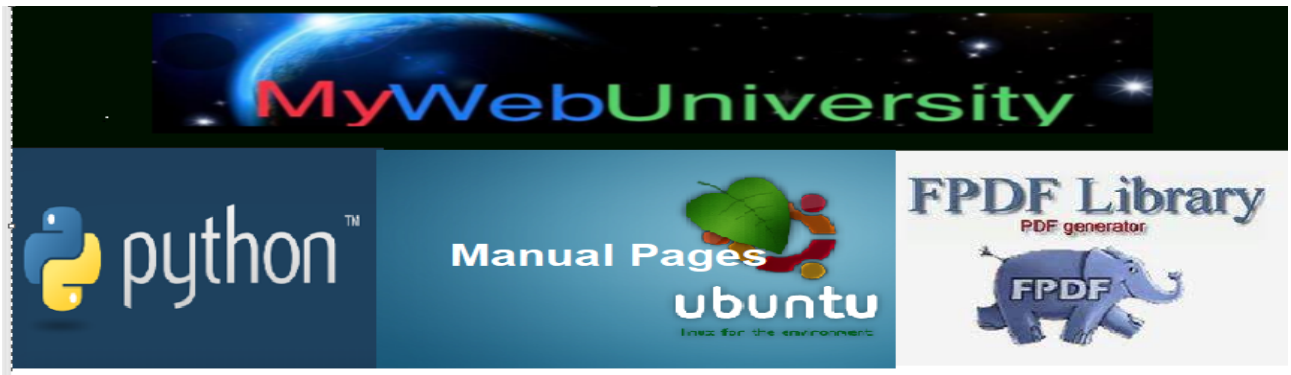

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'pod2text.1'

$ man pod2text.1

POD2TEXT(1) Perl Programmers Reference Guide POD2TEXT(1)

NAME

pod2text - Convert POD data to formatted ASCII text

SYNOPSIS

pod2text [-aclostu] [--code] [-e encoding]

[--errors=style] [--guesswork=rule[,rule...]]

[-i indent] [-q quotes]

[--nourls] [--stderr] [-w width] [input [output ...]]

pod2text -h

DESCRIPTION

pod2text is a wrapper script around the Pod::Text and its subclasses. It uses them to generate formatted text from POD source. It can optionally use either termcap sequences or ANSI color escape sequences to format the text.

input is the file to read for POD source (the POD can be embedded in code). If input isn't given, it defaults to "STDIN". output, if given, is the file to which to write the formatted output. If output isn't given, the formatted output is written to "STDOUT".

Several POD files can be processed in the same pod2text invocation (saving module load and compile times) by providing multiple pairs of input and output files on the command line.

By default, the output encoding is the same as the encoding of the input file, or UTF-8 if that encoding is not set (except on EBCDIC systems). See the -e option to explicitly set the output encoding and "Encoding" in Pod::Text for more discussion.

OPTIONS

Each option is annotated with the version of podlators in which that option was added with

its current meaning.

-a, --alt

[1.00] Use an alternate output format that, among other things, uses a different heading style and marks "=item" entries with a colon in the left margin.

--code

[1.11] Include any non-POD text from the input file in the output as well. Useful for viewing code documented with POD blocks with the POD rendered and the code left intact.

-c, --color

[1.00] Format the output with ANSI color escape sequences. Using this option requires that Term::ANSIColor be installed on your system.

-e encoding, --encoding=encoding

[5.00] Specifies the encoding of the output. encoding must be an encoding recognized by the Encode module (see Encode::Supported). If the output contains characters that cannot be represented in this encoding, that is an error that will be reported as configured by the "errors" option. If error handling is other than "die", the unrepresentable character will be replaced with the Encode substitution character (normally "?").

WARNING: The input encoding of the POD source is independent from the output encoding, and setting this option does not affect the interpretation of the POD input. Unless your POD source is US-ASCII, its encoding should be declared with the "=encoding" command in the source, as near to the top of the file as possible. If this is not done, Pod::Simple will will attempt to guess the encoding and may be successful if it's Latin-1 or UTF-8, but it will produce warnings. See perlpod(1) for more information.

--errors=style

[2.5.0] Set the error handling style. "die" says to throw an exception on any POD formatting error. "stderr" says to report errors on standard error, but not to throw an exception. "pod" says to include a POD ERRORS section in the resulting documentation summarizing the errors. "none" ignores POD errors entirely, as much as possible.

The default is "die".

--guesswork=rule[,rule...]

[5.01] By default, pod2text applies some default formatting rules based on guesswork and regular expressions that are intended to make writing Perl documentation easier and require less explicit markup. These rules may not always be appropriate, particularly

for documentation that isn't about Perl. This option allows turning all or some of it off.

The special rule "all" enables all guesswork. This is also the default for backward compatibility reasons. The special rule "none" disables all guesswork. Otherwise, the value of this option should be a comma-separated list of one or more of the following keywords:

quoting

If no guesswork is enabled, any text enclosed in C<> is surrounded by double quotes in nroff (terminal) output unless the contents are already quoted. When this guesswork is enabled, quote marks will also be suppressed for Perl variables, function names, function calls, numbers, and hex constants.

Any unknown guesswork name is silently ignored (for potential future compatibility), so be careful about spelling.

-i indent, --indent=indent

[1.00] Set the number of spaces to indent regular text, and the default indentation for "=over" blocks. Defaults to 4 spaces if this option isn't given.

-h, --help

[1.00] Print out usage information and exit.

-l, --loose

[1.00] Print a blank line after a "=head1" heading. Normally, no blank line is printed after "=head1", although one is still printed after "=head2", because this is the expected formatting for manual pages; if you're formatting arbitrary text documents, using this option is recommended.

-m width, --left-margin=width, --margin=width

[1.24] The width of the left margin in spaces. Defaults to 0. This is the margin for all text, including headings, not the amount by which regular text is indented; for the latter, see -i option.

--nourls

[2.5.0] Normally, L<> formatting codes with a URL but anchor text are formatted to show both the anchor text and the URL. In other words:

L<foo|http://example.com/>

is formatted as:

foo <http://example.com/>

This flag, if given, suppresses the URL when anchor text is given, so this example would be formatted as just "foo". This can produce less cluttered output in cases where the URLs are not particularly important.

-o, --overstrike

[1.06] Format the output with overstrike printing. Bold text is rendered as character, backspace, character. Italics and file names are rendered as underscore, backspace, character. Many pagers, such as less, know how to convert this to bold or underlined text.

-q quotes, --quotes=quotes

[4.00] Sets the quote marks used to surround C<> text to quotes. If quotes is a single character, it is used as both the left and right quote. Otherwise, it is split in half, and the first half of the string is used as the left quote and the second is used as the right quote.

quotes may also be set to the special value "none", in which case no quote marks are added around C<> text.

-s, --sentence

[1.00] Assume each sentence ends with two spaces and try to preserve that spacing. Without this option, all consecutive whitespace in non-verbatim paragraphs is compressed into a single space.

--stderr

[2.1.3] By default, pod2text dies if any errors are detected in the POD input. If --stderr is given and no --errors flag is present, errors are sent to standard error, but pod2text does not abort. This is equivalent to "--errors=stderr" and is supported for backward compatibility.

-t, --termcap

[1.00] Try to determine the width of the screen and the bold and underline sequences for the terminal from termcap, and use that information in formatting the output. Output will be wrapped at two columns less than the width of your terminal device. Using this option requires that your system have a termcap file somewhere where Term::Cap can find it and requires that your system support termios. With this option, the output of pod2text will contain terminal control sequences for your current terminal type.

-u, --utf8

[2.2.0] Set the output encoding to UTF-8. This is equivalent to "--encoding=UTF-8" and

is supported for backward compatibility.

-w, --width=width, -width

[1.00] The column at which to wrap text on the right-hand side. Defaults to 76, unless -t is given, in which case it's two columns less than the width of your terminal device.

EXIT STATUS

As long as all documents processed result in some output, even if that output includes errata (a "POD ERRORS" section generated with "--errors=pod"), pod2text will exit with status 0. If any of the documents being processed do not result in an output document, pod2text will exit with status 1. If there are syntax errors in a POD document being processed and the error handling style is set to the default of "die", pod2text will abort immediately with exit status 255.

DIAGNOSTICS

If pod2text fails with errors, see Pod::Text and Pod::Simple for information about what those errors might mean. Internally, it can also produce the following diagnostics:

-c (--color) requires Term::ANSIColor be installed

(F) -c or --color were given, but Term::ANSIColor could not be loaded.

Unknown option: %s

(F) An unknown command line option was given.

In addition, other Getopt::Long error messages may result from invalid command-line options.

ENVIRONMENT

COLUMNS

If -t is given, pod2text will take the current width of your screen from this environment variable, if available. It overrides terminal width information in TERMCAP.

TERMCAP

If -t is given, pod2text will use the contents of this environment variable if available to determine the correct formatting sequences for your current terminal device.

AUTHOR

Russ Allbery <rra@cpan.org>.

COPYRIGHT AND LICENSE

SEE ALSO

Encode::Supported, Pod::Text, Pod::Text::Color, Pod::Text::Overstrike, Pod::Text::Termcap, Pod::Simple, perlpod(1)

The current version of this script is always available from its web site at <https://www.eyrie.org/~eagle/software/podlators/>. It is also part of the Perl core distribution as of 5.6.0.