NAME

Pod::Text - Convert POD data to formatted ASCII text

SYNOPSIS

```
use Pod::Text;
my $parser = Pod::Text->new (sentence => 0, width => 78);

# Read POD from STDIN and write to STDOUT.
$parser->parse_from_filehandle;

# Read POD from file.pod and write to file.txt.
$parser->parse_from_file ('file.pod', 'file.txt');
```

DESCRIPTION

Pod::Text is a module that can convert documentation in the POD format (the preferred language for documenting Perl) into formatted ASCII. It uses no special formatting controls or codes whatsoever, and its output is therefore suitable for nearly any device.

As a derived class from Pod::Parser, Pod::Text supports the same methods and interfaces. See *Pod::Parser* for all the details; briefly, one creates a new parser with `Pod::Text->new()` and then calls either parse_from_filehandle() or parse_from_file().

new() can take options, in the form of key/value pairs, that control the behavior of the parser. The currently recognized options are:

alt

If set to a true value, selects an alternate output format that, among other things, uses a different heading style and marks `=item` entries with a colon in the left margin. Defaults to false.

code

If set to a true value, the non-POD parts of the input file will be included in the output. Useful for viewing code documented with POD blocks with the POD rendered and the code left intact.

indent

The number of spaces to indent regular text, and the default indentation for `=over` blocks. Defaults to 4.

loose

If set to a true value, a blank line is printed after a `=head1` heading. If set to false (the default), no blank line is printed after `=head1`, although one is still printed after `=head2`. This is the default because it's the expected formatting for manual pages; if you're formatting arbitrary text documents, setting this to true may result in more pleasing output.

margin

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 the *indent* option. To set the right margin, see the *width* option.

quotes

Sets the quote marks used to surround C<> text. If the value is a single character, it is used as both the left and right quote; if it is two characters, the first character is used as the left quote and the second as the right quoted; and if it is four characters, the first two are used as the left quote and the second two as the right quote.

This may also be set to the special value `none`, in which case no quote marks are added

around C<> text.

sentence

If set to a true value, Pod::Text will assume that each sentence ends in two spaces, and will try to preserve that spacing. If set to false, all consecutive whitespace in non-verbatim paragraphs is compressed into a single space. Defaults to true.

width

The column at which to wrap text on the right-hand side. Defaults to 76.

The standard Pod::Parser method parse_from_filehandle() takes up to two arguments, the first being the file handle to read POD from and the second being the file handle to write the formatted output to. The first defaults to STDIN if not given, and the second defaults to STDOUT. The method parse_from_file() is almost identical, except that its two arguments are the input and output disk files instead. See *Pod::Parser* for the specific details.

DIAGNOSTICS

Bizarre space in item

Item called without tag

(W) Something has gone wrong in internal `=item` processing. These messages indicate a bug in Pod::Text; you should never see them.

Can't open %s for reading: %s

(F) Pod::Text was invoked via the compatibility mode pod2text() interface and the input file it was given could not be opened.

Invalid quote specification "%s"

(F) The quote specification given (the quotes option to the constructor) was invalid. A quote specification must be one, two, or four characters long.

%s:%d: Unknown command paragraph: %s

(W) The POD source contained a non-standard command paragraph (something of the form `=command args`) that Pod::Man didn't know about. It was ignored.

%s:%d: Unknown escape: %s

(W) The POD source contained an E<> escape that Pod::Text didn't know about.

%s:%d: Unknown formatting code: %s

(W) The POD source contained a non-standard formatting code (something of the form X<>) that Pod::Text didn't know about.

%s:%d: Unmatched =back

(W) Pod::Text encountered a `=back` command that didn't correspond to an `=over` command.

RESTRICTIONS

Embedded Ctrl-As (octal 001) in the input will be mapped to spaces on output, due to an internal implementation detail.

NOTES

This is a replacement for an earlier Pod::Text module written by Tom Christiansen. It has a revamped interface, since it now uses Pod::Parser, but an interface roughly compatible with the old Pod::Text::pod2text() function is still available. Please change to the new calling convention, though.

The original Pod::Text contained code to do formatting via termcap sequences, although it wasn't turned on by default and it was problematic to get it to work at all. This rewrite doesn't even try to do that, but a subclass of it does. Look for *Pod::Text::Termcap*.

SEE ALSO

Pod::Parser, *Pod::Text::Termcap*, *pod2text(1)*

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

AUTHOR

Russ Allbery <rra@stanford.edu>, based *very* heavily on the original Pod::Text by Tom Christiansen <tchrist@mox.perl.com> and its conversion to Pod::Parser by Brad Appleton <bradapp@enteract.com>.

COPYRIGHT AND LICENSE