
Pod::Pandoc Documentation

Release 0.5.0

Jakob Voß

May 11, 2018

Contents

1	DESCRIPTION	1
2	OUTLINE	3
3	REQUIREMENTS	5
4	USAGE EXAMPLES	7
4.1	Replace pod2html	7
4.2	Replace pod2markdown	7
4.3	GitHub wiki	7
4.4	Sphinx and Read The Docs	7
5	SEE ALSO	9
6	AUTHOR	11
7	CONTRIBUTORS	13
8	COPYRIGHT AND LICENSE	15
8.1	App::pod2pandoc	15
8.2	Pod::Pandoc::Modules	16
8.3	Pod::Simple::Pandoc	17
8.4	pod2pandoc	19

CHAPTER 1

DESCRIPTION

The Plain Old Documentation format (Pod) is a markup language used to document Perl code (see [perlpod](#) for reference). Several Perl modules exist to process and convert Pod into other formats.

Pod::Pandoc is an attempt to unify and extend Pod converting based on the [Pandoc](#) document converter. Pandoc supports more document formats in a more detailed and uniform way than any set of Perl modules will ever do. For this reason Pod::Pandoc provides methods to convert Pod to the Pandoc document model for further processing with Pandoc.

CHAPTER 2

OUTLINE

- *pod2pandoc* is a command line script to convert Pod to any format supported by Pandoc.
- *App::pod2pandoc* provides functionality of *pod2pandoc* to be used in Perl code.
- *Pod::Simple::Pandoc* converts Pod to the abstract document model of Pandoc.
- *Pod::Pandoc::Modules* manages a set of Pod documents of Perl modules.

CHAPTER 3

REQUIREMENTS

Installation of this module does not require Pandoc but it is needed to make actual use of it. See <http://pandoc.org/installing.html> for installation.

USAGE EXAMPLES

4.1 Replace pod2html

```
# pod2html --infile=input.pm --css=style.css --title=TITLE > output.html
pod2pandoc input.pm --css=style.css --toc --name -o output.html
```

Pandoc option `--toc` corresponds to pod2html option `--index` and is disabled by default. pod2pandoc adds title and subtitle from NAME section.

4.2 Replace pod2markdown

```
# pod2markdown input.pod
pod2pandoc input.pod -t markdown

# pod2markdown input.pod output.md
pod2pandoc input.pod -o output.md
```

4.3 GitHub wiki

The [GitHub wiki](#) of this project is automatically populated with its module documentation. Wiki pages are created with `pod2pandoc` as following (see script `update-wiki.sh`):

```
pod2pandoc lib/ script/ wiki/ --ext md --index Home --wiki -t markdown_github
```

4.4 Sphinx and Read The Docs

The [Sphinx documentation generator](#) recommends documents in reStructureText format. It further requires a configuration file `conf.py` and some links need to be adjusted because Pandoc does not support wikilinks in rst output format (see script `update-docs.sh`):

```
pod2pandoc lib/ script/ docs/ --ext rst --wiki -t rst --standalone
perl -pi -e 's!`([^`]+) <([^>]+)>`__!-e "docs/$2.rst" ? ":doc:`$1 <$2>`" : "`$1 <
↪$2>`__"!e' docs/*.rst
make -C docs html
```

The result is published automatically at <http://pod-pandoc.rtfid.io/en/latest/Pod-Pandoc.html>.

CHAPTER 5

SEE ALSO

This module is based on the wrapper module `Pandoc` to execute `pandoc` from Perl and on the module `Pandoc::Elements` for `pandoc` document processing.

This module makes obsolete several specialized `Pod::Simple::...` modules such as `Pod::Simple::HTML`, `Pod::Simple::XHTML`, `Pod::Simple::LaTeX`, `Pod::Simple::RTFPod::Simple::Text`, `Pod::Simple::Wiki`, `Pod::WordML`, `Pod::Perldoc::ToTocetc`. It also covers batch conversion such as `Pod::Simple::HTMLBatch`, `Pod::ProjectDocs`, `Pod::POM::Web`, and `Pod::HtmlTree`.

CHAPTER 6

AUTHOR

Jakob Voß <jakob.voss@gbv.de>

CHAPTER 7

CONTRIBUTORS

Benct Philip Jonsson

COPYRIGHT AND LICENSE

Copyright 2017- Jakob Voß

GNU General Public License, Version 2

8.1 App::pod2pandoc

8.1.1 SYNOPSIS

```
use App::pod2pandoc;

# pod2pandoc command line script
my ($input, $opt, @args) = parse_arguments(@ARGV);
pod2pandoc($input, $opt, @args);

# parse a Perl/Pod file and print its JSON serialization
pod2pandoc( ['example.pl'], {} );

# parse a Perl/Pod file and convert to HTML with a template
pod2pandoc( ['example.pl'], {}, '--template', 'template.html' );

# process directory of Perl modules
pod2pandoc( [ lib => 'doc'], { ext => 'html' }, '--standalone' );
```

8.1.2 DESCRIPTION

This module implements the command line script *pod2pandoc*.

8.1.3 FUNCTIONS

pod2pandoc(\@input, [\%options,] \@arguments)

Processed input files with given *pod2pandoc* options (data-sections, podurl, ext, wiki, meta, update, and quiet). Additional arguments are passed to pandoc executable via module [Pandoc](#).

Input can be either files and/or module names or directories to recursively search for `.pm` and `.pod` files. If no input is specified, Pod is read from STDIN. When processing directories, the last input directory is used as output directory.

This function is exported by default.

parse_arguments(@argv)

Parses options and input arguments from given command line arguments. May terminate the program with message, for instance with argument `--help`.

8.1.4 SEE ALSO

This module is part of *Pod::Pandoc* and based on the modules *Pod::Simple::Pandoc*, *Pod::Pandoc::Modules*, *Pandoc::Element* and *Pandoc*.

8.2 Pod::Pandoc::Modules

8.2.1 SYNOPSIS

```
use Pod::Simple::Pandoc;

my $modules = Pod::Simple::Pandoc->new->parse_modules('lib');
$modules->serialize( { target => 'doc' }, '--template' => '...' ); # TODO
```

8.2.2 DESCRIPTION

Module to serialize Pod from a set of parsed Perl or Pod files. Can be configured via templates, document rewriting etc. and used with many output formats (html, markdown, and rst to be embedded in static site generators such as Jekyll).

See *Pod::Simple::Pandoc* for how to create instances of this module.

8.2.3 METHODS

add(\$name => \$doc)

Add a module given as *Pandoc::Document* unless a module of same `$name` already exists. As an exception a parsed `.pod` file will override existing entries. The document title is set to the module name if missing.

serialize ([\$dir] [, \%options] [, @args])

Serialize a set of modules into a given directory.

This method is experimental and may change!

dir Output directory.

ext Output file extension. Set to the value of `format` by default.

index Index filename (with or without extension). Set to `index` by default. Use a false value to disable index generation.

wiki Don't create subdirectories and use wiki links for references between files. instead.

update Generate target files even if source files have not been updated.

quiet Don't emit warnings and status information.

index (%options)

Create and return an index document as `Pandoc::Document`.

8.2.4 SEE ALSO

This module is part of *Pod::Pandoc*.

8.3 Pod::Simple::Pandoc

8.3.1 SYNOPSIS

```
use Pod::Simple::Pandoc;

my $parser = Pod::Simple::Pandoc->new( %options );
my $doc     = $parser->parse_file( $filename );

# result is a Pandoc::Document object
my $json = $doc->to_json;
my $markdown = $doc->to_pandoc( -t => 'markdown' );
$doc->to_pandoc( qw( -o doc.html --standalone ) );
```

8.3.2 DESCRIPTION

This module converts Pod format (`perlpod`) to the document model used by `Pandoc`. The result can be accessed with methods of `Pandoc::Elements` and further processed with `Pandoc` to convert it to other document formats (HTML, Markdown, LaTeX, PDF, EPUB, docx, ODT, man...).

See *pod2pandoc* and *App::pod2pandoc* for a command line script and a simplified API to this module.

8.3.3 OPTIONS

parse Parse Pod *data sections* with `Pandoc` and merge them into the document instead of passing them as `RawBlock`. Use `*` to parse all formats supported by `pandoc` as input format. Expects an array reference otherwise.

podurl Base URL to link Perl module names to. Set to <https://metacpan.org/pod/> by default. A false value disables linking external modules and wraps module names in `Spanelements` instead. All module names are marked up with class `perl-module`.

8.3.4 METHODS

parse_file(\$filename | *INPUT)

Reads Pod from file or filehandle and convert it to a `Pandoc::Document`. The filename is put into document metadata field `file` and the module name. The NAME section, if given, is additionally split into metadata fields `title` and `subtitle`.

`parse_module($module)`

Reads Pod from a module given by name such as "Pod::Pandoc" or by URL.

`parse_string($string)`

Reads Pod from string and convert it to a `Pandoc::Document`. Also sets metadata fields `title` and `subtitle`.

`parse_dir($directory)`

Recursively looks for `.pm` and `.pod` files in a given directory and parses them. Returns a hash reference with filenames mapped to `Pandoc::Document` objects. Each document is enriched with metadata fields `base` (relative path from each file to the base directory) in addition to `file`, `title`, and `subtitle`.

`parse_modules($directory, [quiet => 0|1])`

Same as method `parse_dir` but returns a `Pod::Simple::Pandoc::Modules` instance that maps module names to `Pandoc::Document` instances. The source directory can also be specified with option `source`. Option `quiet` disables warnings for skipped files.

`parse_and_merge(@files_or_modules)`

Reads Pod from files or modules given by name and merges them into one `Pandoc::Document` by concatenation.

8.3.5 MAPPING

Pod elements are mapped to Pandoc elements as following:

Formatting codes

Formatting codes for *italic text* (`I<...>`), **bold text** (`B<...>`), and code (`C<...>`) are mapped to Emphasized text (`Emph`), strongly emphasized text (`Strong`), and inline code (`Code`). Formatting code for filenames (`F<...>`) are mapped to inline code with class `filename`...`{.filename}` in Pandoc Markdown). Formatting codes inside code and filenames (e.g. code with `bold` or `http://example.org/` as filename) are stripped to unformatted code. Character escapes (`E<...>`) and `S<...>` are directly mapped to Unicode characters. The special formatting code `X<...>` is ignored.

Links

Some examples of links of different kinds:

`http://example.org/`

`pod2pandoc`

`pod2pandoc`

`perl(1)`

`“ENVIRONMENT” in crontab(5)`

`hell itself!`

Link text can contain formatting codes:

`pod2pandoc`

`“MAPPING”`

mapping from PoD to Pandoc

Titles may contain formatting “codes”!

Lists

1. Numbered lists are
2. converted to `NumberedList` and
 - Bulleted lists are
 - converted to `BulletList`

Definition Lists are

also supported.

=over/=back

An `=over...=back` region containing no `=item` is mapped to `BlockQuote`.

Verbatim sections

```
verbatim sections are mapped
to code blocks
```

Data sections

Data sections are passed as `RawBlock`. `HTML`, `LaTeX`, `TeX`, and `tex` are recognized as alias for `html` and `tex`.

Option `parse` can be used to parse data sections with `podoc` executable and merge them into the result document.

8.3.6 SEE ALSO

This module is based on `Pod::Simple(Pod::Simple::SimpleTree)`. It makes obsolete several specialized `Pod::Simple::...` modules such as `Pod::Simple::HTML`, `Pod::Simple::XHTML`, `Pod::Simple::LaTeX`, `Pod::Simple::RTFPod::Simple::Text`, `Pod::Simple::Wiki`, `Pod::WordML`, `Pod::Perldoc::ToToc` etc.

App::pod2pandoc implements `pod2pandoc` command line script

Pod::Pandoc process Plain Old Documentation format with Pandoc

Pod::Pandoc::Modules set of parsed documentation of Perl modules

Pod::Simple::Pandoc convert Pod to Pandoc document model

pod2pandoc convert Pod to Pandoc document model

8.4 pod2pandoc

8.4.1 SYNOPSIS

```
pod2pandoc [OPTIONS] {INPUT} [ [ -- ] PANDOC_ARGUMENTS ]

pod2pandoc Module.pm -o Module.pdf
pod2pandoc script.pl -t markdown
pod2pandoc Module::Name -o Module-Name.html
pod2pandoc lib/ wiki/ --ext .md --wiki --index Home -t markdown_github
pod2pandoc lib/ docs/ --standalone
```

8.4.2 DESCRIPTION

pod2pandoc converts “Plain Old Documentation” format (aka Pod, see [perlpod](#)) to the [Pandoc](#) document model for further processing to other document formats (HTML, Markdown, LaTeX, PDF, EPUB, docx, ODT, man...).

By default or with input - a document is read from STDIN. Multiple input files are combined to one document and listed in document metadata field `file`. The result is printed in JSON format or passed to Pandoc if additional options such as `-t` or `-o` are given.

To process a whole directory of `.pm` and/or `.pod` files, specify an input directory as first input and an optional output directory as second.

8.4.3 OPTIONS

-help|-h|-? Print out usage information and exit.

- parse FORMATS** Parse and include data sections. Use `*` for all.
- podurl URL** Base URL to Perl modules. Default: `https://metacpan.org/pod/`. A false value such as `0` or `"` will disable linking to external modules.
- ext EXT** Output file extension when processing a directory. Default: `html`.
- index NAME** Index file (e.g. `index.html`) name when processing a directory. Can be disabled with `0`.
- name** Include `NAME` section which is removed by default.
- wiki** Create wikilinks when processing a directory.
- update** Only process when input file is newer than output file.
- default-meta FILE** Read default metadata from given file.
- quiet** Don't emit warnings and status information.

... Additional options are passed through to pandoc executable.

8.4.4 SEE ALSO

This script is part of *Pod::Pandoc* and implemented in *App::pod2pandoc*. Together with Pandoc document converter it can be used as customizable replacement for specialized Pod converter scripts such as `pod2html`, `pod2projdocs`, `pod2man`, `pod2readme`, `pod2usage`, `pod2latex`, `pod2markdown`, `pod2text`, and `pod2projdocs`.