NAME

sty2dtx −− Converts a LaTeX .sty file to a documented .dtx file

VERSION

Version: v2.4 – 2022/10/18

LINKS

CTAN: https://ctan.org/pkg/sty2dtx

Repository: https://github.com/MartinScharrer/sty2dtx

Issues: https://github.com/MartinScharrer/sty2dtx/issues

COPYRIGHT

DESCRIPTION

Converts a .sty file (LaTeX package) to .dtx format (documented LaTeX source), by surrounding macro definitions with 'macro' and 'macrocode' environments. The macro name is automatically inserted as an argument to the 'macro' environemnt. Code lines outside macro definitions are wrapped only in 'macrocode' environments. Empty lines are removed. The script is not thought to be fool proof and 100% accurate but rather as a good start to convert undocumented style files to .dtx files.

Basic Usage

```
perl sty2dtx.pl infile [infile ...] outfile
```

or

```
perl sty2dtx.pl < file.sty > file.dtx
```

Supported Definitions

The following macro definitions are detected when they are at the start of a line (can be prefixed by \global, \long, \protected and/or \outer):

```
\def   \edef   \gdef   \xdef
\newcommand{\name}     \newcommand*{\name}
\newcommand\name       \newcommand*\name
\renewcommand{\name}   \renewcommand*{\name}
\renewcommand\name     \renewcommand*\name
\providecommand{\name} \providecommand*{\name}
\providecommand\name   \providecommand*\name
\@namedef{\name}       \@namedef\name
```

The following environment definitions are detected when they are at the start of a line:

```
\newenvironment{name}  \renewenvironemnt{name}  \provideenvironment{name
```

The macro and environment definition must either end at the same line or with a '}' on its own on a line.

USAGE

```
sty2dtx [<options>] [--<VAR>=<VALUE> ...] [--] [<infile(s)>] [<outfile>]
```

Files

- can be '-' for STDIN or STDOUT, which is the default if no files are given
- multiple input files are merged to one output file

Variables

Variables can be defined using

```
--<VAR>=<VALUE>
```

or

```
--<VAR> <VALUE>
```

and will be used for substitutions in the template file.

Common variables:

```
author, email, maintainer, year (for copyright),
version, date, description (of package/class),
type (either 'package' default or 'class'),
filebase (automatically set from output or input file name),
```

Options

–h Print this help text
–H Print extended help
–V Print version and copyright
–v Be verbose
–o *output* Use given file as output
–O Overwrite already existing output file(s)
–B Use basename of single input file for output file
–I Also create .ins (install) file
–c Only use code section (like v1.0)
–r Remove existing 'macro', 'macrocode', etc. environments.
–R Do not remove existing 'macro', 'macrocode', etc. environments.
–i *ins-file* Create .ins file with given name
–t *template* Use this file as template instead of the default one
–T *template* Use this file as template for the .ins file
–e *file* Export default .dtx template to file and exit
–E *file* Export default .ins template to file and exit
–D Use current date as file date
–F *file* Read more options and variables from file.
–N Do not read default config file; must be the first option

Config files

A default config file either named 'sty2dtx.cfg' or '.sty2dtx.cfg' is searched in the current directory, the users home directory and the directory of this script, in this order. The first one found is loaded. If none is found the 'texmf' tree is searched for a 'sty2dtx.cfg' config file. As

with −F files the config file should contain one option or variable per line. Lines starting with '%' or '#' are ignored.

Examples

Produce 'file.dtx' from 'file.sty':

```
sty2dtx.pl < file.sty > file.dtx
```

or

```
sty2dtx.pl file.sty file.dtx
```

or

```
sty2dtx.pl -B file.sty
```

Produce 'file.dtx' and 'file.ins' from 'file.sty':

```
sty2dtx.pl -I file.sty file.dtx
```

or

```
sty2dtx.pl file.sty -i file.sty file.dtx
```

or

```
sty2dtx.pl -IB file.sty
```

Set custom variable values:

```
sty2dtx.pl --author Me --email me@there.com mypkg.sty mypkg.dtx
```

Produce DTX file for a class:

```
sty2dtx.pl --type class mycls.sty mycls.dtx
```

AUTHOR

Martin Scharrer

E−mail: martin.scharrer@web.de

WWW: <http://www.scharrer−online.de>