NAME

edtx2dtx – Convert .edtx into .dtx

SYNOPSIS

edtx2dtx *filename.edtx* > *filename.dtx*

DESCRIPTION

This utility wraps code chunks by a (commented and properly indented) `\begin{macrocode}` and `\end{macrocode}` pair. A code chunk is any sequence of lines not introduced by a single (possibly indented) comment character (`%`); i.e. both non–commented lines and lines introduced by multiple comment characters count as code, and documentation lines may be indented.

The part of the file which should be processed should be marked by a `\begin{macrocode}` and `\end{macrocode}` pair. In other words, everything preceding `\begin{macrocode}` is considered a header and not changed, and everything following `\end{macrocode}` is considered a trailer and not changed. In fact, any number of `\begin{macrocode}` — `\end{macrocode}` pairs is allowed. (The system allows for the driver to remain as is, and also makes it easy to paste a `.dtx` file into an `.edtx`, as using this utility on a `.dtx` does not change the file.)

The utility furthermore replaces the first occurrence of `<filename>.edtx` in the header, followed by an optional parenthesized note, with `<filename>.dtx`, plus the note on how the file was generated.

OPTIONS

–c, ––comment

The input comment character; the default is `%`. (Any input comment characters in the header and the trailer will be replaced by the output comment character, which is always `%`.)

–b, ––begin–macrocode *regex*

–e, ––end–macrocode *regex*

Use these options to change the default *input* `\begin{macrocode}` and `\end{macrocode}` markers. (The *output* markers are always `\begin{macrocode}` and `\end{macrocode}`.) The markers should be given as regular expressions matching the entire line. The input markers are not kept in the file, i.e. they are replaced by the output markers.

–B, ––Begin–macrocode *regex*

–E, ––End–macrocode *regex*

As `-b` and `-e`, but the input markers are kept in the output.

–s, ––strip–empty

If this option is given, empty lines are not included in the output.

–h, ––help

Show help and exit.

–V, ––version

Show the version number and exit.

EXAMPLES

If `input.edtx` is just like a `dtx` file, just without `\begin{macrocode}` and `\end{macrocode}` around every chunk of code:

```
edtx2dtx input.edtx > output.dtx
```

Memoize uses the invocation below to produce a .dtx from documented Python code of memoize-extract.py. The value of −c adapts the edtx2dtx to Python comments. −B marks everything preceding the version number statement as a header, and keeps the version number statement in the output. Similarly, −E says that the processing should stop when encountering Emacs’ local variables (which are kept, unchanged, in the output).

```
edtx2dtx -s -c '#' -B '^__version__' -E '^# Local Variables:' $< \
```

SEE ALSO

easydoctex.el