dtxgen - generate template for LaTeX self-extracting .dtx file

Usage

```
dtxgen [options] basename.[sty|cls]
```

Options

dtxgen recognizes the following options:

`-s, --short=...`	A short, one-liner, description for the class or package. By default, the string 'A new LaTeX class' or 'A new LaTeX package' will be used.
`-n, --name=...`	Your name (first name, followed by surname). Alternatively, you can set a default value in the environment variable `NAME`; if you do so and still use this option, the option's value will have priority.
`-m, --mail=...`	Your email address. Alternatively, you can set a default value in the environment variable `EMAIL`; if you do so and still use this option, the option's value will have priority.
`-c, --class=...`	For class templates only: inserts a `\LoadClass {...}`, so that the new class will start with the properties of the ... class. The default is article.
`-d, --date=...`	Set the initial version's date. By default, the current date will be used. The date should be entered in the `yyyymmdd` format, although it will be stored the LaTeX way: `yyyy/mm/dd`.
`-V, --version`	Prints the script's version and exits.
`-h, --help`	Prints help information and exits.

Description

This script is based on the self-extracting model dtx file described by Joseph Wright, see http://www.texdev.net/2009/10/06/a-model-dtx-file/.

You can use **dtxgen** when you decide to create a new LaTeX class or style. For example, when you decide to make a class named *myclass*, you could run:

```
dtxgen -n 'your name' -m 'your@email.ad' myclass.cls
```

or, if you have an environment with your name and email address defined in NAME and EMAIL, you could simply type:

```
dtxgen myclass.cls
```

and you would end up with five files: myclass.dtx, myclass.cls, myclass.pdf, README, and Makefile.

The resulting .dtx file is a template file in which some minimal example content will have to be replaced with real content, but the dates, basename, author's name and email address are already in place and, depending on whether you use used a .cls or a .sty extension in the argument, it is formatted to be either a class or a package source file.

Makefile

The Makefile can be used to compile new versions of your work; it contains the following targets:

`all`	(the default) generate the style or class file, the pdf-documentation, and a README file.
`distclean`	remove all files that can be regenerated,
`clean`	same, except the style or class file, the pdf-documentation, and a README file.
`inst`	install in the user's TeX tree,
`install`	install in the local TeX tree (uses sudo)
`zip`	produce a zip file ready for upload to CTAN