dtxgen generate template for LaTeX self-extracting .dtx file

doc generated from the script with gendoc — bash script, version=1.04

Synopsis

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

Description

dtxgen creates a template for a self-extracting .dtx file, based on the model described by Joseph Wright. It is useful for those who plan to create a new Documented LaTeX Source (.dtx) file.

Usage example:

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

The script takes some variables such as:

- name and email address of the author,
- a short description of the class or package generated from the .dtx file,
- a date

from environment variables, or from command line options and generates, among more, a template for the .dtx file with some minimal examples. Of course, the user will have to replace those examples with the real work, 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.

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`.

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 wills have priority.

`-m,--mail=...`

Your email address. Alternatively, you can set a default value ins 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`.

`-q,--quiet`
: Run quietly

`-V,--version`
: Prints the script's version and exits.

`-h,--help`
: Prints help information and exits.

`-H,--Help`
: print full documentation via less and exit

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

Changes

Changes with respect version 1.03:

- Clearer comments
- scripts' README information incorporated in the documentation.

Author and copyright

`Author`	Wybo Dekker
`Email`	wybo@dekkerdocumenten.nl
`License`	Released under the GNU General Public License