

# Advanced glossaries package use and latexmk in T<sub>E</sub>XShop

Herbert Schulz  
herbs2@mac.com

2017/01/29

## 1 Introduction

The latexmk engines supplied with T<sub>E</sub>XShop are compatible with both the deprecated glossary package and simple uses of its modern replacement, the glossaries package. The glossaries package has many added features (e.g., using xindy rather than makeindex for processing specific languages and utf-8 encoded documents) that are inaccessible using the default latexmk settings supplied with T<sub>E</sub>XShop.

### 1.1 Supplied Files

The latexmkrcDONTedit file supplied with this document is set up to enable those advanced features of glossaries at the cost of losing compatibility with the glossary package. A platexmkrc file is also supplied to restore glossary compatibility for old projects using that package.

The next two sections of this document have instructions for installing the latexmkrcDONTedit file as well as the use of the platexmkrc for old projects. The final section is about creating custom glossary/acronym lists.

## 2 Installing the new latexmkrcDONTedit file

First open up ~/Library/TeXShop/bin. **Note: ~/Library is the Library folder in your HOME folder; NOT /Library, the Library folder at the root of your Hard Drive. By default ~/Library is hidden in OS X 10.7 and later. To open ~/Library hold the Opt key down while clicking on the Go Menu in Finder; a new item to open that Library folder will appear. Recent versions of T<sub>E</sub>XShop have a T<sub>E</sub>XShop → Open ~/Library/TeXShop menu command to open the proper folder.**

Next, if you have customized the latexmkrcedit file in ~/Library/TeXShop/bin, rename it OLDlatexmkrcedit so you can copy your changes into a new latexmkrcedit edit file that will be created the next time you execute one of T<sub>E</sub>XShop's latexmk engines. Otherwise simply delete the latexmkrcedit file from ~/Library/TeXShop/bin.

Finally, open the tslatexmk folder inside ~/Library/TeXShop/bin, rename latexmkrcDONTedit to OLDlatexmkrcDONTedit (just in case you want to go back later) and add the new latexmkrcDONTedit file supplied in the enclosed For ~/Library/TeXShop/bin/tslatexmk folder.

The next time you execute any of T<sub>E</sub>XShop's latexmk engines (e.g., pdflatexmk) a new latexmkrcedit, with the updated data, will automatically be created in ~/Library/TeXShop/bin.

That's it. From now on the normal as well as the advanced features of the glossaries package will automatically be taken care of by any of T<sub>E</sub>XShop's latexmk engines. See the next section for dealing with old projects that use the glossary package.

## 3 Working with the glossary package

Recent versions of the latexmk engines for T<sub>E</sub>XShop allow a “project wide” configuration file, named platexmkrc (**p** for **p**roject), for the special needs of a particular project. Just create any

special commands needed by latexmk for that project and place it, along with all the files for that project, into a folder. Then any files that use a latexmk engine for T<sub>E</sub>XShop in that folder will read in that configuration file.

In the For glossary package projects folder is a platexmkrc file containing the commands necessary to bring back support for the old glossary package. The glossary package will be supported again once that file is placed in the directory with the old project files.

#### 4 Adding custom acronym lists to glossaries

If your project creates a custom glossary/acronym list, e.g., using

```
\newglossary[mygl]{mygloss}{mygn}{mygr}{My Glossary}
```

(where mygl is the log file for that glossary, mygloss is the internal name of the glossary, mygr is the extension of the file produced by the glossaries package for processing, mygn is the extension of the output file from processing the mygr file and My Glossary is the title of the glossary in the final document) you need to give the latexmk engine the “rule” used to convert the mygr file to the mygn file so it can automate the process of conversion.

To do that just create a platexmkrc file for that project with the line

```
add_cus_dep('mygr', 'mygn', 0, 'makeacn2acr');
```

(where the first and second arguments [mygn and mygr in the example above] are what you gave in the newglossary command *in reverse order* in your document and the last argument, makeacn2acr, *shouldn't be changed*) with a blank line below.

From then on whenever a new mygr file is created the latexmk engine will create an updated mygn file, if needed, and continue processing the document.