Using latexmk With TeXShop.

Herbert Schulz
herbs2@mac.com

What is latexmk?

Compiling a tex file that contains cross-references, bibliographic references and/or indexes is a multi-pass process; i.e., you've got to run (pdf/xe)latex multiple times with possible intermediate runs of bibtex and/or makeindex until all references are resolved. The latexmk perl program, rewritten and presently maintained by John Collins[1], automates this multi-pass process. By default it first runs (pdf/xe)latex on a source file, determines file dependencies by examining the log and aux files produced by the run and then automatically runs bibtex and/or makeindex, if needed, and the correct number of additional runs of (pdf/xe)latex to generate the bibliography, index and cross-references. Recent versions of latexmk also work correctly with the nomencl package as well as the glossary and glossaries packages and other packages that produce multiple bibliographies or indexes.

What is here?

There is a set of five engine files to be moved from ~/Library/TeXShop/Engines/Inactive/Latexmk/ (where you are reading this document) two directories up, ~/Library/TeXShop/Engines/ and then restarting TeXShop if it is already running. A second set of eight files are already in ~/Library/TeXShop/bin/ and consist of the latexmk program, five initialization (rc) files used by the five engine files and two shell scripts used for pdftricks and pst-pdf figure processing.

Using latexmk with TeXShop.

There are five engine files; one for running latex (with a final run through dvips and ps2pdf14) [latexmk.engine], one for using pdflatex [pdflatexmk.engine], one for xelatex [xelatexmk.engine] and two for using the pdftricks or pst-pdf packages with pdflatex [pdftricksmk.engine or pst-pdfmk.engine respectively]. The exact form of the commands and options used are in the corresponding rc file (e.g., latexmkrc for the latexmk.engine) in ~/Library/TeXShop/bin/.

You can use these engine files by using the drop down menu on the source tool bar or placing the line

```
% !TEX TS-program = pdflatexmk
```

(for using pdflatex—similar lines for latex and xelatex) at the top of your document; then simply using Typeset (⌘-T) will automatically run the proper engine. Using latexmk with the epstopdf, pdftricks and pst-pdf packages is discussed later.

I have only tested these engines with relatively trivial distributed documents (which include other files using \include commands) but it appears that latexmk deals with them properly. Note that when compiling a file with name rootname.tex a file

[1] The latexmk web site is <http://www.phys.psu.edu/~collins/software/latexmk-jcc/>. You can get the latest version of latexmk at <http://www.phys.psu.edu/~collins/software/latexmk-jcc/versions.html>.

with name rootname.fdb_latexmk[2] is created. This file contains the dependency information for the distributed document so making changes in an included file will force the recompile of the root document by latexmk.

Noteworthy Changes with latexmk.

Versions of latexmk prior to 3.21c weren't able to deal with the glossary, glossaries or nomencl packages because they re-write their output file(s) with each run of (pdf/xe)latex or use custom file extensions. This changed with latexmk 3.21c. The rc files included with this version of latexmk for TeXShop are set to recognize the standard file extensions produced by the these packages and process them correctly and "auto-magically." If you are creating custom glossaries or indexes you will have to properly edit the rc files (e.g., pdflatexmkrc) found in the ~/Library/TeXShop/bin/ directory to add the dependencies; it should be fairly clear from the contents of the rc files what has to be added to those files.

Another major addition in latexmk since 3.21c is support for packages that create multiple bibliographies and/or indexes; e.g., when the bibunits, chapterbib, multibib, multind or similar packages are used. The extra processing needed for those packages happens automatically. Unfortunately, the index package uses the same naming scheme[3] as the glossary and glossaries packages (see the sub-section below) so you need to define extra dependencies and processing rules in the provided rc files. There was a bug in latexmk 3.21j that didn't allow it to work properly with the index package when creating an ordinary index (an .idx file); this was corrected with version 4.01 of latexmk.

With latexmk 4.11 comes three bug fixes:

1. Corrects a bug with distributed documents using bibtex where changes in bibliography citations did not always trigger a rerun of bibtex.
2. Fixed a problem when latexmk did not detect changed aux files, etc., on a small document when the run of (xe/pdf)latex was within the 1-second granularity of file times.
3. Improved start-up times on some large documents by avoiding unnecessary recalculations of md5 checksums.

Using the epstopdf package with latexmk.

A word about MacTeX 2009

There are three changes to the graphics sub-system that first appear in MacTeX 2009:

1. The graphic(s/x) package now loads the epstopdf package when compiling with pdflatex. This is an attempt to make eps graphics inclusion under pdflatex a bit more transparent. (Note: This feature may not be present in the inital release of MacTeX 2009.)
2. The epstopdf package now defaults to using the [update,append] option. This has consequences if you don't use extensions when you include graphics files in your document.
3. To prevent any problems with overwriting a foo.pdf the default conversion is now foo.eps → foo-eps-converted-to.pdf[4].

[2] The dependency file had extension dep in previous versions of latexmk but didn't do a complete job of keeping track of those dependencies.

[3] Custom extensions rather than standard extensions with custom root file names.

[4] This suffix can be customized.

The third of the changes to epstopdf leads to problems with latexmk version 4.08 and earlier since the base file name changes. To make the latest epstopdf operate properly with latexmk version 4.08 and earlier I suggest creating an epstopdf-sys.cfg file which contains the line

```
\epstopdfsetup{update,prepend,prefersuffix=false,suffix=}
```

and placing it in ~/Library/texmf/tex/latex/config thus making epstopdf behave as before; the conversion becomes foo.eps → foo.pdf. Using latexmk version 4.10 or later requires no changes to epstopdf behavior but you may still do so if you wish to retain the pre-2009 behavior. You can find out the version number of the latexmk program you are using by running the command

```
~/Library/TeXShop/bin/latexmk -v
```

in Terminal.

Working with epstopdf.

Versions of epstopdf from 1.5 on will automatically update a previously generated pdf file if the corresponding eps file is updated[5]. To let latexmk "know" that it should allow runs of pdflatex if the corresponding eps file is updated the necessary rc files (the ones that run pdflatex rather than latex; pdflatexmkrc, pdftricksmkrc and pst-pdfmkrc) contain a special dependency and rule

```
add_cus_dep('eps', 'pdf', 0, 'cus_dep_require_primary_run');
```

which passes latexmk the proper behavior.

If you are using epstopdf 1.5 or later with earlier TEX distributions you should invoke it using the [update,prepend] options. For versions of epstopdf earlier than 1.5 you should edit the pdflatexmkrc, pdftrcksmkrc and pst-pdfmkrc files by commenting out the original dependency (place a # before the line

```
add_cus_dep('eps', 'pdf', 0, 'cus_dep_require_primary_run');
```

in that file) and uncommenting the new dependency (remove the # from the start of the line

```
#add_cus_dep('eps', 'pdf', 0, 'cus_dep_delete_dest');
```

in that same file). This will have latexmk remove the pdf file before running pdflatex so epstopdf will recreate the pdf file.

In version 1.5 and later of the epstopdf package you can also specify non-default processing for the eps to pdf conversion[6]. Since latexmk lets the epstopdf package do all of the necessary processing of the eps file any customized processing defined in the tex source file will be used.

Note: I have noticed that there are times when including the eps extension in \includegraphics still gives rise to additional runs of pdflatex so I still recommend you leave off the extension in \includegraphics commands.

Using the pdftricks package with latexmk.

The pdftricks package allows the inclusion of pstricks graphics in documents compiled with pdflatex. The package generates a file for each postscript figure included in the document. Each of those figure files is then processed to produce a pdf file containing a figure with a tight enclosing bounding box. The pdftricksmk engine included with this version of latexmk processes the original file, the figure files, etc., all only if they have changed. To use the engine place the line

[5] Versions of epstopdf earlier than 1.5 never updated the pdf file once it existed.

[6] The default processing uses the epstopdf command which, in turn, uses ghostscript.

```
% !TEX TS-program = pdftricksmk
```

at the start of the file and Typeset the file. The processing steps for each of the figure files is latex→dvips→ps2pdf14→pdfcrop to ensure the proper bounding box is created for each figure. **NOTE: you must use the [noshell] option to the pdftricks package or latexmk will get into a run-on condition. All figure processing will be taken care of by latexmk.**

Using the pst-pdf package with latexmk.

The pst-pdf package also allows the inclusion of pstricks graphics in documents compiled with pdflatex. When the source file is compiled with latex a dvi file containing all of the figures is created. Further processing through the sequence dvips→ps2pdf14→pdfcrop produces a single file that contains all of the figures with proper bounding boxes. A run of pdflatex on the source file then includes all of the figures previously generated. The pst-pdfmk engine takes care of all of the intermediate processing of the figures as well as the final run(s) of pdflatex, etc. To use the engine place the line

```
% !TEX TS-program = pst-pdfmk
```

at the start of the file and Typeset the file.

The glossary, glossaries and such packages.

Packages that produce multiple and custom indexes, glossaries, etc., use one of two naming schemes for the multiple files they create:

1. The first uses standard extensions but special files names for the generated files. Latexmk can keep track of real changes in and "knows" how to process these files. The multibib and multind packages are examples that use this method.
2. The second uses the source file name for the file but uses custom extensions to create the files. Latexmk needs "help" to know how to process these files in the form of dependencies and rules. Dependencies tell latexmk what the input and output extensions are and which rule to use to go from input to output. The index, glossary and glossaries packages are examples that use this second method.

In addition, while the glossaries package supersedes the glossary package the order of the file extensions created by acronym and custom lists, processed by makeindex and then read in by subsequent runs of (xe/pdf)latex are reversed in the two packages. This latest version of latexmk configured for TeXShop works correctly for both packages. If you need to create your own custom lists see the examples in the rc files for creating dependancies and rules for latexmk.

What these engines won't do, etc.

There are many features of latexmk that aren't used in these simple engine files. See the documentation for latexmk in the supplied full distribution.

In addition, the placement of the latexmk program in ~/Library/TeXShop/bin/ is non-standard; that directory is not on the standard path. It is possible to put the program in /usr/local/bin/ and it will then be usable from the command line. If you install the program there you should modify the engine files to reflect the change in location.

The contents of the rc files corresponds to the the settings for commands for TeXShop on my system. They are simply text files. Please read the latexmk documentation before changing the contents.

Finally, changes in `eps` files *included in figures* created by the `pdftricks` or `pst-pdf` packages are *not* detected by this packaging `latexmk` at this time. I hope to correct that problem at a later date.

Update for TeXShop 2.18 (and later) with MacTeX 2008 (ditto).

The `rc` files for this version of `latexmk` for use with TeXShop have been updated to allow use of `synctex`, a `tex↔pdf` synchronization technology, with MacTeX-2008 and TeXShop 2.18. If you are using MacTeX-2007 or earlier TeX distributions and the inconsequential error message about an unknown option bothers you, remove the `-synctex=1` options provided in the supplied `rc` files.

Try it... I hope you like it.

Good Luck,
Herb Schulz
(herbs2@mac.com)