Using latexmk with TEXShop

Herbert Schulz
herbs2@mac.com

1 What is latexmk?

Compiling a tex file containing 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[2] 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.

2 Quick Start!

This section will get you started quickly. Unless you are trying to customize the behavior of the supplied engines or want to use the more esoteric engines this really is all you need!

2.1 Quick Install.

To activate the latexmk engine files simply move all the files with extension .engine from `~/Library/TeXShop/Engines/Inactive/Latexmk/` two folder levels up, to `~/Library/TeXShop/Engines/`, and (re-)start TEXShop. (Note: `~/Library/` is the Library folder in your HOME folder.) When you click on the popup engine menu on the Source toolbar the newly enabled engines names should appear; see Figure (1) to see how that menu changes. **Note: the engine names will *not* appear in the Typeset Menu.**

2.2 Quick Use.

At the top of your Source file place the line

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

to use the pdflatexmk engine which will use pdflatex to typeset your document. Substitute latexmk or xelatexmk for pdflatexmk to use latex or xelatex to typeset your Source. From then on simply using Typeset → Typeset (Cmd-T) will run through the complete process of fully typesetting your document.

3 What is here?

There is a set of ten engine files to be placed in `~/Library/TeXShop/Engines/`. There is a tslatexmk folder already placed in `~/Library/TeXShop/bin/`. The files in that folder consist of the latexmk program, ten basic initialization (rc) files used by the ten engine files, a common file for personal settings (latexmkrcDONTedit) and two shell scripts used for pdftricks and pst-pdf figure

[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`>.

[2] As of version 4.22 latexmk will automatically choose between running bibtex or biber as required.

processing. When any of the new engines is first run the latexmkrcDONTedit file will automatically be copied to `~/Library/TeXShop/bin/latexmkrcedit` if it doesn't already exist. You may copy the file there manually if you wish. **Any changes or additions to the configuration (e.g., new dependencies and rules) should be placed in the laxtexmkrcedit file. When TEXShop is updated the files in the `~/Library/TeXShop/bin/tslatexmk` may automatically get updated; don't edit them or your changes may get lost.**

4 What is New with this Version

The engine files supplied with this version of latexmk for TeXShop now allow you to have a platexmkrc file, containing latexmk configuration information, in the same folder as the file you typeset (i.e., the root file for a distributed document). This can be useful if your project needs special configuration options for a certain task.

E.g., you wish to use texindy instead of makeindex to process the idx file into a ndx file you might include a platexmkrc file in the same directory as the root file of a project with contents

```
$makeindex = "texindy %O -o %D %S";
```

to use texindy rather than the default makeindex; make sure you end the file with a blank line. You could also add special options to the processing for a particular situation. Make sure you understand the syntax used by latexmk for customzing commands before playing with this feature.

One warning: if you are going to use this feature understand that the platexmkrc file will be used for *any* file in that folder that is typeset.

5 Using latexmk with TEXShop.

NOTE: If you are updating to this version of latexmk for TEXShop from a previous version you need only activate the engine files, as noted above, and restart TEXShop after installing the files.

There are ten engine files; two for running latex (one with a final run through dvips and ps2pdf [latexmk.engine] and one with a final run through dvipdfmx [dvipdfmxmk.engine]), two for using pdflatex [pdflatexmk.engine and sepdflatexmk.engine] (the second one for use when you need to use --shell-escape), one for using xelatex [xelatexmk.engine], one for using lualatex [lualatexmk.engine], two for using the pdftricks or pst-pdf packages with pdflatex [pdftricksmk.engine or pst-pdfmk.engine respectively] and one for use with files that use the asymptote package [asymptotemk.engine]. The final engine is a very basic engine for typesetting dtx files for a package into the final documentation [dtxmk.engine]. 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/tslatexmk/` and shouldn't be changed.

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

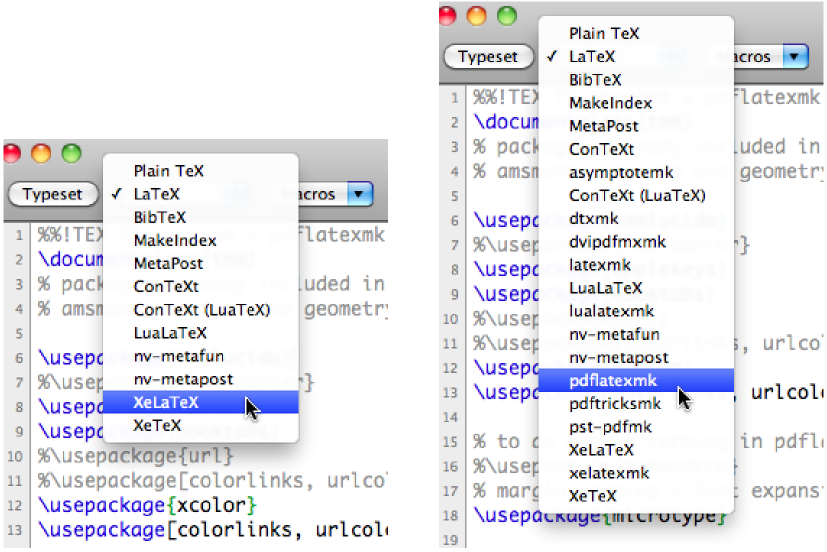
```
% !TEX TS-program = pdflatexmk
```

(for using pdflatex—similar lines for latex and xelatex, etc.) at the top of your document[3]; then simply using Typeset (Cmd-T) will automatically run the proper engine. Note: these engines *don't* appear under the Typeset Menu but only under the pop-up menu on the source toolbar. Figure (1) shows the default and updated pop-up menu after installing the latexmk engine files.

Detailed information about 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

[3] For the dtxmk engine the line should be placed just below the initial "% \iffalse" line of the dtx file.



```
%%!TEX            latexmk
\docum
% pack            uded in
% amsm            geometr

\usepa
%\usep            }
\usepa
\usepa
%\usepackage{url}
%\usepackage[colorlinks, urlco
\usepackage{xcolor}
\usepackage[colorlinks, urlcol
```

```
%%!TEX            latexmk
\docum
% pack            uded in
% amsm            geometr

\usepa
%\usep            }
\usepa
\usepa
%\usep
%\usep            , urlco
\usepa
\usepa            urlcol

% to c            in pdfl
%\usep
% marg            expans
\usepackage{microtype}
```

Figure 1: Default and updated versions of the engine pop-up menu after installing the latexmk engines.

that when compiling a file with name rootname.tex a file with name rootname.fdb_latexmk[4] 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.

5.1 Using the epstopdf package with latexmk.

5.1.1 A word about MacTEX 2009 & 2010

There are two changes to the graphics sub-system that first appear in MacTEX 2009:

1. 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.

2. The default conversion is now foo.eps → foo-eps-converted-to.pdf[5] to prevent any problems with overwriting a foo.pdf .

The second 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.cfg file, to be placed in `~/Library/texmf/tex/latex/config` and containing the line

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

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/tslatexmk/latexmk -v
```

in Terminal.

Starting with MacTEX 2010 the graphic(x/s) package will automatically load the epstopdf package if it detects that the file is being compiled using pdftex in pdf mode (normal for pdflatex).

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

[5] This suffix can be customized.

You no longer need to explicitly use the epstopdf package. Not only that but, if you haven't defined custom conversion and are only trying to convert eps → pdf there isn't even a need to use the --shell-escape flag: edit the latexmkrcedit file to eliminate it from all of the engines.

5.1.2 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[6]. 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, pst-pdfmkrc and asymptotemkrc) 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, pst-pdfmkrc and asymptotemkrc 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. NOTE: These files may be automatically updated when TEXShop is updated and you may lose your changes!

In version 1.5 and later of the epstopdf package you can also specify non-default processing for the eps to pdf conversion[7]. Since latexmk lets the epstopdf package to do all of the necessary processing of the eps file any customized processing defined in the tex source file will be used.

5.2 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

```
% !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 → ps2pdf → 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.**

5.3 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 → ps2pdf → 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

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

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

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

at the start of the file and Typeset the file.

5.4 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 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 latexmkrcedit file for creating dependancies and rules for latexmk.

6 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.

The placement of the latexmk program in `~/Library/TeXShop/bin/tslatexmk/` is non-standard; that directory is not on the standard path. It is possible to put the program in `/usr/local/bin/` or use the version of latexmk that is part of MacTEX-2008 and later and it will then be usable from the command line. If you use the program in one of those locations 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.

7 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.

8 Update for TEXShop 2.30 (and later).

The --file-line-error flag has been set for all compiles in the basic rc files. TEXShop 2.30 and later uses the information provided by this flag to localise the location of compile errors when you use the Go to Error command.

9 Update for TEXShop 2.32 (and later).

Starting with TEXShop 2.32 when TEXShop is updated any updates to the files in the `~/Library/TeXShop/bin/tslatexmk/` folder will automatically be installed. Any changes directly made to those files will be lost. Most of the extra dependencies and rules that were common to all the rc files have been moved to the new `~/Library/TeXShop/bin/latexmkrcedit` file and all additional personal dependencies and rules should be moved to that file. The latexmkrcedit file will *not* be updated automatically.

Try it. . . I hope you like it.

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