

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 Collins¹, 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 seven 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, seven basic initialization (`rc`) files used by the seven engine files, a common file for personal settings (`latexmkrcDONTedit`) and two shell scripts used for `pdftricks` and `pst-pdf` figure 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 `latexmkrcedit` 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.**

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 seven 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`]), one for using `pdflatex` [`pdflatexmk.engine`], one for using `xelatex` [`xelatexmk.engine`], two for using the `pdftricks` or `pst-pdf` packages with `pdflatex` [`pdftricksmk.engine` or `pst-pdfmk.engine` respectively] and, finally, one for use with files that use the `asymptote` package [`asymptotemk.engine`]. The exact form of the commands and options used are in the corresponding `rc` file (e.g.,

¹The `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>.

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; 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 with name rootname.fdb_latexmk² is created. This file contains the dependency information for the distributed document so making changes in an included file will force the recompilation 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/xelatex) or use custom file extensions. This changed with latexmk 3.21c. The latexmkrcedit, generated the first time you run one of the new engines, is set to recognize the standard file extensions produced by these packages and process them correctly and "auto-magically." If you are creating custom glossaries or indexes you will have to properly edit the latexmkrcedit file found in the ~/Library/TeXShop/bin/ directory to add the dependencies; it should be fairly clear from the contents of that file what has to be added.

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 scheme³ 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 came three bug fixes: i) Corrected a bug with distributed documents using bibtex where changes in bibliography citations did not always trigger a rerun of bibtex. ii) 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. iii) Improved start-up times on some large documents by avoiding unnecessary recalculations of md5 checksums. In particular, the last item seems to result in a very noticeable improvement in performance.

Latexmk 4.12 adds one feature. If you have a tex file and an associated bbl file but *not* the original bib file the -bibitex-cond option tells latexmk *not* to run bibtex which would overwrite the bbl file with an empty bibliography. If the bib file is present and along the standard search path latexmk behaves identically to its previous versions. Versions 4.13 and 4.13a have now set this option as the default behavior. The 4.16 version of latexmk fixes a problem with misparsed log files in some versions

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

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

of pdflatex and updates documentation to mention previously undocumented feature about the use of temporary files in making ps and pdf files. Version 4.16a fixed some bugs that don't effect its behavior under T_EXShop.

Using the `epstopdf` package with `latexmk`.

A word about MacT_EX 2009

There are two changes to the graphics sub-system that first appear in MacT_EX 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. To prevent any problems with overwriting a `foo.pdf` the default conversion is now `foo.eps` → `foo-eps-converted-to.pdf`⁴.

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-sys.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.

Working with `epstopdf`.

Versions of `epstopdf` from 1.5 on will automatically update a previously generated pdf file if the corresponding eps file is updated⁵. 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 `asympotemkrc`) 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 T_EX 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 `asympotemkrc` 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 T_EXShop 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 conversion⁶. Since `latexmk` lets the `epstopdf` package

⁴This suffix can be customized.

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

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

do all of the necessary processing of the eps file any customized processing defined in the tex source file will be used.

Using the pdftricks package with latexmk.

The pdftricks package allows the inclusion of pstricks graphics in documents compiled with pdf_latex. 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.**

Using the pst-pdf package with latexmk.

The pst-pdf package also allows the inclusion of pstricks graphics in documents compiled with pdf_latex. 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 pdf_latex 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 pdf_latex, 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 file 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 T_EXShop 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.

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.

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.

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.

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)