

Using latexmk With T_EXShop

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/x_e)`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/x_e)`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/x_e)`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 eight 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, eight basic initialization (`rc`) files used by the eight 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 T_EXShop 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 T_EXShop.

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

There are eight 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`], 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, finally, one for use with files that use the `asymptote` package [`asymptotemk.engine`]. The exact form of the commands and

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>.

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; 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`.

Using the `epstopdf` package with `latexmk`.

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. To prevent any problems with overwriting a `foo.pdf` the default conversion is now `foo.eps → foo-eps-converted-to.pdf3`.

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.

In 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`). 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 `latexmkrc` file to eliminate it from all of the engines.

Working with `epstopdf`.

Versions of `epstopdf` from 1.5 on will automatically update a previously generated pdf file if the corresponding `eps` file is updated⁴. To let `latexmk` "know" that it should

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

³This suffix can be customized.

⁴Versions of `epstopdf` earlier than 1.5 never updated the pdf file once it existed.

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 \TeX distributions you should invoke it using the `[update,prepend]` options. For versions of `epstopdf` earlier than 1.5 you should edit the `pdflatexmkrc`, `pdftricksmkrc`, `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 \TeX Shop 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` conversion⁵. 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.

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`.

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

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

at the start of the file and Typeset the file.

⁵The default processing uses the `epstopdf` command which, in turn, uses `ghostscript`.

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 `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`.

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)