

# Updated latexmk engines, T<sub>E</sub>XShop's parameter directive and sample project files

Herbert Schulz  
herbs2@mac.com

2017/10/04

## 1 Introduction

T<sub>E</sub>XShop, version 3.77 and later, has a new directive<sup>1</sup> that can pass information to any T<sub>E</sub>XShop engine that is designed to use it; `% !TEX parameter = ...`. The enclosed replacement 'latexmk' engines now support that directive and pass the information to the underlying typesetting engine (latex, pdflatex, xelatex or lualatex) as command line options (e.g., `--shell-escape`). See Section (2) for information about the updated engines and Sub-Section (2.1) for installation instructions.

In addition, the latexmk engines have allowed the use a configuration file on a project wide basis, `platexmkrc`, for a while now. This can be very useful if you need special processing for a project; e.g., having the latexmk engines use `xindy` rather than `makeindex` or using a special `bibtex` style file. There are several sample `platexmkrc` files supplied that show it's use. Some even show how a `platexmkrc` file can replace specialized latexmk engines. Section (3) has information about them.

## 2 Updated latexmk Engines

While T<sub>E</sub>XShop recently introduced the

```
% !TEX parameter = ...
```

directive there aren't many engines that actually use that possible information.

The updated 'latexmk' engines supplied with this document now use that passed parameter as command line options to the underlying typesetting engine. So the line

```
% !TEX parameter = --shell-escape
```

will turn on shell-escape for processing that document. More than one option can be specified so

```
% !TEX parameter = --shell-escape --nonstopmode
```

will send both of those options to the underlying typesetting engine. Parameters that change the location of generated files, e.g., `--output=directory`, are not allowed here for two reasons: the latexmk script also needs to 'know' where to find the files and T<sub>E</sub>XShop won't be able to find those files. The former can be overcome by using a `platexmkrc` file that contains the line

```
$out_dir = "path/to/directory" ;
```

where `path/to/directory` is the full or relative path to the folder where the files are stored. I'm not sure if there is a solution for letting T<sub>E</sub>XShop find the files.

---

<sup>1</sup>A  $\LaTeX$  comment that is read and interpreted by T<sub>E</sub>XShop itself rather than a typesetting engine.

The updated latexmk, pdflatexmk, xelatexmk and lualatexmk engines are the basic ones to be used for typesetting with latex (with with dvips → ps2pdf post processing), pdflatex, xelatex or lualatex respectively. There is also an update to the dvipdfmxmk engine which uses latex (with dvipdfmx post processing).

For backward compatibility there is an update to the asymptotemk engine although it can be easily set up to use a platexmkrc file along with one of the basic latexmk engines; see Section (3) below. The advantage is that you can use any of the basic engines to typeset asymptote figures.

The dtxmk engine has *not* been updated since it also can be easily set up to use a platexmkrc file with one of the basic ‘latexmk’ engines.

Finally, for backward compatibility there is an update to the sepflatexmk (se = shell-escape) engine even though it can easily be replaced by the two declarations

```
% !TEX program = pdflatexmk
% !TEX parameter = --shell-escape
```

in the root source file.

## 2.1 Installation Instructions

If you are reading this document in the `~/Library/TeXShop/Engines/Inactive/Latexmk` folder you have a version of TeXShop that already has the new engines installed. If you previously installed TeXShop you’ll have to activate the new pdflatexmk, sepflatexmk and any other of the engines you previously used by copying the new versions of the engines two folders up, to `~/Library/TeXShop/Engines`, which will replace the older versions.

To install the new engines in an older version of TeXShop (3.77 to 3.88) copy all the enclosed files to `~/Library/TeXShop/Engines/Inactive/Latexmk`, replacing any versions already there, and then copy the pdflatexmk, sepflatexmk and any other of the engines you previously activated two folders up, to `~/Library/TeXShop/Engines`. That’s it! (Note: you can open the `~/Library/TeXShop` folder using the TeXShop → Open `~/Library/TeXShop` menu item.)

## 3 Sample platexmkrc Project Files

For quite some time now the TeXShop ‘latexmk’ based engines have allowed a special platexmkrc (p = project) latexmk configuration file located in the same directory as a root file for the special needs of a project. When any of the latexmk engines runs, the contents of that file are read into latexmk for any custom configurations. These can be anything from using a special .bst file when running bibtex, using xindy instead of makeindex for generating indexes or adding dependencies and rules for automatic processing of files with special extensions. That file will be read when any .tex file in the folder containing the platexmkrc file is processed by a latexmk engine; it’s advisable to place a special project into it’s own folder to prevent errors for other files.

While the use of the platexmkrc file has been around for a while there has been a lack of examples of its use. The platexmkrc samples folder that comes with the engine updates contains six examples, most with a sample file to typeset, for using the platexmkrc file in different ways. Hopefully they will act as templates for any special processing you may need to do with your next project.

The asymptote and dtx versions let you use one of the basic ‘latexmk’ engines instead of the specialized engines use before. The axodraw and glossaries versions define dependencies and rules to allow latexmk do specialized processing of files produced by those packages. Finally, the bibtex8 and imakeindex versions re-define internal latexmk commands to execute specialized versions for a particular need.