Using eps and tiff Graphics with pdflatex

by
H. Schulz and R. Koch

2012/05/30

Introduction

Users who include eps and tiff figures in documents are sometimes surprised to find that they have a problem when compiling those files using pdflatex. By default pdflatex can include jpg, png and pdf files but not eps or tiff files. This document discusses methods of including eps and tiff files in documents compiled with pdflatex. In the end these techniques use Ghostscript or Apple's distiller to convert eps files to pdf format, and Apple's built-in sips program or convert from ImageMagick to convert tiff files to png format.

Conversion of eps to pdf is Automatic in TeX Live 2011 and Later

Since eps illustrations cause such trouble for users switching from TEX to pdfTEX, the authors of TeX Live 2011 made a concerted effort to solve the problem once and for all. Instead of having to explicitly use the epstopdf package the good news is that conversion of eps files to pdf files is now automatic in TeX Live 2011 and later if you use the graphicx package, as almost all users do.

Here's how it works. Suppose your document contains the line

```
\includegraphics{foo.eps}
```

then the graphicx package will automatically convert foo.eps to foo-eps-converted-to.pdf and load that pdf file. When the file is typeset again, foo.eps will be converted again only if it is newer than foo-eps-converted-to.pdf, but in any case foo-eps-converted-to.pdf will be loaded.

Some users omit the extension and just write

```
\includegraphics{foo}
```

which will also work as above: the file foo.eps will be converted to foo-eps-converted-to.pdf and so forth. But there is one tricky point. Suppose the source folder contains both foo.eps and a completely unrelated file named foo.pdf. Then the graphics command will load foo.pdf first and not even notice the eps file.

Incidentally, the system names the converted file foo-eps-converted-to.pdf rather than foo.pdf precisely because a few users might have source files containing unrelated eps and pdf files with the same name; these users wouldn't appreciate an automatic conversion that destroyed their original pdf file!

Users upgrading from earlier versions of TeX Live will be familiar with the command

```
\usepackage{epstopdf}
```

This command is not needed in TeX Live 2011 and later, although it does no harm, because the graphicx package automatically loads it. Users may also be familiar with the TEX command-line flag "--shell-escape", which gives TEX the ability to call external programs during typesetting. This flag is also not needed in TeX Live 2011 or later for eps to pdf conversion because the system implements "restricted-shell-escape" which allows a small list of special system commands to be executed even if shell escape is not set.

Converting TIFF Files

The most common graphic file formats are eps, pdf, png, jpg and tiff. In particular, many scanners output tiff files. The authors of TeX Live 2011 were unable to automate conversion of tiff to png because sips is only available on the Macintosh and convert could not be added to the list of trusted programs — on Windows machines there is an unrelated program named convert which would cause security concerns if called indiscriminately from TEX. So it is necessary to convert tiff to png by hand. There are several ways to do this:

- Apple's Preview program will convert tiff files to png format. Open the tiff file in Preview and then Export the result as a png.
- It is often easier to use Terminal in /Applications/Utilities. If you have the ImageMagick convert program, change directory to the folder containing the tiff file, which we assume is named "myfile.tiff", and then

  ```
  convert myfile.tiff myfile.png
  ```

  If you do not have convert, use Apple's sips program instead:

  ```
  sips -s format png myfile.tiff --out myfile.png
  ```

- TEXShop 2.46 and later and TEXShop 3.10 and later have built-in tiff to png conversion. With a document window active, select Convert TIFF in the File menu. A dialog appears listing all tiff files in the folder containing the source file, and in sub-folders of this file. Select tiff files to be converted — multiple files can be selected — and push Convert. See Figure (1) one page 3.
- Herbert Schulz wrote a Drop Script which can convert all tiff files in a folder to png in a single operation. Drag the individual tiff file or folder to the Drop Script. This is available as 'Tif(f) to Png.zip' at <https://dl.dropbox.com/u/10932738/index.html>.

Adding More Automatic Conversions to TEX

It is possible to modify TEX so it will perform additional conversions automatically during typesetting. The rest of this document is about those modification, which are not for the faint-hearted.

The modifications require additions to the header of the source file — that is the easy part. In the end, these header commands tell TEX to call a conversion program during typesetting, either Apple's Distiller /usr/bin/pstopdf or Apple's sips or ImageMagick's convert. The problem is that TEX is usually configured to disallow calling external programs during typesetting, since calling such programs is a security risk. There are two ways around this problem.

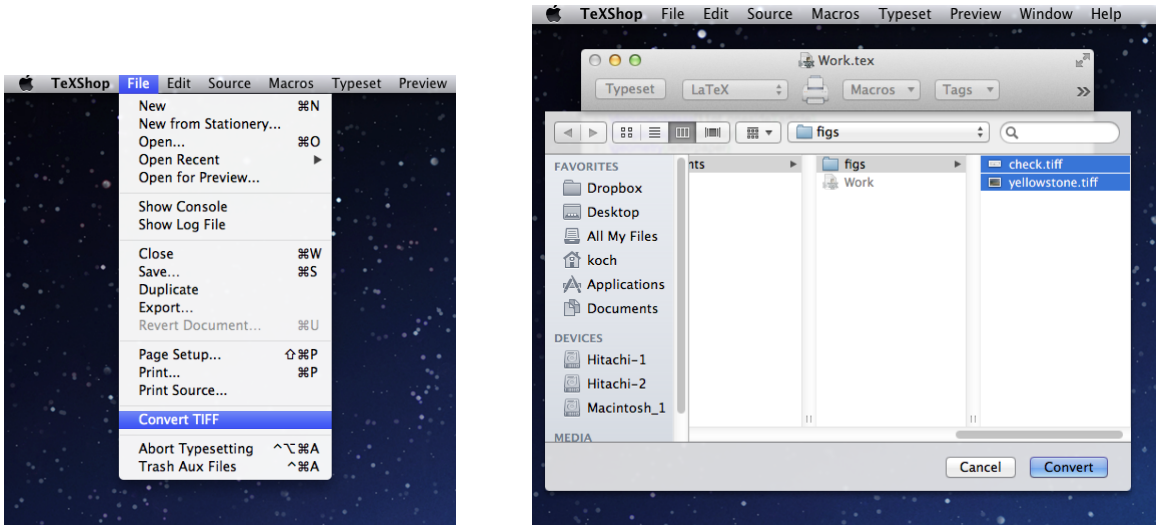

Figure 1: The Convert Tiff Menu in TEXShop 3.10

- The first is to reconfigure TEX by adding the flag --shell-escape to its calling sequence. The effect is to allow TEX to call *any* program, including for instance, a shell command to erase all files in the home directory. How one adds the flag will depend upon which application you are using to typeset your document but adding the flag is usually very easy. Adding --shell-escape to TEXShop is discussed in more detail in Appendix A on page 5.

- The second method, available with TeX Live 2012 or later, is to add one or all of sips, convert and/or /usr/bin/pstopdf to the list of trusted programs for restricted-shell-escape. This involves directly manipulating a TeX Live configuration. This is discussed in more detail in Appendix B on page 5

More Automatic Conversions

Once you allow sips and/or convert and/or pstopdf to be automatically executed using one of the methods described in the previous section you can simply add some lines to your document and have conversions happen on the fly. The rest of this section contains information about what has to be added to your document.

Using Apple's Distiller

By default, conversion of eps files to pdf files is done with Ghostscript. Apple provides an alternate conversion program (often called a distiller), pstopdf. To use this program instead of Ghostscript, add the following line to the document header immediately following inclusion of the graphicx package:

```
\usepackage{epstopdf}
\epstopdfDeclareGraphicsRule{.eps}{pdf}{.pdf}{%
    /usr/bin/pstopdf #1 -o \OutputFile
    }
```

Converting tiff to png

To automate conversion of tiff to png, add the following commands to the document header immediately following inclusion of the graphicx package.

```
\usepackage{epstopdf}
\epstopdfDeclareGraphicsRule{.tif}{png}{.png}{convert #1 \OutputFile}
\epstopdfDeclareGraphicsRule{.tiff}{png}{.png}{convert #1 \OutputFile}
\PrependGraphicsExtensions{.tif, .tiff}
```

If you wish to use Apple's sips converter rather than convert from MacTEX, replace the above commands with these.

```
\usepackage{epstopdf}
\epstopdfDeclareGraphicsRule{.tif}{png}{.png}{%
    sips -s format png #1 --out \OutputFile}
\epstopdfDeclareGraphicsRule{.tiff}{png}{.png}{%
    sips -s format png #1 --out \OutputFile}
\PrependGraphicsExtensions{.tif, .tiff}
```

Automating More Conversions

If you use tiff conversions often you may wish to always load the lines given above automatically. You may do this by creating an epstopdf.cfg file containing only the conversion lines

```
\epstopdfDeclareGraphicsRule{.tif}{png}{.png}{convert #1 \OutputFile}
\epstopdfDeclareGraphicsRule{.tiff}{png}{.png}{convert #1 \OutputFile}
\PrependGraphicsExtensions{.tif, .tiff}
```

or

```
\epstopdfDeclareGraphicsRule{.tif}{png}{.png}{%
    sips -s format png #1 --out \OutputFile}
\epstopdfDeclareGraphicsRule{.tiff}{png}{.png}{%
    sips -s format png #1 --out \OutputFile}
\PrependGraphicsExtensions{.tif, .tiff}
```

and place it in your personal texmf tree at `~/Library/texmf/tex/latex/config/`. But remember that others that see your source file may not have your epstopdf.cfg file and a compile will fail for them. You may also place the epstopdf.cfg file in `/usr/local/texlive/texmf-local/tex/latex/config/` for use by all users of your system; make sure to run

```
sudo mktexlsr
```

in Terminal giving your administrator's password when requested.

If you create the epstopdf.cfg file you needn't include the `\usepackage{epstopdf}` in your source document.

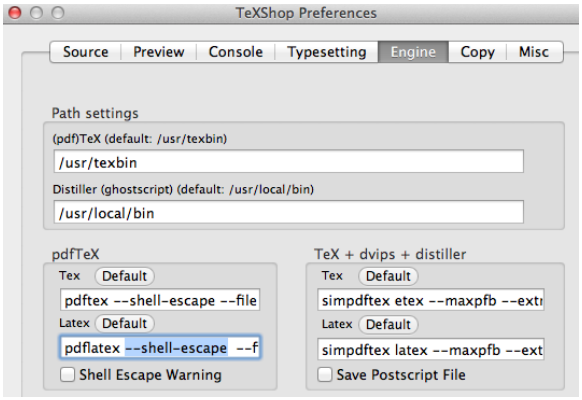

Figure 2: Adding the --shell-escape flag to TeXShop Preferences.

Appendix A — Adding --shell-escape to TeXShop

Note: The following information is only needed by new users of TeXShop or previous users that have manually removed the --shell-escape flag from their preferences. Other users have that flag already set and need not add it. We'd recommend removing it unless you need it!

It is relatively easy to get the basic "engines" to use the --shell-escape flag. There are two places in the Engine tab of TeXShop → Preferences. Simply add --shell-escape, with a surrounding space on each side to the two pdfTeX engine lines. See Figure (2) on page 5.

If you have used any of the latexmk engines, e.g., pdflatexmk, you can add the --shell-escape flag to all of the engines by editing the `~/Library/TeXShop/bin/latexmkrcedit` file. Change the line

```
$TSUserCompileOptions = '';
```

to

```
$TSUserCompileOptions = '--shell-escape';
```

i.e., from an empty string to one that includes the --shell-escape flag, and save the file. Remember `~/Library/` is the `Library` folder in your HOME folder. In Mac OS X 10.7 (Lion) you can open the `~/Library/` folder by pressing the Opt key while clicking on the Finder's Go Menu.

Appendix B — Adding Programs to the Restricted Program List

Note: This method of adding programs to those accepted for the restricted-shell-escape mode is only available in TeX Live 2012 and later. It is much more secure than setting the --shell-escape flag, but it could open a security hole by adding an unsecure program to the list.

In previous examples we used pstopdf, convert or sips as external programs. In the following example we will add Apple's sips program to the allowed list; others can be added as needed but be careful what you are adding.

First create a simple text file that contains

```
shell_escape_commands = \
bibtex,bibtex8,\
kpsewhich,\
makeindex,\
mpost,\
repstopdf,\
sips,\
```

and name it `texmf.cnf`. **Note: the file must end with at least one completely blank line! I typically use two blank lines to make sure I don't mess it up.** The first part duplicates the default list since this command overrides the default rather than adding to the default. Instead of the `sips,\` line you could add `convert,\`, etc., or multiple commands, each on its own line.

Using the "Go to Folder. . . " command in Finder's Go menu, open the

```
/usr/local/texlive/texmf-local/web2c
```

folder and copy the `texmf.cnf` file there. Finally run the command

```
sudo mktexlsr
```

in Terminal giving your administrator's password when requested.

From then on the program(s) you added to the list will be allowed to execute in the default restricted-shell-escape mode without the need for the all encompassing --shell-escape flag.

Note: this method can be useful in other situations. For example, the tkz-fct package uses gnuplot to create graphs on the fly. By adding gnuplot to the list of accepted programs there is no need to have the --shell-escape flag active when using this package.