

# Using eps and tiff Graphics with pdflatex

by  
H. Schulz and R. Koch

2012/05/26

## 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 a way of including eps and tiff files in documents compiled with pdflatex. It involves the ability to convert one graphic format to another on the fly and then include the converted file in the document. In the end these methods use Ghostscript or Apple's distiller and "convert" from ImageMagick or Apple's sips program, so conversion may fail if you did not install Ghostscript and ImageMagick.

## Conversion 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 epsstopdf 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 graphics package, as almost all users do.

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

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

then the graphics 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.

## 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
}
```

Note: **The -shell-escape flag must be active for this to work.**

## More Conversions

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 using the `convert` program from ImageMagick because on Windows machines there is an unrelated program named `convert` which would cause security concerns if called indiscriminately from TeX.

It is possible to automate this conversion on the Macintosh. But in that case, the `epstopdf` package must be explicitly loaded, **and the “-shell-escape” flag must be active**. Then 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}
```

or

```
\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}
```

if you wish to use Apple’s `sips` converter rather than `convert` installed by MacTeX.

## 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.

If you create the `epstopdf.cfg` file you needn't include the `\usepackage{epstopdf}` in your source document but **you still must use the command-line “-shell-escape” flag** to do the extra conversions.

## Why all this talk about the “-shell-escape” flag?

You'll notice that except for the default `eps` to `pdf` conversion noted above all the other conversions require the “-shell-escape” flag to be set to allow `pdflatex` to execute external commands. However, this ability to execute any program from within a `pdflatex` run is somewhat of a security concern.

`TEXShop 3.10` and later are now shipped with the “-shell-escape” option turned off by default. It is believed most other “front end” applications for the `TEX` distribution already do this. This will immediately change things for new users of `TEXShop` but also previous users if they ever have to go back to the default preferences. So most of the conversions given above may stop working out of the box!

## What do I have to do to “fix things up”?

There are basically three methods to convert your `tiff` files to `png` so you can get back to work.

### Recommended: convert `tiff` to `png` before using the graphic

To make up for the loss of automating `tiff` to `png` conversion `TEXShop 3.10` has an additional menu item, `File` → `Convert Tiff` (see the left side of Figure ??). The resulting listing of possible `tiff` files to convert in the source file folder allows you to maneuver to the folder where the files are located and also choose multiple files (see the right side of Figure ??).

If you prefer using the command line you use ‘`convert`’ or Apple’s ‘`sips`’ to do the conversion: use

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

or

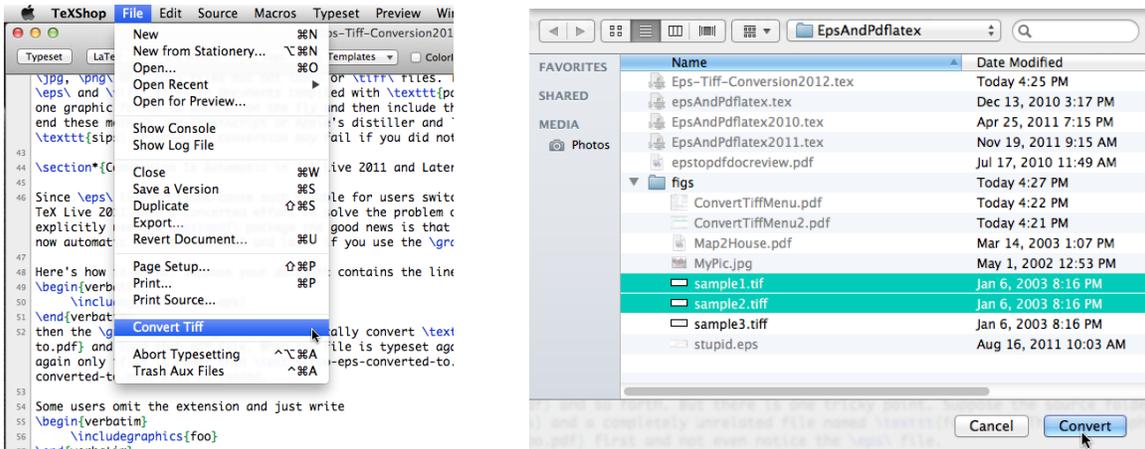


Figure 1: The Convert Tiff Menu in TeXShop 3.10

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

to get myfile.tiff converted to myfile.png.