A Modified Version of `pdftricks.sty`

by
Herbert Schulz[*]

November 29, 2010

Why Does This Version of `pdftricks.sty` Exist?

The `pdftricks` package saves each of the `pstricks` figures as a separate file, then processes that file into a `.pdf` file and finally includes that `.pdf` file in the original file using the `graphicx` package. It uses the sequence `latex` → `dvips` → `ps2eps` → `epstopdf` to produce the included `.pdf` file. Unfortunately `ps2eps` seems to be a bit buggy; it doesn't always produce the correct BoundingBox and then the graphic is clipped. This modified version of the `pdftricks` package alters that processing sequence and adds some other features.

What Does This Version of `pdftricks.sty` Do Differently?

New Processing Sequence

A more reliable processing sequence seems to be `latex` → `dvips` → `ps2pdf` → `pdfcrop` which bypasses the use of `ps2eps` and produces a correctly cropped `.pdf` file. This modified version of `pdftricks` uses that sequence and seems to fix the `BoundingBox` problem. You *must* have the Ghostscript package installed so that `ps2pdf` is installed on your system.

AutoRotate Turned Off by Default

In addition, `ps2pdf`, via ghostscript (`gs`), sometimes rotates a figure depending upon how much text is rotated and the aspect ratio of the figure. This can be very annoying when you know how the figure should be oriented. By default his version of `pdftricks` calls `ps2pdf` with the option `-dAutoRotatePages=/None` set so no automatic rotation will take place. A new option, `autorotate`, is available in this version of `pdftricks`: it will turn *on* the autorotate switch if it is used.

Setting Font Size for Figures

Some code is added to make any text in the figure default to the same size as that in the document. Right now it will only do `10pt`, `11pt` and `12pt` if the document class uses the standard method for saving the point size; using anything else (e.g., the `memoir` class with the `14pt` option) will default to `10pt` in the attached graphic. There are options to allow other choices; see the sub-section below.

---

[*] E-mail address: <herbs2@mac.com>.

There are also options for `pdftricks` to allow defined font sizes. The `10`, `11`, `12` options change the default font size for the figure files to `10pt`, `11pt` and `12pt` respectively. Note: there is no `pt` in the options to prevent interference with possible class options. The default is the class option default or `10pt` in all other cases[1]. These options still exist for backward compatibility but are deprecated; they are being replaced by the `ptsize` option given below.

New Options

Some document classes (e.g., the `memoir` class) allow a wider range of font sizes than the standard classes (e.g., `9pt` or `14pt`) and might be handy for text used in figures. As noted above, this modified `pdftricks` package can automatically select only three sizes and will select `10pt` in all cases where it can't "decide" what the document class is selecting.

Two more options, that should be used in conjunction with each other and with caution, have now been added to this modified `pdftricks` package. These options are dangerous in the sense that if they aren't set properly the figure files won't compile—not a disaster but it should be noted.

The new options are chosen via the `key=value` technique. The options are:

`ptsize`:	Default value `10pt`. Note the `pt`; very different from the older options which didn't have the `pt`. This sets the optional argument of the `\documentclass` command to that point size for the figure files. If you use the this option the point size optional argument for the class will be *forced* to that value.
`docclass`:	Default value `article`. This sets the document class in the figure file to the chosen value.

where the default value of each option is the value it takes if no `=value` is given. E.g.,

```
\usepackage[ptsize,docclass]{pdftricks}
```

is the same as

```
\usepackage[ptsize=10pt,docclass=article]{pdftricks}
```

both giving a first line in the figure files of

```
\documentclass[10pt]{article}
```

and is different from

```
\usepackage{pdftricks}
```

which will use the `article` class but with the font size (`10pt`, `11pt`, `12pt`) used with the overlying document class; while

```
\usepackage[ptsize=9pt,docclass=memoir]{pdftricks}
```

will give the first line

```
\documentclass[9pt]{memoir}
```

for a smaller than "normal" figure font.

I can't emphasize too much that it is up to *you* to have the two options consistent.

---

[1] This includes the case where a class doesn't define the standard class options.

Batch Processing

If you don't like to enable shell escape you can call the pdftricks package with the `[noshell]` option. This will produce the `*-fig*.tex` (from now on called `pdftfig`) files and the common `tmp.inputs` file but won't attempt to process them into `pdf` figure files.

Enclosed are a shell script, `Pdft-fig2Pdf`, and a Dropscript, `DropPdft-fig2Pdf`, to batch process the `pdftfig` files.

The Shell Script

The script should work with any UNIX(-like) system. Simpy put the script in your `~/bin/` directory and make it executable. The command

```
~/bin/Pdft-fig2Pdf [*-fig*.tex -or- directory with *-fig*.tex files]
```

within the Terminal application will process all of the `pdftfig` files. If the list contains a directory then all the `pdftfig` files in that directory will be processed. The `[...]` above means one or more and both relative and absolute paths are accepted. The processing of a particular `pdftfig` file will be canceled if a `tmp.inputs` file doesn't exist in the same directory as the `pdftfig` file. Output to the terminal tracks the progress of the processing.

The Dropscript

The Dropscript is Mac OS X only. It also assumes that the PATH is correctly set by either `/etc/profile` or `~/.profile` since the internal script runs under the Bourne Shell; this is set properly in gwTEX.

The Dropscript behaves in a similar fashion but through the GUI. Drop one or more `pdftfig` files or directories containing those files onto the Dropscript and they will get batch processed.

A `Pdft-fig2PdfLog` file will be produced in each directory that has a processed `pdftfig` file. The Log file lists the processing of each of the files in that directory. The processing takes some time and it may be a while before the processing completes and the Finder updates the window for the directory[2]; you can help force the update by closing and opening the directory window. Make sure the Log file contains the `Processing Complete` message before getting panicked.

NOTE

Please put this revised `pdftricks` package in your personal texmf tree (`~/Library/texmf/tex/latex/`) until it has been thoroughly tested.

This version of the `pdftricks` package uses the original processing when used in Windows (e.g., `MikTeX`); i.e., if the `miktex` option is set the original processing steps are followed. In any case, I certainly cannot test it under Windows[3]. If you are trying to use this modified package under windows I'd advise against using any of the added options

Let me know if you have any problems with the modified package.

---

[2] Under `OS X 10.4.x` the Finder seems to update much faster; you can see all the intermediate files appear and disappear. The processing should be complete when you only have the original files, the `Pdft-fig2PdfLog` file and the final `.pdf` files for each of the figures processed present.

[3] The `large` and `total` versions of `MikTeX` both seem to have `pdftricks.sty` *and* `pdfcrop` so `pdftricks.sty` should be able to be adapted to that distribution but I have no experience using Windows.