Up & Running: PDFLaTeX in MikTeX 1.09

David Meeker

February 8, 1998

1 Configuration

I upgraded from MikTeX 1.08 to 1.09 just by downloading and unzipping the distribution files ending in 109 from CTAN. Since I already had the full set of Type1 CM fonts stored in a directory outside my `texmf` tree, I had to modify `\texmf\miktex\config\miktex.environment`, line 64 to reflect where I have my fonts. If the whole 1.09 distribution is unpacked as-is, then this modification is probably unnecessary. To ensure that everything is consistent with this change, run `configure.exe` in same directory.

I also discovered that I did not have copies of the files `supp-pdf.tex` and `supp-mis.tex` that are necessary to properly include graphics into PDFLaTeX files. They can be downloaded off of CTAN from the directory `macros/pdftex/graphics/`. They should be moved to `\texmf\tex\latex\base`, where the can be found by PDFLaTeX.

2 Running PDFLaTeX

A few special commands need to be put into your LaTeX document so that it will produce PDF correctly. In preamble (just before `\begin{document}`) you need the following:

`\pdfcompresslevel=9`	Requests full compression of the PDF
`\pdfoutput=1`	Tells the program to produce PDF instead of DVI

To get automatic links in your pdf, as well as a table of contents, use the package 'hyperref' (available from CTAN in `/macros/latex/contrib/supported/hyperref`). It is called with `\usepackage[pdftex]{hyperref}`, which should be the last package called before `\begin{document}`. Your document should be compiled at least three times in a row so that all the bookmarks are included properly in the PDF.

PDFLaTeX is simple to run; it runs just like LaTeX. Just substitute `pdflatex` for `latex` in the command line.

3 Graphics

To include graphics in your document, you can use the `graphicx` package by including `\usepackage[pdftex]{graphicx}` in the preamble. The `graphics` package can also be used in place of `graphicx`. So that the right graphics file extensions are searched, include
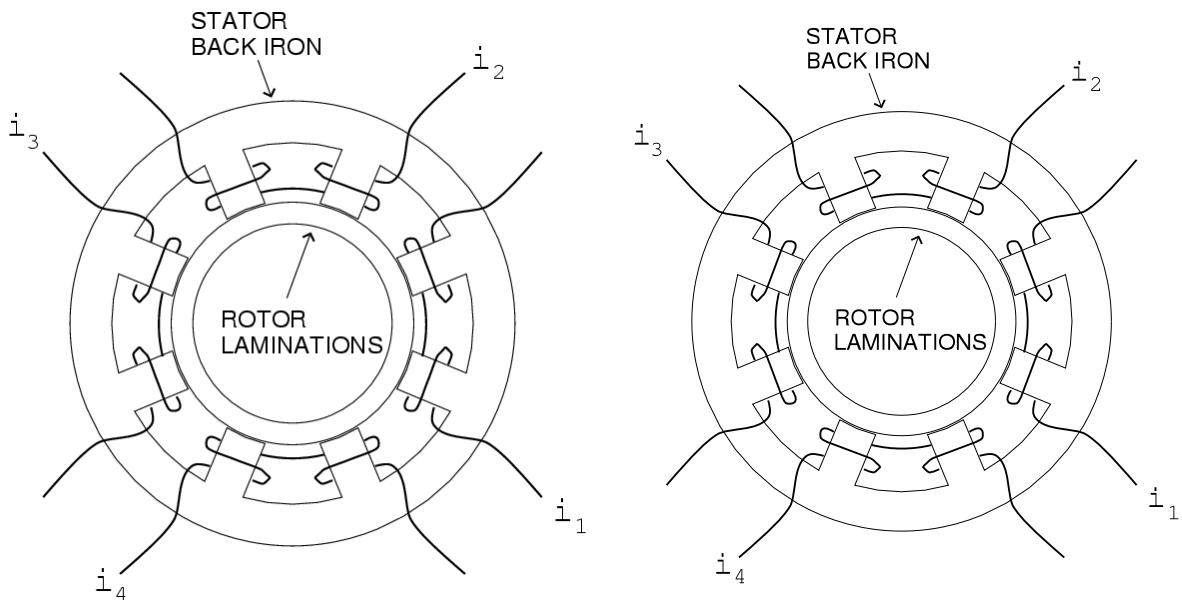
Figure 1: PNG and PDF graphics in PDFLATEX.

$$\verb|\DeclareGraphicsExtensions{.pdf,.png,.mps}|$$

as line after `\begin{document}`. Only PNG, PDF, or metapost figures can be used with PDFLATEX currently.

3.1 PNG Graphics

One way to create PNG directly from ghostview or gsview using the one of the png devices: `pnggray, pngmono, pnggray, png16, png256, png16m`. device. PNGs can also be created from GIFs using program gif2png available at

`ftp://ftp.leo.org/pub/comp/platforms/pc/msdos/graphics/graphutils/`

Yet another way to produce PNG files is through the use of the netpbm tools. If you haven't used them before, netpbm is a bunch of command-line tools to convert between virtually any image formats. The tool kit consists of two sorts of programs: ones that convert from a source graphics format to the intermediate PNM format; and ones that convert from PNM to the destination graphics format. An executable version of this toolkit is available as part of the web2c distribution at:

`ftp://ftp.duke.edu/tex-archive/systems/win32/web2c/netpbm-win32.tar.gz`

However, the `pnmtopng.exe` that is included with this distribution requires `libpng.dll`, which is not included in this archive–It's probably elsewhere in the distribution. An executable of `pnmtopng.exe` that doesn't require this `dll` is available at `http://www.hensa.ac.uk`.

PNG files are then included in the usual graphicx style, *e.g.* `\includegraphics{file.png}` They seem to be able to figure out their own bounding boxes.

An example of a PNG included in a PDF is shown on the left in Figure 1. This PNG was

produced by the netpbm route method. The original postscript figure was piped through ghostscript to convert to 72 dpi anti-aliased pgmraw format. Then, some netpbm tools (`pnmcrop` and `pnmtopng`) were used to crop the white space and convert to PNG. Although this procedure sounds convoluted, it can be put into a batch file to make the conversion painless.

3.2 PDF Graphics

High-quality drawings can be included by the use of PDF figures; however, the inclusion of PDF figures is somewhat tricky. PDF figures cannot have any text in them, or the resulting PDF output will be corrupted. A work-around is to use a program like `pstoedit` using the command line:

```
pstoedit -dt -f pdf figure.eps figure.pdf
```

which converts a postscript figure to PDF, and converts the fonts into a bunch of polygons that look the same as the regular fonts, are scalable, etc, but are not actually text. The latest source code for `pstoedit` is available at:

```
ftp://ftp.x.org/contrib/applications/pstoedit/pstoedit.html
```

For those who don't have a C++ compiler, I have put up my executable for version 2.60 that I compiled with the DJGPP C++ compiler (see `http://www.delorie.com`). It is provided in accordance with the terms of the GNU public license, which says that any executable is to be distributed with the full, unmodified source code. You can download it from

```
ftp://members.aol.com/dcm3cftp/p2e260.zip
```

However, I highly recommend that you compile your own version. My executable doesn't include some potentially useful destination formats such as WMF and CGM, nor does it include the GUI (that can only be built with Borland C++).

More caveats–the PDF files you create to use as figures must have the graphics positioned at the lower left lower left-hand corner of the page. Graphicx with PDFTeX only seems to understand bounding boxes for pdf files where the lower left corner has the coordinates (0,0). However, the location of the upper right corner can be adjusted via the usual bounding box option in graphicx. An alternative is to modify the /MediaBox line near the end of your PDF to reflect what bounding box you would like. Then, no bounding box needs be explicitly specified; graphicx will figure out the right one.

4 More Information

PDFTeX is an evolving program, so the best source may be the PDFTeX mailing list:

```
http://tug.cs.umb.edu/cgi-bin/lwgate/pdftex/
```

The entire mailing list archive can be downloaded via ftp at:

```
ftp://ftp.tug.org/mail/archives/pdftex/
```

To get more information about GSView and Ghostscript, try

$$\texttt{http://www.cs.wisc.edu/~ghost/index.html}$$

For MikTEX in general, there is the homepage at

$$\texttt{http://www.inx.de/~cschenk/miktex/}$$

In general, a lot of interesting stuff about PDF and Postscript lives at

$$\texttt{http://www.tinaja.com/acrob01.html}.$$

As an example, this document is also available in source format.