Linux, LaTeX and Lulu: Utilities, Tips, and Howtos

Soren Harward <sth@freeshell.org>

Introduction

In December 2005, I finished publishing an anthology of personal writings. This was a major project that took over 6 months and covered just about every major task you can do in desktop publishing, from writing text to drawing figures to page layout and typesetting to the technical nitty-gritty of generating PDFs that Lulu likes. I wrote this guide to help those that might be following in my path, and so that I don't have to retrace all my steps next time I do a project like this.

This guide is going to focus mainly on making PDFs that Lulu will print. Mostly, it's about how to make PDFs that satisfy the two most challenging demands that Lulu places on submitted PDF's: that they be the proper page size, and that they have all their fonts embedded properly. "Why bother going through all this trouble when I could just upload some other format and let Lulu convert it to PDF?" you ask? The huge advantage to submitting your own PDFs to Lulu is that there's no guesswork about how Lulu is going to print up the document that you've spent so long working to make. And this process is really not all that hard; it's just poorly documented. Until now.

Utilities You Should Have

All of these should be available for all major Linux distributions or other flavors of UNIX. They're also available for Windows as part of the Cygwin distribution.

Program	Version
LaTeX (TeTeX distribution)	2.0.2
Ghostscript	7.07.1
xpdf	3.00
ImageMagick	6.2.4.2

You don't necessarily need the same versions as I used. Just something close. You might also be able to use MiKTeX, a distribution of LaTeX for Windows, but I haven't tried this because I'm a Linux user.

LaTeX

LaTeX is well known in the technical writing and publishing field because there's no program that better handles complicated tasks like typesetting mathematics, generating tables of contents and indexes. But you don't have to be publishing a technical document to benefit from using LaTeX. It's a lot easier to handle a large project in LaTeX than in any traditional word processor, and the quality of LaTeX's output is second to none. It's what I used for all three of my 400-page books.

The **pdflatex** program is included with all modern distributions of LaTeX, and it's

the easiest way to create PDFs from LaTeX files. You can change the page size using LaTeX's "geometry" package, and by adding the "pdftex" option in the options you pass to the "geometry" package, **pdflatex** will create a PDF with properly-sized pages.

You may run into a little snag with font embedding and **pdflatex**: it doesn't embed the "14 Base Postscript Fonts" (such as Helvetica) by default. To change this, you're going to have to edit the file called "updmap.cfg", which is probably located in "/usr/share/texmf/web2c". Find the line that says

```
pdftexDownloadBase14 false
```

and change it to

```
pdftexDownloadBase14 true
```

You'll have to run the program **updmap** to get **pdflatex** to recognize the change. Now **pdflatex** will embed all your fonts for you, and Lulu won't complain any more about missing fonts. There's probably a similar procedure for MiKTeX.

LaTeX comes with a limited number of fonts available, and adding more is not trivial. However, once you manage to install them properly and make them available to **pdflatex**, then **pdflatex** will embed them properly in its output. So it's not easy to add new fonts like Arial, Myriad, and Garamond to LaTeX, but it's possible. And once you've done so you can use them in your Lulu documents. Some of the examples I show below used Myriad.

Ghostscript for Graphics

Ghostscript is a set of programs that manipulate Postscript files. It is very powerful, but unfortunately it is not the most user-friendly program around. Many Linux utilities use Ghostscript to handle their PS and PDF processing, so it helps to know how to get Ghostscript to do what you want it to.

The following gives an example of how you can create a PDF for your single-piece book cover. First, lay out the cover using a graphics program like The GIMP or Inkscape. When you're preparing the cover, Lulu tells you the required dimensions of the cover in points (72 per inch), so you have to convert that to pixels (300 per inch) if you're going to use The GIMP. Now export your graphic as a PNG (I'll call it "cover.png"). Now you can use ImageMagick to convert the PNG to a Postscript graphic by doing

```
convert -verbose -density 300 cover.png cover.ps
```

Note that the "-density" option should match the DPI that you created the PNG at, in this case 300 dpi. Then you use Ghostscript to convert the PS file to PDF:

```
ps2pdf -dPDFSETTINGS=/printer -dDEVICEWIDTHPOINTS=WWW \
-dDEVICEHEIGHTPOINTS=HHH cover.ps
```

Change WWW and HHH to the size (in Postscript points) that the book cover is supposed to be. For some odd reason, Ghostscript refuses to read the proper page size out of the Postscript file.

Even though ImageMagick can create a PDF directly from a PNG, there's apparently a bug in my version of ImageMagick that creates a spurious reference to the Helvetica font, and that made Lulu's parser very unhappy. This might be fixed in a future release of ImageMagick, so it's worth trying something like

```
convert -verbose -density 300 cover.png cover.pdf
```

and double-checking with **pdffonts** to see if ImageMagick converted it properly without any embedded fonts.

Verifying Everything is Okay

Your best friends for making sure your PDFs are ready to submit to Lulu are **pdfinfo** and **pdffonts**, which are part of the **xpdf** package. **pdfinfo** tells you how large the pages are:

```
Creator:        TeX
Producer:       pdfTeX-1.10b
CreationDate:   Fri Nov 25 03:00:00 2005
Tagged:         no
Pages:          415
Encrypted:      no
Page size:      432 x 648 pts
File size:      1536851 bytes
Optimized:      no
PDF version:    1.4
```

This is from a document with 6in × 9in pages, so 432pt × 648pt is the right size. Now let's use **pdffonts** to check that all the fonts are embedded properly.

```
name                                 type         emb sub uni object ID
------------------------------------ ------------ --- --- --- ---------
QFPFKG+Myriad-Bold                   Type 1       yes yes no       6  0
KMZKHF+Myriad-Roman                  Type 1       yes yes no       9  0
OQWKWN+NimbusRomNo9L-Regu            Type 1       yes yes no      16  0
GSVXED+NimbusMonL-Regu               Type 1       yes yes no      19  0
FIQVDS+Myriad-CnWeb                  Type 1       yes yes no      31  0
YQQGJJ+NimbusRomNo9L-Medi            Type 1       yes yes no      34  0
NRMQIA+NimbusRomNo9L-ReguItal        Type 1       yes yes no      43  0
```

As you can see from looking down the column labeled “emb”, all of them have been embedded properly. This document is ready to send to the press.

If you don't have all the fonts embedded properly, then you'll get output like

```
name                                 type         emb sub uni object ID
------------------------------------ ------------ --- --- --- ---------
SIJIRL+Myriad-Bold                   Type 1       yes yes no       6  0
JBOKJH+Myriad-Roman                  Type 1       yes yes no       9  0
ABLJHV+NimbusRomNo9L-Regu            Type 1       yes yes no      16  0
YZCBHM+NimbusMonL-Regu               Type 1       yes yes no      19  0
DHYIXU+Myriad-CnWeb                  Type 1       yes yes no      25  0
TVHBDB+NimbusRomNo9L-Medi            Type 1       yes yes no      28  0
UKAALS+NimbusRomNo9L-ReguItal        Type 1       yes yes no      34  0
ACHVYI+StandardSymL                  Type 1       yes yes no      43  0
VFCRTG+CMSY10                        Type 1       yes yes no     126  0
Helvetica                            Type 1       no  no  no     213  0
HRQVSS+Helvetica Bold                Type 1C      yes yes no     214  0
```

See how “Helvetica” didn't get embedded? You need to figure out where it's being used, and what isn't embedding it properly. You may have forgotten to make the changes to updmap.cfg, or to run **updmap** afterwards. Or you might be including a graphic as a PDF, and the PDF does not have that font embedded in it (which is the case here). You will need to re-create the graphic and make sure that all fonts get embedded in it.