

0.1 Using different fonts in LyX

by ROBIN TURNER



LyX currently uses a fairly limited selection of fonts. While it may come as a disappointment to people used to the thousands of TrueType fonts available to conventional word processors, there are in fact good reasons for this choice. LyX actually has no fonts of its own; for the screen it uses whatever fonts are available to the X-window system, and for output it uses the fonts of the TeX setup it is operating with. The selection of default fonts correspond to the most useful and commonly-found fonts available in TeX. The “L” at the beginning of this section is a case in point: most TeX distributions include the `yinit` font, so you should see a nice decorated “L” at the beginning; some distributions may not, so you won’t.

Another point is that the profusion of fonts which TrueType brought about has generally had a bad effect on desktop publishing, leading to documents with inappropriate, badly-scaled or simply too many fonts. You should therefore think carefully before using a different default font, or mixing font families in the same document.

Nevertheless, you can, with a little Evil Red Text, make LyX produce just about any font you want, and even design your own fonts. What follows is a “quick fix” guide; it is no substitute for reading a good book on (La)TeX.

0.1.1 Changing the default font

The easiest and most reliable way to change the default font is from within LyX, through **Layout**▷**Document**▷**Fonts**. The next easiest is to use a package contained in your TeX distribution. For example, the rather pretty Pandora font family can be accessed simply by putting `\usepackage{pandora}` in your L^ATeX preamble (**Layout**▷**LaTeX preamble**). Similarly, `\usepackage{oldgerm}` in the preamble gives you access to Yannis Haralambous’ Old German (Gothisch) fonts via the specially-defined `\gothfamily`, `\swabfamily`, and `\frakfamily` commands, which you insert in your document (as TeX) to obtain the desired font (note that nothing will change in your screen display, which considering the difficulty involved in reading some of these fonts is just as well). Other useful packages are `concrete`, which gives access to both the Concrete Roman and the Euler math fonts, and `chancery`, which gives you the Zapf Chancery font

A less reliable method is the `\familydefault` command. If you have a font installed and know its family name, you can put something like

```
\renewcommand{\familydefault}{pzc}
```

in the preamble (“pzc” is the family name for the Zapf Chancery font). You should be aware, however, that this method may produce strange results if the font you have selected does not have the sizes or shapes you want (e.g., adding

emphasis to text has no effect on Zapf Chancery, and choosing the sans serif or typewriter series will cause it to revert to the standard font).

0.1.2 More than one font family in one document

As I've said, different font families in the same document can spell trouble. Unlike many human families, the members of a font family work well together, so the eye is not overly disturbed when changing from a medium roman to a bold sans serif font, for example. Different font families may not have this visual compatibility, and clashing fonts are a common reason for amateur publishing looking amateurish. (End of sermon.)

Nevertheless, you may need different fonts for some reason: maybe you have to include a different alphabet, like Klingon or Cuneiform¹, or you want a structural/aesthetic effect, such as putting poetry in a different font from prose, like this:

```
Myself when young did eagerly frequent  
Emacs and vi and heard great Argument
```

One way to do this is to declare your own font commands. For example, writing

```
\newfont{\avant}{pagd}
```

will define a command, `\avant`, which can be invoked anywhere in your document to change to the **Avant Garde font**. You can return to your default font by typing `\normalfont`.

Some fonts will perform adequately with just the basic font name (as above); others require that you specify a font size; for example, the verse above, aside from mangling Omar Khayyam, used the font command `eurm8` (i.e. Euler Roman 8pt).

Some fonts may also need scaling using the `scaled\magstepn` option, since their idea of, say, 10pt, may not be the same as that of your default font family. If the font looks too small, try a command like this:

```
\newfont{\zapf}{pzci10 scaled\magstep1}
```

This scales the rather small-looking Zapf Chancery font so that it looks the same size as the default (10pt) font. If your font looks too big, select a smaller fontsize (e.g., 8pt) and scale up.

0.1.3 Finding new fonts

So where do you find all these exciting new fonts? There are three main methods:

1. Look around in your \TeX distribution.

¹These can be downloaded from CTAN. Remember, though, that some “normal” fonts can be turned into a different alphabet by changing the font encoding; see a good \LaTeX book if you want to do this.

2. Download T_EX fonts.
3. Convert TrueType fonts to T_EX.

The first stage is as far as most people will want to go. Have a look in `$TEXMF/fonts/tfm` to see what you've got (where `$TEXMF` is the location of your T_EX system, usually something like `/usr/share/texmf`). When you see something that looks promising, test it in a L_AT_EX document using the method described above, i.e. make a new font command and use it on some text, then preview and see what, if anything, you get. Note that the font name you want is usually the first four letters of the file name, e.g., a file named `pncb8a.afm` is actually the bold (b) version of Adobe's (p) New Century (nc) family. If in doubt about names, check out the files in `$TEXMF/fontname`.

T_EX fonts can be downloaded from your nearest CTAN mirror (e.g., `ftp.ivorytower.edu/pub/tex/ctan/fonts`). At first, look for font directories which include a `.sty` file which you can invoke in your preamble, as this will make life easier (as in the case of Pandora). Failing that, look for directories which give you the full set of T_EX font files. Failing *that*, look for an `.afm` file and run the program `fontinst` on it (read, or at least skim, the manual first—you've probably got it in `$TEXMF/doc/fontinst/base/fontinst.dvi`).

If all you have is an `.mf` file, you need to run `METAFONT` on it., which is beyond the scope of this guide: read *The METAFONT book* or Normal Walsh's *Making T_EX Work*, Chapter 11.

You should now have a directory filled with downloaded or converted files; now you need to put the font files in the right places. Unless you're running Windows 9* or have an entire T_EX system in your home directory, you need to be root to do this. The basic rule is to look at the extension of the file and move it to the corresponding directory, and put `.sty` files in the `/latex` directory:

- `.sty`, `.cls` or `.fd` → `$TEXMF/tex/latex/<package_name>/`
- `.dvi`, `.ps` or `.pdf` → `$TEXMF/doc/latex/<package_name>/2`
- `.tfm` → `$TEXMF/fonts/tfm/<supplier>/<font_name>/`
- `.vf` → `$TEXMF/fonts/vf/<supplier>/<font_name>/`
- `.afm` → `$TEXMF/fonts/afm/<supplier>/<font_name>/`
- `.pfb` → `$TEXMF/fonts/type1/<supplier>/<font_name>/`
- `.ttf` → `$TEXMF/fonts/truetype/<supplier>/<font_name>/`

If this sounds confusing, check out <http://www.ctan.org/installationadvice/>.

Thought you were finished? Not quite. First you need to hack the file `$TEX/dvips/base/psfonts.map` (there is a script to do this in a more orderly

²These will be documentation files for the font, not fonts themselves.

manner if you want—look at the comments at the top of the file for details).³ You need to add a line describing your new font so that `dvips` can recognise it. Look at the other lines to get an idea of the format. The basic format is:

```
TeX_name Real_name <encoding (optional) <pf filename.pfb>
```

If you're not sure what to write here, run `afm2tfm` on an `.afm` file, and enter what comes up in the terminal.

Now run `texhash` (as root, of course), or `initexmf -update-fndb`, if you're using MikTeX on Windows.

If that wasn't enough for you, you may want to try converting your favourite TrueType font into a form that L^AT_EX can get at. It's a long, tiring process, but if you really want that font, it's worth it. First download the program `ttf2pt`, run it on your font, then run `fontinst` on the output, and proceed as before. Have a look at <http://www.pegasus.rutgers.edu/~elflord/unix/latex/no-bs.html>.

0.1.4 Making your own fonts

There are (at least) three ways to do this.

The proper way Set aside some time—maybe a summer vacation. Get a copy of *The METAFONT book*. Learn the METAFONT program. Produce beautiful fonts.

A quick kludge Get a TrueType font editor. Make a TrueType font. Convert to T_EX. This may work, but may produce rather ugly results.

A very quick kludge If all you want is a few words (e.g., for a heading), rather than making an entire font, write what you want by hand and scan it, then insert it as a graphic. You could in theory make a whole alphabet like this and insert each letter as an inline `.eps` file, but it would be tedious, and probably not very aesthetic.

0.1.5 Recommended reading

For a good overview of fonts in L^AT_EX:

Michel Goossens, Frank Mittelbach and Alexander Samarin: *The L^AT_EX Companion*. Addison-Wesley, 1994

For more technical information about fonts and some samples:

Norman Walsh: *Making T_EX Work*. O'Reilly, 1994

If you really, really want to design fonts:

Donald Knuth: *The METAFONTbook*. Addison Wesley, 1986.

³If you don't want to, or can't change the system-wide font map, you can create a `.dvipsrc` file in your home directory which will add your own font lines. See Norman Walsh's *Making T_EX Work*, Chapter 5.