

B Installing new fonts

Directories (Folders)

In the examples below, all the folders (directories) are assumed to be in your Personal T_EX Directory unless otherwise explicitly given. See [section A.2 on page 217](#) for how to find this out.

Different fonts come in a variety of packagings: the most commonly used with T_EX systems are *PostScript* fonts and METAFONT fonts, but X_YL_AT_EX also lets you use TrueType and OpenType fonts. How you install them and where they go depends on how you installed L^AT_EX: all I can deal with here are the standard locations within the TDS. These typefaces come supplied as one or more font 'outline' files and a number of ancillary files:

METAFONT typefaces: have a number of `.mf` source (outline) files, possibly also some `.fd` (font definition) files and a `.sty` (style) file. The `.tfm` (T_EX font metric) files are not needed at installation, as they get generated from the outlines automatically the first time you use the font.

PostScript typefaces: come as a pair of files: a `.pfb` (PostScript font *binary*) or `.pfa` (PostScript font *ASCII*) outline, and an `.afm`

(Adobe font metric) file. There may also be `.inf` and other files but these are not needed for use with \TeX systems.

TrueType and OpenType typefaces: are a single `.ttf` or `.otf` file, which combines outlines and metrics in one.

The instructions for Type 1 and METAFONT typefaces here assume the use of the New Font Selection Scheme (NFSS) used in $\LaTeX 2_{\epsilon}$. If you are running the obsolete $\LaTeX 2.09$, upgrade it now, because none of this will work.

As TrueType and OpenType are the easiest to deal with, I'll mention them first.

B.1 TrueType and OpenType fonts

This is the simplest, as they are already recognised by your computer, so you just install them in the normal way for your computer system (usually double-click on the font file and select `Install` from the menu).

On Windows systems, the fonts are made available automatically; on UNIX & GNU/Linux systems, including Apple Macintosh OS X, run the `fc-cache` utility to give \TeX fast-load access to all the fonts you have installed. Run the program like this:

```
fc-cache -fv
```

It can take several minutes, especially if you have a lot of fonts. Once it has finished, you can query the database with the `$fc-list` command and `$grep` for the string you want:

```
fc-list|grep -i comic|sort
/home/peter/texmf/fonts/type1/comicsans/rcomic8r.pfb: Comic Sans MS:
/home/peter/texmf/fonts/type1/comicsans/rcomicbd8r.pfb: Comic Sans M
/usr/share/texlive/texmf-dist/fonts/type1/rozyński/comicneue/ComicNe
...
```

B.2 Installing METAFONT fonts

This is the simplest installation. When you download METAFONT fonts from CTAN, you'll usually find a number of outline files (`.mf` files) and maybe some other types as well (see below).

Installation of METAFONT fonts

1. In your `Personal TeX` directory, create a new subdirectory called `fonts/source/public/name` named after the typeface you're installing¹
2. Copy all the `.mf` files to this directory;
3. Copy any `.fd` file[s] to your `tex/latex/mfnfss` subdirectory (create it if it doesn't already exist);
4. Copy any `.sty` (style) files to a subdirectory (create it too), named after the typeface, eg `tex/latex/name`;
5. If you are using MiKTeX, run your TeX indexer program (see step 4 on page 64).
UNIX & GNU/Linux systems, including Apple Macintosh OSX, don't need this last step.

That's it. Unlike *PostScript* fonts, METAFONT fonts generate their font metric files (`.tfm` files) automatically on-the-fly the first time the font is used, so there should be nothing else to install.

Now you can put a `\usepackage` command in your Preamble with whatever name the `.sty` file was called, and read the documentation to see what commands it gives to use the font (refer to section 3.2.1.2 on page 60 and step 2 on page 62).

If the font came *without* `.fd` or `.sty` files, you'll need to find someone who can make them for you (or follow the outline in section B.3 on page 231, step 8 on page 238).

B.3 Installing PostScript fonts

Lots of people will tell you that *PostScript* fonts and PostScript output are dead and that TrueType or OpenType fonts and PDF output are the way to go. While this may be true for some cases, standard L^AT_EX does

¹ On UNIX & GNU/Linux systems, including Apple Macintosh OSX, the easiest way to do this is in a Terminal window, in your Personal TeX Directory, using the command `$mkdir -p fonts/source/public/whatever`, as this creates any intervening subdirectories for you. Under Windows, you have to create each subsubdirectory individually.

not work with TrueType fonts and does not produce PDF directly. Only $X_{\text{L}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and $\text{PDFL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ do that, and there are still many printers whose typesetters and platemakers require *PostScript* files rather than PDFs. In addition, operating system support for scalable fonts is still very poor on Unix and GNU/Linux systems, despite the advances in recent years, and many rebranded ('knock-off' or pirated) TrueType or OpenType fonts supplied with other systems are of very poor quality. So in many cases it still makes sense to use $\text{T}_{\text{E}}\text{X}$'s built-in support for PostScript fonts. When $X_{\text{L}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ becomes the default processor (it's currently still $\text{PDFL}^{\text{A}}\text{T}_{\text{E}}\text{X}$), it will no longer be necessary to install different types of font files separately, as $X_{\text{L}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ is able to use all fonts natively from your system's font folders.

Two files are needed for each *PostScript* Type 1 font: the `.afm` Adobe Font Metric (AFM) and the `.pfb` PostScript Font Binary (PFB) files. *You must have both types of file for each separate font before you start.* If you only have the near-obsolete `.pfa` PostScript Font ASCII (PFA) files, it may be possible to generate the `.pfb` files using the `t1binary` program from the `t1utils` suite (see gnuwin32.sourceforge.net/packages/t1utils.htm) or the excellent *FontForge* font editor (from fontforge.sourceforge.net). There are unfortunately still some companies distributing Type 1 fonts in `.pfa` format (Mathematica is one reported recently).

I'll repeat this: before you start, make sure you have all the `.afm` and `.pfb` files for the typeface you want. In the example below, I'm going to use a single font from an imaginary typeface called Foo, so I have `foo.afm` and `foo.pfb` files.

Installation of *PostScript* Type 1 fonts: Preparation

This part is very quick, but needs explaining.

1. **Put the files into a temporary directory**

This is `/tmp` on Unix & GNU/Linux, including Apple Macintosh OSX; `C:\tmp` or `C:\temp` on Windows 95-XP; and should be `Computer\System\Users\your_name\tmp` on Windows 7 and up.

2. **Decide on the short font name to use inside $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.**

This is *not* the full descriptive name (eg Baskerville Italic Bold Extended) but an encoded font name in the format `fnnssseev`,

devised by Karl Berry, which stores the same information in no more than eight characters for compatibility with systems which cannot handle long filenames (and incidentally makes it far easier to type). The letters in the format above have the following meanings (see the `fontname` documentation on your computer for more details). Lists of the codes used are in the files `supplier.map`, `weight.map`, `width.map`, `variant.map`, and the various `.map` files for each foundry, which are in your TeX installation.

Letter	Meaning	Examples
<code>f</code>	foundry	<code>b</code> =Bitstream, <code>m</code> =Monotype, <code>p</code> =Adobe
<code>nn</code>	typeface name	<code>ba</code> =Baskerville, <code>tm</code> =Times, <code>pl</code> =Palatino
<code>ss</code>	series/shape	<code>r</code> =roman, <code>b</code> =bold, <code>i</code> =italic, etc
<code>ee</code>	encoding	<code>8r</code> =TeXBase1, <code>8t</code> =Cork
<code>v</code>	variant	smallcaps, outline, script, etc

The `fonts/map/fontname` directory in your main (*not* local) installation directory of TeX has files for several foundries giving fully-formed names like these for common fonts (eg `ptmr8r` is [Adobe] PostScript Times Roman in an 8-bit revised TeX encoding; `bgsllly` is Bitstream Gill Sans Light in Y&Y's TeX'n'ANSI encoding [LY1]).² Read the documentation in *Fontname: Filenames for TeX fonts* to find out how to make up your own short names if the foundry and font you want is not shown in the lists in the `fonts/map/fontname` directory.

In this example we'll call our mythical example typeface 'zork' (standing for Zfonts Ordinary Bookface: `z` for a foundry otherwise undefined; `or` for Ordinary; and `k` is the letter used for Book designs, `b` being already the code for bold). We'll assume the font comes in the two files `foo.afm` and `foo.pfb` that I mentioned above.

While the `font/map/fontname` directories have ready-made maps of these names for popular collections of typefaces, making

² Confusingly, Bitstream fonts (and others from similar sources) mostly have different names from the original fonts, to avoid copyright issues, so what they call Humanist 521 is actually Gill Sans. Until recently, US law only allowed the *names* of typefaces to be copyrighted, not the font designs themselves, leading to widespread piracy.

them up requires some knowledge of typographic terms and a careful reading of the *fontrname* documentation.

3. Decide on your encoding

Encoding is needed because Adobe fonts store their characters in different places to the T_EX standard. This is what tripped me up the first few times until someone pointed me at Y&Y's³ T_EX'n'ANSI encoding which at the time was the only one that included the glyphs I want where I expected them to be.⁴ Now, however, I recommend using the 8r encoding for all PostScript fonts. The encoding vector file `8r.enc` should be in your main (*not* local) T_EX installation directory in `fonts/enc/dvips/base`.

To avoid having to type the long path each time below, just copy this file to the temporary directory where you're doing all this stuff.

Now you're ready to convert the files to T_EX format.

Installation of *PostScript* Type 1 fonts: Conversion

1. Convert the `.afm` files to `.tfm` and `.vpl` format

The Adobe Font Metric files have to be converted to T_EX Font Metric and Virtual Font files. The *afm2tfm* and *vptovf* programs are standard T_EX utilities in the `bin` directory of your *main* T_EX installation.

In a command window, we type:

```
afm2tfm foo.afm -v zork8r.vpl -p 8r.enc rzork8r.tfm >zork.id
```

This reads `foo.afm` using the `8r.enc` encoding file, and creates a special 'raw' T_EX Font Metric file (hence the `r` prefix on `rzork8r.tfm`) that L^AT_EX can use, with a list of all its properties encoded with 8r in the `.vpl` or Virtual Property List file.

³ Y&Y, Inc has ceased trading and their T_EX distribution is not longer available, although there is email support at lists.ucc.ie/lists/archives/yandytex.html, and their encoding files continue to be used.

⁴ The only one I had problems with is 'Å', which for some weird reason isn't catered for in this encoding.

Many people will tell you that virtual fonts are dead and that this is the wrong way to do it, but no-one has ever shown me an alternative that works, so I stick with it.

2. Small caps (optional)

If you want a small caps variant faked up (perhaps because the typeface family doesn't have a real small-caps font), repeat the medicine like this:

```
afm2tfm foo.afm -V zork8rc.vpl -p 8r.enc rzork8r.tfm >>zork.id
```

Note the *capital V* option here. Yes, it *does* overwrite the `rzorkly.tfm` created in the first command — it doesn't matter; let it (it will be the same information, anyway). And those are *two* 'greater-than' signs before the `zork.id` filename because we want to append to it, not overwrite it.

3. Create the virtual font

The `vptovf` program turns the `.vpl` files into `.vf` and `.tfm` pairs. \LaTeX uses these to map the character positions from Adobe's encoding to its own.

```
vptovf zork8r.vpl zork8r.vf zork8r.tfm  
vptovf zork8rc.vpl zork8rc.vf zork8rc.tfm
```

Now we can install the files.

Installation of *PostScript* Type 1 fonts: Installation

1. Make directories to hold the files

Under your Personal \TeX Directory, create a `fonts` subdirectory, and in it, create an `afm`, `tfm`, `type1`, and `vf` subsubdirectories (they may already exist from a previous font installation);

2. In each of these four, create a subdirectory for the foundry, making up a name if it's not an established one (see the `suppliers.map` file);

3. Within each of the foundry directories, create a directory for the typeface (using an abbreviated human-readable typeface name,

NOT the short Karl Berry fontname). On my computer, this means doing:

```
cd ~/texmf/fonts
mkdir -p afm/zfonts/zork
mkdir -p tfm/zfonts/zork
mkdir -p type1/zfonts/zork
mkdir -p vf/zfonts/zork
cd /tmp
```

Under Windows, where the `$mkdir` has no `/p` option, you probably have to create each subsubdirectory individually.

4. Copy the files to their rightful places

Copy the four groups of files to the four new directories:

```
cp *.afm ~/texmf/fonts/afm/zfonts/zork/
cp *.tfm ~/texmf/fonts/tfm/zfonts/zork/
cp *.pfb ~/texmf/fonts/type1/zfonts/zork/
cp *.vf ~/texmf/fonts/vf/zfonts/zork/
```

where `~/texmf` is the root of your Personal TeX Directory. You can of course do all this with a directory window and mouse if you find it easier.

Last stage is to create the font map, style file, and font definition files.

Installation of *PostScript* Type 1 fonts: Configuration

1. Create a font map

The font map is what tells TeX's font driver which PFB file to use for which font.

Open your editor and create a new file (if it asks for a name, call it `zor.map`);

- Font entries MUST be on a *single* line each, with no line-breaking. Each entry has five fields separated by a space: *a*) the short fontname; *b*) the full (Adobe) font name; *c*) the PostScript encoding parameters (in quotes, with a leading and trailing space); *d*) the encoding file used; and *e*) the name of the PostScript outline

file. Fortunately, we captured almost all of this in those `.id` files in step 2 on page 235, so all you need to do is copy all those `.id` files into this file (eg `$cat *.id >>zor.map`), and add a space and the less-than sign and the name of the appropriate `.pfb` file to the end of each line, eg

```
rzork8r ZorkOrdinary-Book " TeXBase1Encoding ReEncodeFont " <8r.enc <zork.pfb
```

3. Save the file in `fonts/map` in your Personal \TeX Directory;
4. Update the map file into your \TeX 's file maps with the command:

```
updmap --enable Map=zor.map
```

5. Create a style file

\LaTeX needs a style file to implement the interface to the font. Call it after the typeface or something related; in this example we'll call it `foozork.sty` and it needs to be saved into `tex/latex/zork/` in your Personal \TeX Directory.

In it go some details of the name and date we did this, what version of \LaTeX it needs, and any other command necessary to operate the font, like the font encoding and whether it is to supersede the current default Roman font.

Use your editor to open (create) this file;

6. Insert the following lines (obviously replacing the `foozork` stuff with meaningful values for your own fonts:

```
% fozork - LaTeX style file for the Zork font
\def\fileversion{1.0}
\def\filedate{2016/12/03}
\def\docdate{2016/12/03}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{foozork}[\filedate\space
\fileversion\space
Zfonts Ordinary PSNFSS2e package]
\RequirePackage[T1]{fontenc}
\renewcommand{\rmdefault}{zor}
\endinput
```

Note the following:

- The first argument to `\ProvidesPackage` MUST be the same as this style file name.
- If this is a typewriter font, change the renewed command `\rmdefault` into `\ttdefault`. If it's a sans-serif font, make it `\sfdefault` instead.
Omit this command completely if you don't want the style file to supersede the current defaults but simply to make the font available for manual use.
- If you do that, you probably want to write a new command or two to use it, typically one unscoped command for grouped use and one scoped one for argument use:

```
\newcommand{\zork}{\fontencoding{T1}%  
  \fontfamily{zor}\selectfont}  
\newcommand{\textzork}[1]{\zork#1}}
```

7. Save and close the file.

8. Create the Font Definition file

The last file needed is the *font definition* (`.fd`) file. This is named following the pattern `eefnn.fd`, using the same conventions as before, by prepending the (lowercase) encoding abbreviation to the foundry letter and fontname abbreviation, so our example would be `t1zor.fd` for the T1 (8r) encoding and the `zor` short fontname.

Use your editor to create the file in `/texmf/tex/latex/zork` (replacing `zork` with the name of the directory you created there).

9. Enter the following lines (see the notes below about what they mean):

```
\ProvidesFile{t1zor.fd}[2016/12/03 v0.1 manual  
  font definitions for T1/zor.]  
  
\DeclareFontFamily{T1}{zor}{}
```

```
\DeclareFontShape{T1}{zor}{k}{n}{<-> zork8r}{}  
\DeclareFontShape{T1}{zor}{k}{sc}{<-> zork8rc}{}
```

10. Save and close the file.

Now you can `\usepackage{foozork}` in your \LaTeX file to make it the default font. To use the font incidentally instead of as the default, you can use the commands you added at the end of step 5 on page 237:

```
This is {\zorkfamily ZORK} or \textzork{ZORK}
```

B.3.1 Font Definitions

FD files typically use one `\DeclareFontFamily` command which specifies the encoding and the short font name. This is followed by as many pairs of `\DeclareFontShape` commands as you converted fonts (assuming you did both normal and small caps for each font: see step 2 on page 235; if you didn't, then only one such command per font is needed here). The arguments to the `\DeclareFontShape` command to watch are the 3rd (weight/width, here `k` for Book, remember?), 4th (shape), and 5th (the size range and the name of the entry in the font map, minus the `r` prefix): the rest are static for each `.fd` file and simply identify the encoding and the font family.

The codes to use are given on pages 414–15 of the *Companion* and should also be in your copies of `weight.map` and `width.map`. The rules for combining weight and width need care: read the documentation for the `fontname` package. There is no `shape.map` in `fontname` because it's not part of font file names, it's purely a \LaTeX creation, so here's what the same book says:

Character	Meaning
<code>n</code>	normal (upright)
<code>it</code>	italic
<code>sl</code>	slanted
<code>sc</code>	small caps
<code>ui</code>	upright italic
<code>ol</code>	outline

Add your own for other oddities, but be consistent: I use `cu` for cursive (scripts), for example, and `k` for blackletter faces (not to be confused with `k` as a *width* for 'book').

The default size range `<->` in the fifth argument of the command `\DeclareFontShape` means that all sizes are to come from the same font outline (remember if this was a METAFONT font with different design sizes like `CM` it would be much more complex).

If the face has only a few variants, you can create any other entries for bold, italic, slanted, etc with the relevant weight and width and shape values pointing at the relevant outline file.

If you want one font to substitute for a missing one (for example italics to substitute for slanted in a typeface which has no slanted variant of its own) give the `ssub` ('silent substitution') command in the fontspec: for example to make all references to `sl` (slanted) type use an existing italic font, make the 5th argument like this:

```
{<-> ssub * zor/k/it}
```

If you find the x-height of a font too big or too small to sort well with another font you are using, you can specify an `s` ('scale') factor in this argument instead: this example will shrink the result to 80% of normal:

```
{<-> s * [0.8] zork8r}
```

B.4 Updating your font maps

Because \LaTeX can handle so many different types of font file, it needs to know what to do when you use a particular typeface. We already saw in [section B.1 on page 230](#) how to re-index your system's font cache (Unix and GNU/Linux). That tells \LaTeX *where* to find a font; the font map tells it *what to do with it* when it's found it.

Every font family or typeface prepared for use with \LaTeX MUST have a `.map` file to do this with, *except* METAFONT fonts, which don't need them (we saw in [step 4 on page 237](#) how to install the map file we created for a Postscript fonts).

B.4.1 Updating font maps on Mac and Linux (and T_EX Live on Windows)

In Unix and GNU/Linux systems (including Mac OS X) it's very simple, as we saw in step 6 on page 159:

```
updmap --enable Map=xxxxx.map
```

where xxxxx is the name of the font map. This updates your local (personal) font mappings. If you need to do this on a shared machine (eg in a lab or library), you need the Admin or root password, and use the command

```
updmap-sys --force --enable Map=xxxxx.map
```

B.4.2 Updating font maps in MiK_TE_X

This procedure comes from the T_EX Users Group page at www.tug.org/fonts/fontinstall.html.

1. Edit the map configuration file `updmap.cfg` in a DOS/Command Prompt window; type the command:

```
initexmf --edit-config-file updmap
```

Edit this file in a text editor such as *Notepad*.

2. Add this one line to `updmap.cfg` and save it:

```
Map newfont.map
```

3. Back at the DOS/Command prompt, type:

```
initexmf --mkmaps
```

Ignore any error messages.

