Building virtual fonts with 'fontinst'*

Alan Jeffrey

alanje@cogs.susx.ac.uk

Abstract

This document gives a brief overview of the `fontinst` package. The `fontinst` package is used to build *virtual fonts* (VFs) which allow PostScript fonts to be used as drop-in replacements for the Computer Modern fonts in TEX.
Below, I'll describe VFs briefly, and describe how they can be built using the `fontinst` package.

1 A problem with fonts

One of the biggest problems about using fonts in TEX is *encodings*, that is the order the characters come in the font. For example, the default encoding for Adobe's Times-Roman font is the 'Adobe Standard encoding':

```
  ! " # $ % & ’ ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
‘ a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~
  ¡ ¢ £ ⁄ ¥ ƒ § ¤ ' “ « ‹ › fi fl
  – † ‡ · _ ¶ • ‚ „ ” » … ‰   ¿
  ` ´ ˆ ˜ ¯ ˘ ˙ ¨   ˚ ˝ ˛ ˇ
—
  Æ   ª       Ł Ø Œ º
  æ     ı     ł ø œ ß
```

The default encoding for TEX, however, is the 'TEX text encoding'. The Adobe Times-Roman font in the 'TEX text encoding' is:[1]

```
Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω ff fi fl ffi ffl
ı ■ ` ´ ˘ ¯ ˚ ¸ ß æ œ ø Æ Œ Ø
■ ! ” # $ % & ’ ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; ¡ = ¿ ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ “ ] ^ ˙
‘ a b c d e f g h i j k l m n o
p q r s t u v w x y z – — ˝ ~ ¨
                    Ł
                    ł
```

There are many other competing encodings: 'ISO Latin-1', 'TEX extended text' (or 'Cork'), 'Macintosh', the list is seemingly endless.

In addition, different encodings contain different glyphs. The TEX text encoding is supposed to contain a dotless 'j' character, and a slash for building 'ł' and 'Ł', but very few fonts contain these characters, and their places are taken by black squares above.

The problem of incompatible font encodings is addressed in TEX by *virtual fonts*.

2 A solution: virtual fonts

As far as TEX is concerned a virtual font (VF) is a font like any other. It has a TEX font metric file, which contains the dimensions of each character, together with ligatures, kerning, and other typographical information.

However, a VF does not have an associated bitmap, Type 1 font, TrueType font, or other information about what the font should look like.

Instead, a VF has an associated `.vf` file, which contains a small fragment of `DVI` file for each character in the font. This `DVI` fragment may contain characters from other fonts, rules or `\specials`.

For example, the 'Adobe Standard' encoded Times-Roman font above is a 'raw' Type 1 font, but the 'TEX text' encoded Times-Roman font is a virtual font.

- The 'ff', 'ffi' and 'ffl' ligatures are faked by putting an 'f' next to an 'f', 'fi' or 'fl'.
- The missing 'dotless j' and 'ł-slash' are rules, together with a 'Warning: missing glyph' `\special`.
- The Greek upper case come from the Symbol Type 1 font.
- The other characters come from the Times-Roman Type 1 font.

Any `DVI` driver which understands VFs and can use Type 1 fonts can use the TEX text Times-Roman VF as a drop-in replacement for Computer Modern.

*Reprint from the Annals of the UK TEX Users' Group: **Baskerville**, Volume 4.1, Januari 1994.

[1] The TEXnically minded may note that the glyphs 'ł' and 'Ł' are not normally in the 'TEX text encoding'. This is because Computer Modern has a special 'ł-slash' glyph for building 'ł' and 'Ł', which Adobe Times-Roman does not have. Its place is therefore taken by a black square, and there are ligatures with 'l' and 'L' to produce 'ł' and 'Ł'. Thus this font is drop-in compatible with Computer Modern, despite the lack of an 'ł-slash' glyph.

3 A problem with virtual fonts

One stumbling block about using VFs is that they are not very easy to generate. Despite having been in existence for four years, there are very few tools for creating VFs.

The most important tool is Knuth's `vptovf`, which converts *Virtual Property Lists* (vpls) into VFs. Unfortunately, the vpl language is rather opaque; for example the vpl code for the Adobe Times character 'ff' is:

```
(CHARACTER D 11
   (CHARWD R 6.47998)
   (CHARHT R 6.81995)
   (CHARDP R 0.0)
   (MAP
      (SELECTFONT D 1)
      (SETCHAR D 102)
      (MOVERIGHT R -0.17993)
      (SETCHAR D 102)
      )
   )
```

Editing vpl files by hand is something of a black art, and there are few tools for manipulating them.

The main tool for generating vpls is Rokicki's `afm2tfm`, which converts the *Adobe Font Metric* (*AFM*) files which come with every PostScript font into vpls. Unfortunately, `afm2tfm` cannot produce fonts with more than one raw font (for example the 'TEX text' encoded Times-Roman uses Symbol for the upper case Greek) and had problems with math fonts.

4 A solution: the 'fontinst' package

The `fontinst` package is designed to read AFMs and produce vpls. It:

- Is written in TEX, for maximum portabilty (at the cost of speed).
- Supports the TEX text, TEX math, and extended TEX text encoding.
- Allows fonts to be generated in an arbitrary encoding, with arbitrary 'fake' characters—for example the 'ff' character can be faked if necessary by putting an 'f' next to a 'f'.
- Allows caps and small caps fonts with letter spacing and kerning.
- Allows kerning to be shared between characters; for example 'Å' can be kerned as if it were an 'A'. This is useful, since many PostScript fonts only include kerning information for characters without accents.
- Allows the generation of math fonts.
- Allows more than one PostScript font to contribute to a TEX font; for example the 'ff' ligature can be taken from the Expert encoding, if you have it.
- Automatically generates an `fd` file for use with LATEX 2_ε.

The `fontinst` package is available as freeware from the CTAN archives, along with a selection of VFs which have been generated with `fontinst`.

Version 0.19 of `fontinst` is described in the proceedings of the Aston TUG AGM (*TUGboat* 14(3)). This description is now largely out of date.

The VFs generated by `fontinst` will be the standard VFs for use with Sebastian Rahtz's `psnfss` package for LATEX 2_ε.

5 Using the 'fontinst' package

The `fontinst` package comes with full documentation in the file `fontinst.tex`. The simplest way to start to use `fontinst` is to edit the file `fonttime.tex`, shown in Table 1 This tells TEX to create the Adobe Times Roman fonts in the 'TEX extended text' (T1) encoding, using the files:

- `ptmr0.afm`, `ptmri0.afm`, `ptmb0.afm` and `ptmbi0.afm`, the Times-Roman afm files.
- `latin.mtx`, the *TEX metric* file containing the default Latin characters.
- `T1.etx` and `T1c.etx`, the *TEX encoding* files containing the 'TEX extended text' and 'TEX extended tex caps & small caps' encodings.

This produces a number of PL and vpl fonts, which can be converted into TEX fonts using `pltotf` and `vptovf`.

For example by replacing every occurrence of `ptm` by `ppl` you can install the Adobe Palatino fonts.

If you generate any fonts with `fontinst` which you think other people might want to use, please send them to me, and if I like them, I'll include them in the `fontinst` contributors directory.

```
\input fontinst.sty
\needsfontinstversion{1.303}

\installfonts
   \installfamily{T1}{ptm}{}
   \installfont{ptmrq}{ptmr0,latin}{T1}{T1}{ptm}{m}{n}{}
   \installfont{ptmrcq}{ptmr0,latin}{T1c}{T1}{ptm}{m}{sc}{}
   \installfont{ptmriq}{ptmri0,latin}{T1}{T1}{ptm}{m}{it}{}
   \installfont{ptmbq}{ptmb0,latin}{T1}{T1}{ptm}{bx}{n}{}
   \installfont{ptmbcq}{ptmb0,latin}{T1c}{T1}{ptm}{bx}{sc}{}
   \installfont{ptmbiq}{ptmbi0,latin}{T1}{T1}{ptm}{bx}{it}{}
\endinstallfonts
\bye
```

Table 1: The file `fonttime.tex`