Virtual fonts — a tutorial

Thomas A. Schmitz

Lots of information on TeX's virtual fonts can be found on the web and in books (e.g., Knuth's very own "Virtual Fonts: More Fun for Grand Wizards", `http://www.ctan.org/tex-archive/info/virtual-fonts.knuth`). However, there doesn't seem to be a step-by-step tutorial for non-wizards like myself. I have experimented with virtual fonts recently. It took me a while to understand the basics, so I thought that other people might find it useful to hear about this and avoid some common mistakes.

1 Basic facts about virtual fonts

Let's start by discussing two immensely useful things that virtual fonts can do.

First, they can remap characters within the same font. If you have a font `foo` with files `foo.pfb` and `foo.tfm`, you can make a virtual font `foobar` that will be identical to `foo` but print an "A" whenever you have a "B" in your TeX file.

This may sound absurd at first, but there are cases where it is useful. For instance, some fonts offer alternative forms for letters. With the help of a virtual font, you can remap the letters and thus switch to these alternative forms without changing your TeX source.

The second use for virtual fonts is much more common: Given a font `foo`, you can create a virtual font `foobar` that will include some characters from a second font, say `bar`. This is often used to include old-style numerals or additional ligatures that are not provided by the normal font.

2 Copying the font files

So, let's begin. We assume that we will be using two PostScript fonts, `foo` and `bar`. Usually, for each of these fonts, we will have two files `foo.tfm` (resp. `bar.tfm`) and `foo.pfb` (resp. `bar.pfb`) and nothing else, so we need to create a virtual font (`.vf`) file from scratch. I didn't find it mentioned anywhere that this is not only possible, but even fairly easy.

We'll perform these operations on the command line in a working directory, such as `/tmp`. So the first step is to copy `foo.tfm` to this directory:

```
cd /tmp
cp /PATH/TO/foo.tfm .
```

(Watch the trailing period, it's necessary!)

3 Create a human-oriented property list

The file `foo.tfm` is a binary file, in a format that TeX can read. If we want to edit it, we will have to convert it to a so-called "property list" file (typically given the extension `.pl`), which is a plain text file that can be read by humans. We will be using tools that come with any complete TeX installation. From the command line:

```
tftopl foo foo
```

(Yes, that's right: we have to type `foo` twice!)

4 Open the property list

We now have a new file `foo.pl` which contains all the information about the font that TeX needs. Open it in your favorite text editor. If you're editing in a non-Unix environment, such as Windows or Mac OS X, make sure that your editor is set to use Unix line endings, unless you know for certain that your TeX utilities don't mind. The first few lines will read like this:

```
(FAMILY TEX-FOO)
(FACE F MRR)
(CODINGSCHEME FONTSPECIFIC + TEX TEXT)
(DESIGNSIZE R 10.0)
...
(LIGTABLE ...
```

If there is a line `(CHECKSUM O ...)`, delete it; it will be regenerated later.

5 Editing the property list

In order to generate a virtual font, we need to modify this file. First, we have to tell TeX which fonts our new virtual font will be referring to. Let's say they are `foo.tfm` and `bar.tfm` — needless to say, both have to be installed and functional in your TeX installation.

As a first step, we will create a virtual font that will remap some characters within foo. So just before the line starting with `(LIGTABLE`, add this:

```
(MAPFONT D 0
   (FONTNAME foo)
   (FONTDSIZE R 10.0)
   )
```

The `FONTDSIZE` of `foo` is found from the
`(DESIGNSIZE R 10.0)`
line above; all we have to do is copy this information.

6 Remapping a character

Now let's scroll down in this file. The `LIGTABLE` (containing information about ligatures and kerning) will end with two lines

```
(STOP)
)
```

After this, the section with information about all the defined characters in the font will follow, probably starting something like this:

```
(CHARACTER O 0
   (CHARWD R 0.674)
   (CHARHT R 0.726)
   )
```

TeX itself only cares about the dimensions of characters, as stored in the `.tfm` file, when doing the typesetting; it essentially leaves room for an empty box with these dimensions. The actual characters (the visible "glyphs") are put into these boxes only when the final PostScript or PDF output is made.

```
(CHARACTER C A
   (CHARWD R 0.747)
   (CHARHT R 0.747)
   )
(CHARACTER C B
   (CHARWD R 0.739)
   (CHARHT R 0.726)
   )
```

TeX will be using the box described as here, so we want the box for "B" to have the dimensions of the box for "A". Hence, the first thing to do is copy the dimensions of "A" into "B". Then the section should look like this:

```
(CHARACTER C A
   (CHARWD R 0.747)
   (CHARHT R 0.747)
   )
(CHARACTER C B
   (CHARWD R 0.747)
   (CHARHT R 0.747)
   )
```

Next (and this is the magic of virtual fonts) we tell TeX that it should remap "B" to "A". Just before the closing parenthesis of `CHARACTER B`, we insert a new section, so that "B" will look like this:

```
(CHARACTER C B
   (CHARWD R 0.747)
   (CHARHT R 0.747)
   (MAP
      (SETCHAR C A)
      )
   )
```

One of the things that can be a bit confusing about these property lists is that (apart from the numbers and the 26 letters of the English alphabet) characters are referred to by "octal numbers". If you want to know what character corresponds to what octal number, you can have a look at the tables created by testing the font (section 9).

7 Saving the file

That's it! We have modified the font description; now we need to generate the binary files for TeX to use. The next step is extremely important: **save the file to a different name**.

In our case, let's say we call the new virtual font `foobar`. The base name doesn't much matter, but the extension should be `.vpl`; so let's save to `foobar.vpl`.

8 Generating the binary files

Back to the command line. We now run a program that will convert `foobar.vpl` into two new files, `foobar.tfm` and `foobar.vf`:

```
vptovf foobar.vpl
```

This will not only do the conversion, it will also check whether the `.vpl` file is in good order. It is very picky about the right indentation level and parentheses; if there is a problem it will give the exact line number. So if you get errors, just go back and edit `foobar.vpl` again.

`vptovf` may also tell you that it had to "round some units"; that's OK.

9 Installing the new font

So now we should have `foobar.vf` and `foobar.tfm`. The next step is to copy both files into the right place. I would suggest you create your own `texmf`-branch in your home directory, for instance, under `~/Library/texmf`, or `~/` (depending on your local setup as defined in `texmf.cnf`). For the sake of our example, we'll use the former:

```
cp foobar.tfm ~/Library/texmf/fonts/tfm/
cp foobar.vf ~/Library/texmf/fonts/vf/
```

(You will have to create these directories if they don't exist yet.)

Since we're only using characters from within a single font (`foo`), we don't need to fiddle with any "map files". When the final output is made, only font `foo` will be needed, which was already functional.

Before embarking on a long journey with this new virtual font (say your 1200-page thesis that is due in two weeks), let's test it on its own:

```
cd ~
pdfetex testfont
```

`pdfetex` will respond something like this:

```
This is pdfeTeX, Version 3.1415...
...
(/usr/local/.../plain/base/testfont.tex
Name of the font to test =
```

We now type the name of our font:

```
foobar
```

and `pdfetex` will respond:

```
Now type a test command (\help for help):)
*
```

We give the command:

```
\table
```

pdfetex will come back with another asterisk, and now we're done:

```
    \end
```

If all goes well, a file testfont.pdf will be created with a table showing that font foobar does not have a letter "B", but twice the letter "A" — which is just what we wanted.

If your TeX distribution includes the ConTeXt format (see http://tug.org/pracjourn/2005-1/peter/), you can also create a very nice colorized table. Make a file test.tex like this:

```
\starttext
\showfont[foobar]
\stoptext
```

Then, run this file through ConTeXt:

```
    texexec --pdf --nonstopmode test.tex
```

You'll get a table with all the glyphs; every cell will indicate the decimal, hexadecimal, and octal value of the glyph (very handy for editing property lists).

10 Adding a second font

Now let's work on including further characters, like old-style numerals or ligatures from font bar. We go back to our working directory and delete the old file foobar.vpl. Don't worry, we'll create it again:

```
    rm foobar.vpl
    vftovp foobar foobar foobar
```

We do this because vftovp will automagically include one important piece of information; every character description will now look like this:

```
(CHARACTER C A
   (CHARWD R 0.747)
   (CHARHT R 0.747)
   (MAP
      (SETCHAR C A)
      )
   )
```

That's already not bad, but to mix glyphs from two fonts, we have to say which font to use in every instance. So in an editor, we perform a "find and replace" that will find every instance of (MAP and replace it with

```
(MAP
   (SELECTFONT D 0)
```

In Emacs, it is possible to include the line break in the replace pattern. (I don't know how other editors can handle it.)

Now add the information about the second font, just as described in section 5:

```
(MAPFONT D 1
  (FONTNAME bar)
  (FONTDSIZE R 10.0)
  )
```

To get the DSIZE of bar, we can again just convert bar.tfm into bar.pl, open this file and look into the first lines. Or, if you want to be fancy:

```
    tftopl ‘kpsewhich bar.tfm‘ | grep DESIGNSIZE
```

(watch the "backticks", those are single opening quotes!).

11 Including glyphs from a second font

Now look for the section containing the numerals. It should start like this:

```
(CHARACTER C 0
   (CHARWD R 0.514)
   (CHARHT R 0.628)
   (CHARDP R 0.1)
   (MAP
      (SELECTFONT D 0)
      (SETCHAR C 0)
      )
   )
```

Again, the first thing to do is copy the dimensions of CHARACTER C 0 from bar.pl to foo.pl. Then the new step, to use bar instead of foo: replace (SELECTFONT D 0) with (SELECTFONT D 1). Repeat for all the other numerals.

We then follow the same procedure as before: generate the tfm/vf pair (section 8) and copy these files to the right directories (section 9). Now TeX and friends will look at the new tfm/vf, and take the numerals from font bar, and everything else from font foo. Again, test the results!

12 Post-install

After you have edited your virtual font, you can discard the .vpl file. If you ever want to edit your font again, you can recreate it by copying both the .vf and the .tfm into the same directory and running

```
    vftovp foo foo foo
```

Have great fun and feel like a great wizard!

13 Further reading

http://www.cl.cam.ac.uk/users/rf/pstex/index.htm
http://homepage.mac.com/bkerstetter/tex/fonttutorial-current.html
http://zoonek.free.fr/LaTeX/Fontes/fontes.html (in French)

◇ Thomas A. Schmitz
Institut fuer Klassische und
Romanische Philologie
Universitaet Bonn, 53113 Bonn
Germany
thomas.schmitz@uni-bonn.de