

PostScript Fonts in L^AT_EX 2_ε

Alan Jeffrey

School of Cognitive and Computing Sciences

University of Sussex

Falmer

Brighton

BN1 9QH

UK

alanje@cogs.susx.ac.uk

Abstract

This paper describes the L^AT_EX 2_ε PostScript fonts package `psnfss`, and the Adobe Times math fonts package `mathptm`. This paper describes some of the design decisions made in `psnfss`, and gives an overview of how other fonts can be used in a similar fashion.

Introduction

This paper describes Sebastian Rahtz's `psnfss` package for using PostScript fonts in L^AT_EX. The `psnfss` software has been the standard way of using PostScript fonts in L^AT_EX for a number of years, and has recently been updated for L^AT_EX 2_ε and the fonts generated by the author's (1994) `fontinst` package.

The `psnfss` package aims to make using PostScript fonts as simple as possible. Once `psnfss` has been installed, users can select a L^AT_EX 2_ε package such as `times` and their document will be set in Adobe Times. The `psnfss` package comes with a standard set of T_EX font metric (`tfm`) files, so `psnfss` documents should be portable between different T_EX implementations.

One of the new features of `psnfss` is the `mathptm` package, which allows Adobe Times to be used as a math font as well as a text font. This only uses standard fonts (Adobe Times, Adobe Symbol and Computer Modern) so `mathptm` documents are portable, and the resulting PostScript files can be distributed without worrying about proprietary fonts.

Together with David Carlisle and Sebastian Rahtz's (1994) `graphics` and `color` packages, `psnfss` will help to free L^AT_EX from its popular image as only setting academic texts in Computer Modern with `picture` mode `graphics`.

Using `psnfss`

Once `psnfss` has been installed, it is very simple for users to use. They just select an appropriate package, for example:

```
\documentclass{article}
\usepackage{times}
```

Some document classes will be designed for use with PostScript fonts, and will automatically select PostScript fonts without the user selecting a package. For example, a publishing house producing *The Journal of Dull Results* may have their own `jdullres` document class, with options for pre-prints or final copy. An author would type:

```
\documentclass[preprint]{jdullres}
```

and would get a pre-print document set in Adobe Times, whereas the production staff would type:

```
\documentclass[crc]{jdullres}
```

to get camera-ready copy set in Autologic Times.

Documents written using `psnfss` can be printed with an appropriate dvi driver such as `dvips`, or `OzTEX`. Some previewers, such as `xdvi`, cannot preview PostScript fonts without turning the fonts into bitmaps (using up valuable disk space). But the PostScript can be previewed, using `ghostview` or `pageview`.

PostScript math fonts

One of the common complaints about using L^AT_EX with PostScript fonts is that the mathematics is still set in Computer Modern, for example as in Figure 2. This is unfortunate, since Computer Modern is a much lighter, wider and more cursive font than suits Adobe Times.

Until recently, the only thing that could be done about this was to use the MathTime fonts, available from Y&Y. These are fine fonts, and can produce excellent math setting. Unfortunately, they are proprietary software, and so cannot be distributed as freely as Computer Modern.

Suppose $f \in \mathcal{S}_n$ and $g(x) = (-1)^{|x|} x^\alpha f(x)$. Then $g \in \mathcal{S}_n$; now (c) implies that $\hat{g} = D_\alpha \hat{f}$ and $P \cdot D_\alpha \hat{f} = P \cdot \hat{g} = (P(D)g)$, which is a bounded function, since $P(D)g \in L^1(\mathbb{R}^n)$. This proves that $\hat{f} \in \mathcal{S}_n$. If $f_i \rightarrow f$ in \mathcal{S}_n , then $f_i \rightarrow f$ in $L^1(\mathbb{R}^n)$. Therefore $\hat{f}_i(t) \rightarrow \hat{f}(t)$ for all $t \in \mathbb{R}^n$. That $f \rightarrow \hat{f}$ is a *continuous* mapping of \mathcal{S}_n into \mathcal{S}_n follows now from the closed graph theorem. *Functional Analysis*, W. Rudin, McGraw–Hill, 1973.

Figure 1: Computer Modern text with matching math

Suppose $f \in \mathcal{S}_n$ and $g(x) = (-1)^{|x|} x^\alpha f(x)$. Then $g \in \mathcal{S}_n$; now (c) implies that $\hat{g} = D_\alpha \hat{f}$ and $P \cdot D_\alpha \hat{f} = P \cdot \hat{g} = (P(D)g)$, which is a bounded function, since $P(D)g \in L^1(\mathbb{R}^n)$. This proves that $\hat{f} \in \mathcal{S}_n$. If $f_i \rightarrow f$ in \mathcal{S}_n , then $f_i \rightarrow f$ in $L^1(\mathbb{R}^n)$. Therefore $\hat{f}_i(t) \rightarrow \hat{f}(t)$ for all $t \in \mathbb{R}^n$. That $f \rightarrow \hat{f}$ is a *continuous* mapping of \mathcal{S}_n into \mathcal{S}_n follows now from the closed graph theorem. *Functional Analysis*, W. Rudin, McGraw–Hill, 1973.

Figure 2: Adobe Times text with Computer Modern math

Suppose $f \in \mathcal{S}_n$ and $g(x) = (-1)^{|x|} x^\alpha f(x)$. Then $g \in \mathcal{S}_n$; now (c) implies that $\hat{g} = D_\alpha \hat{f}$ and $P \cdot D_\alpha \hat{f} = P \cdot \hat{g} = (P(D)g)$, which is a bounded function, since $P(D)g \in L^1(\mathbb{R}^n)$. This proves that $\hat{f} \in \mathcal{S}_n$. If $f_i \rightarrow f$ in \mathcal{S}_n , then $f_i \rightarrow f$ in $L^1(\mathbb{R}^n)$. Therefore $\hat{f}_i(t) \rightarrow \hat{f}(t)$ for all $t \in \mathbb{R}^n$. That $f \rightarrow \hat{f}$ is a *continuous* mapping of \mathcal{S}_n into \mathcal{S}_n follows now from the closed graph theorem. *Functional Analysis*, W. Rudin, McGraw–Hill, 1973.

Figure 3: Adobe Times text with matching math

A less beautiful, but cheaper, solution is to use the `mathptm` package. This provides drop-in replacements for Computer Modern using virtual fonts built from Adobe Times, Symbol, Zapf Chancery, and Computer Modern. The results can be seen in Figure 3.

The `mathptm` fonts are distributed free, and the resulting PostScript documents can be made available for anonymous `ftp` without having to worry about unscrupulous readers stealing the fonts from the PostScript documents.

Roadmap

The rest of this paper describes some technical details about the implementation of `psnfss`, for the \TeX nically minded.

The unpacked `psnfss` package comes as a number of files:

- Files ending with `sty` are \LaTeX 2 ϵ packages. For example, `times.sty` contains the `times` package.
- Files ending with `fd` are \LaTeX 2 ϵ font definition files. For example, `T1ptm.fd` contains the font definitions for Adobe Times. This tells \LaTeX 2 ϵ that, for example, Adobe Times bold italic is called `ptmbiq`.
- Files ending with `tfm` are \TeX font metric files. For example, `ptmbiq.tfm` contains the font in-

formation which \TeX needs for Adobe Times bold italic.

- Files ending with `vf` are virtual fonts. For example, `ptmbiq.vf` contains the font information which some printers and previewers need for Adobe Times bold italic. This tells the printer that, for example, the character ‘ \acute{C} ’ is made from an ‘acute’ and an ‘C’.

The `sty` and `fd` files are used by \LaTeX , the `tfm` files are used by \TeX , and the `vf` files are used by printers and previewers.

Document portability

The `psnfss` package is intended to make documents as portable as possible. To achieve this, the `sty`, `fd` and `tfm` files should be the same at all sites. This means that documents using the `times` package will print identically on different sites.

Although the `psnfss` package distributes all of the files that are used by \LaTeX and \TeX , it does *not* include the files which are used by particular printer drivers, since these change from site to site. The `psnfss` package makes no requirements on the printer driver, except that it can print with PostScript fonts.

Virtual fonts

Although `psnfss` comes with virtual fonts, these are an optional part of the package. Some printer drivers (such as Y&Y's `dvipsone`) use PostScript font re-encoding rather than virtual fonts. The advantages of virtual fonts include:

- More than one font can be combined together. For example, many PostScript fonts contain ‘ff’ ligatures in the Expert fonts, but T_EX requires ligatures to be in the same font.
- Composite letters such as ‘Š’ and ‘Ć’ can be produced. T_EX requires such letters to be in a font for the hyphenation algorithm, but most PostScript fonts do not contain them.
- Most printer drivers and previewers can use virtual fonts, so they are portable between systems.

The advantages of PostScript font re-encoding are:

- PostScript font re-encoding is faster, for example Textures previewing with virtual fonts can be twice as slow as with raw fonts.
- PostScript font re-encoding is a standard technology supported by many other applications.
- In order to access characters like ‘Á’, the printer driver has to use PostScript font re-encoding anyway, so virtual fonts need two levels of re-encoding rather than one.

Since there is a trade-off between virtual fonts and PostScript font re-encoding, the `psnfss` package makes no assumptions about using virtual fonts. The `vf` files are available for those who want to use them, but not all sites will want to use them.

Existing PostScript fonts

In the past, there have been problems with installing `psnfss` on systems which have already got PostScript fonts generated using Tom Rokicki's `afm2tfm` converter.

The main difference between the `fontinst` fonts and the `afm2tfm` fonts is that the `fontinst` fonts are designed to be drop-in replacements for Computer Modern, and can be used with no new macros. The `afm2tfm` fonts contain some new characters (such as ‘©’) and some old characters (such as ‘”) in different slots, so need special macros. See Tables 2 and 3.

One of the most important features of L^AT_EX 2_ε is that different fonts can be used without new macros, which is one of the reasons for using the `fontinst` fonts rather than the `afm2tfm` fonts.

In the past, there were problems using `psnfss` on systems which had already got the `afm2tfm` fonts,

because they used the same font names. This has now been changed, and `fontinst` uses the letters `7t` to indicate a ‘7-bit T_EX text’ font, and `0` to indicate a ‘Adobe Standard’ font. For example the filenames for Adobe Times are:

<i>encoding</i>	<i>fontinst name</i>	<i>afm2tfm name</i>
Adobe Standard	<code>ptmr0</code>	<code>rptmr</code>
7-bit T _E X text	<code>ptmr7t</code>	<code>ptmr</code>
8-bit T _E X text	<code>ptmrq</code>	<code>none</code>

There should now be no clashes between the fonts generated by `afm2tfm` and those generated by `fontinst`.

Font naming

The `psnfss` fonts are named using Karl Berry's naming scheme. This tries to fit as many font names as possible into the eight letters provided by some operating systems. For most fonts here, the names are given by:

- One letter for the font foundry, e.g. `p` for Adobe.
- Two letters for the font family, e.g. `tm` for Times Roman, `hv` for Helvetica or `cr` for Courier.
- One letter for the weight, e.g. `r` for regular or `b` for bold.
- An optional letter for the shape, e.g. `i` for italic, `o` for oblique, or `c` for small caps. No letter means ‘upright’.
- One or two letters for the encoding, e.g. `q` for ‘Cork’ encoding, `0` for ‘Adobe Standard’ encoding or `7t` for Knuth's ‘T_EX text’ encoding.

For example:

- `ptmr7t` is Adobe, Times Roman, regular weight, upright shape, T_EX text encoding.
- `phvbcq` is Adobe, Helvetica, bold weight, small caps shape, Cork encoding.
- `pcrr0` is Adobe, Courier, regular weight, oblique shape, Adobe Standard encoding. This is the font Adobe call ‘Courier-Oblique’.

The naming scheme is described in more detail by Berry (1994). It is far from ideal, but does allow most fonts to be named in a consistent fashion. Most systems require a translation from the T_EX font name to whatever the local font name is. For example, `dvips` can be told that `pcrr0` is Adobe Courier with a line in the `psfonts.map` file:

```
pcrr0 Courier
```

OzT_EX requires a line in the Default configuration file:

```
= pcrr0 Courier Courier Mac.enc
```

Textures can add new font names using the EdMetrics application.

33MHz 68030) but this is acceptable since it is not run often.

Version 0.19 was described by the author (1993), but the user interface has changed considerably since then.

To install a new font using `fontinst`, you need the Adobe Font Metrics (`afm`) files for the fonts. These should be named with Karl Berry's naming scheme, for example `ptmr0.afm` for Adobe Times Roman. They should be in a directory which can be read by `TeX`.

Installing Latin fonts is quite simple. For example, to install the Adobe Times fonts, you run `TeX` on `fontinst.sty` and say:

```
\latinfamily{ptm}{}
```

If you have bought the Expert fonts, you can install Adobe Times Expert and Adobe Times Old Style by saying:

```
\latinfamily{ptmx}{}
\latinfamily{ptm9}{}
```

Once `TeX` has finished, it produces:

- `pl` files, which are text representations of `tfm` files. They can be converted to `tfm` files with `pltotf` (part of EdMetrics in Textures and OzTools in OzTeX).
- `vp1` files, which are text representations of `vf` files. They can be converted to `vf` files with `vp1tovf` (part of EdMetrics in Textures and OzTools in OzTeX).
- `fd` files, which are used by $\LaTeX 2\epsilon$.

The Adobe Times fonts can then be used in \LaTeX by saying:

```
\renewcommand{\rmdefault}{ptm}
\rmfamily
```

If you want to write your own package similar to `times`, you just create a `sty` file containing these lines. For example, an Adobe Times Old Style package would contain:

```
\ProvidesPackage{timesold}
 [1995/04/01 Times old style digits]
\renewcommand{\rmdefault}{ptm9}
\rmfamily
```

It is also possible to create customized fonts (such as the `mathptm` math fonts) using `fontinst`, but this is somewhat trickier, and is described in the `fontinst` documentation.

If you use `fontinst` to install some fonts, and you would like to distribute the results, please mail them to me, and I'll include them in the `fontinst` distribution.

Ongoing work

There are a number of areas of ongoing work with fonts and `TeX`.

There is a TUG working group on `TeX` directory structures, which will recommend how fonts should be installed on any `TeX` system. This will make it easier to distribute `TeX` software, because there will be a standard directory structure to refer to. The CTAN `fonts/metrics` directory will reflect the TDS structure.

The fonts produced by `fontinst` do not include 'eth' or 'thorn' because these characters are not available in Adobe Standard encoding. We are working on developing a suitable replacement for Adobe Standard encoding which will allow access to 'eth', 'thorn' and the other missing characters.

There is a TUG/ \LaTeX working group on math font encodings, which will develop symbol encodings which will be supported by `fontinst` (this working group has been rather slowed down by the production of $\LaTeX 2\epsilon$).

Bibliography

- Berry, Karl. *Filenames for fonts*, available as `fontname.texi` from CTAN, 1994.
- Carlisle, David and Rahtz, Sebastian. *The color and graphics $\LaTeX 2\epsilon$ packages*, available from CTAN, 1994.
- Goossens, Mittelbach and Samarin, *The \LaTeX companion*, Addison-Wesley, 1993.
- Jeffrey, Alan. A PostScript font installation package written in `TeX`, in *Proc. 1993 TUG AGM*.
- Jeffrey, Alan. *The fontinst package*, available from CTAN, 1994.
- Rahtz, Sebastian. *Notes on setup of PostScript fonts for $\LaTeX 2\epsilon$* , part of the `psnfss` package, available from CTAN, 1994.