Disclaimer

These notes were originally prepared at the request of Brian Reid, for informal distribution. They are based on the author's review of available literature on the subject of typeface protection, and on personal experience in registering types for trademark, copyright, and patent. However, they are not legal advice. If one is contemplating protecting or plagiarizing a typeface, and seeks legal opinion, it is advisable to consult an attorney. The term "plagiarize" (and words derived from it) is used here in its dictionary sense of "to take and use as one's own the ideas of another" and does not mean that the practice of typeface plagiarism is illegal, as that is determined by the laws of a particular country.

The author is a professor of digital typography as well as a professional designer of original digital typefaces for electronic printers and computer workstations. He therefore has an obvious bias toward the inculcation of ethical standards and the legal protection of artistic property. Other commentators might have a different perspective.

Building Computer Modern fonts

John Sauter

When **METAFONT** version 1.0 was released I eagerly obtained a copy, because I wanted to use the Computer Modern fonts on my DEC LN03. By experiment, and with some assistance from Professor Knuth, I determined the **METAFONT** device-dependent parameters for the LN03. The non-obvious parameters are: *blacker* 0.65, *fillin* −0.01, *o_correction* 0.5.

I then found that the process of building all of the font files on a VMS system, while straightforward, is not trivial. I have written some VMS command procedures which build the font files, and I would be happy to share them with the TeX community. They require about $2\frac{1}{2}$ days of CPU time on a VAX-11/785, so I would also be willing to share the results of this process.

Of course, I was not satisfied with just the 75 standard fonts in the standard 7 magnifications. I also wanted Computer Modern Symbols in 12-point, since I use 12-point a lot due to the low resolution of the LN03. In addition, I like to use Computer Modern Sans Serif for acronyms, but CMSS12 looks a little too large, so I wanted Computer Modern Sans Serif 11-point. I could have gotten these using magnified fonts, of course, but that doesn't seem right: I wanted fonts that were designed for the size in which I was using them. I expected to want more fonts than these someday, like a 20-point monospaced font, so I wanted a way to build non-standard Computer Modern fonts in a more-or-less automatic way.

My solution to this problem was to create alternative parameter files to produce the Computer Modern fonts. By ignoring point size I counted 31 Computer Modern font families: CMB, CMBSY, CMBX, CMBXSL, CMBXTI, CMCSC, CMDUNH, CMEX, CMFF, CMFI, CMFIB, CMINCH, CMITT, CMMI, CMMIB, CMR, CMSL, CMSLTT, CMSS, CMSSBX, CMSSDC, CMSSI, CMSSQ, CMSSQI, CMSY, CMTCSC, CMTEX, CMTI, CMTT, CMU and CMVTT. I created the corresponding 31 parameter files, each of which takes a point size as input. If you give the alternative parameter file a point size which corresponds to one of the standard Computer Modern fonts, it produces exactly the same results as the standard parameter file. If you give it a point size which does not correspond to a standard font, it interpolates or extrapolates each of the font parameters to produce what seems to me to be a reasonable value, based on all the standard values for that parameter in that font family. In font families in which only one point size is given, such as CMFF10, I couldn't do more than linear extrapolation. In most of the families, though, I was able to write an algorithm for each parameter which gave reasonable results for all point sizes between 5 and 100. Sometimes the formulae are quite complex, in order to exactly match the standard values at the standard point sizes.

Wherever possible I tried to use common files for similar calculations. For example, the upper case part of CMCSC10 is almost identical to CMR10, so I used a single file, `COMPUTE_CMR`, to compute the common parts of both. Thus, even though CMCSC is given only in 10-point, I can use the algorithms of CMR to give better values than I could have gotten through linear extrapolation. Similarly, CMSY is very similar to CMMI, so I can use the CMMI calculations to give a better CMSY12.

To guard against typographical error, I have used these alternative parameter files to create all of the standard point sizes and magnifications for the LN03, and compared them with the files produced by the standard parameter files. The `.TFM` files had to match exactly, or the fonts could not be called

Computer Modern. I was willing to accept some variation in the pixel files, but as it turned out after I removed all the errors from the parameter files there were no differences in the pixel files either.

I had some trouble matching the `.TFM` files because of roundoff errors, particularly in fonts like CMCSC, where some parameters are computed by modifying others. I solved this by using units of $\frac{1}{360}$ of a point rather than the standard $\frac{1}{36}$ of a point when I could do so without overflow, and by adding "fudge factors" to correct the remaining small differences.

I would like to distribute these files through the normal mechanism from Stanford, since I have received very good service from Maria Code, but I don't know how. I intended to research this problem before writing this article, but the deadline for this issue snuck up on me so I won't have time. Therefore, if anybody wants VMS Backup copies of all that I have described above—command files, alternative parameter files, and the resulting `.TFM` and pixel files for the DEC LN03—just write me and I'll send you a magnetic tape by return mail. If you can't read 6250 BPI tapes be sure to let me know, since that is my default density: it lets me use a smaller tape.

Editor's note: Arrangements are being made to include at least some of the files described above on the VAX/VMS distributions from Stanford and from Kellerman & Smith. The files have also been offered for inclusion on the VAX/Unix tape; this may take a bit more time to effect, since the algorithms must be translated to a Unix shell script, and VMS dependencies removed. Anyone wishing to volunteer to undertake a translation to Unix should communicate with Pierre MacKay or Barbara Beeton.

Output Devices

TEX and Macintosh – New Directions in Preview

Rick Jansen
Academic Computing Services Amsterdam
(SARA)

At SARA [1], the Academic Computing Services Amsterdam, the Macintosh application TEX Preview has been developed, a tool that you can use to view DVI files with a Macintosh microcomputer directly after running TEX on the host computer.

Why Preview

TEX is not an easy to use word processor, in fact TEX commands are quite error prone, so "debugging" a TEX document can be a rather tedious process. This is especially true if the DVI files are printed with an off-line typesetting machine. At SARA it may take as much as three days to get your DVI file printed. Clearly, it is very frustrating for the user to get (expensive) output with some typing errors or an entirely italic paragraph because there was a "}" missing.

Previewing of documents before actually typesetting them can be very useful to correct errors and to try things out. It saves you time, money and a lot of frustration if you can get a view of the results directly after running TEX.

Why Macintosh

To represent pages formatted by TEX with different fonts and styles you need a graphics device with very powerful graphics. As speed and cost are the main considerations, a graphics terminal and some kind of driver program on the host computer do not suffice. The previewing device must have these capabilities of its own. Therefore the Apple Macintosh was chosen for a TEX Preview facility. Macintosh is a microcomputer with very powerful graphics, different fonts, fontsizes, styles, etc. Its excellent datacommunication facilities enable you to easily connect Macintosh to the host computer running TEX. You can use Macintosh as a terminal for editing the TEX input and for transferring the DVI file from the host computer to a Macintosh diskette for previewing.