



It would be easy enough for me to natter on about the wonders of the new and improved version of METAFONT: The code for the font you are now reading was only a gleam in my eye in mid-December of 1985; The code for the masthead on this column took less than eight hours for me to write; and so on and so on. I will not belabor the point that METAFONT's way of precisely describing how a character is drawn (rather than simply drawing it) allows the designer literally to go from alpha to omega:

Α Α Δ Δ Ω Ω Ω

At present I wish to focus on the *product*, not the *process*, of METAFONT. Most users of T<sub>E</sub>X believe (rightly or wrongly) that the *process* of METAFONT is something wholly beyond their requirements; its *product*, on the contrary, is fundamental to the effective use of T<sub>E</sub>X.

Let me digress briefly to enumerate some of the details that must be included in a good font. The first consideration that comes to mind is the wealth of detail in the images of the characters that make up the font. Curves must be smooth and pleasing to the eye. The ratio of descender to mean-height must 'look right' for the character's representation at a given point size, and must be harmonious from one point size to another; in many of the classic typefaces, this visual harmony is much too complex to be adequately captured by simple scaling from a single template.

dp dp  
dp dp

On the left, we have letters from a 48 point Roman font and the 12 point version produced by simply scaling down from it; the ratios of ascender and descender to point size are the same for both sizes. On the right, we have the same 48 point, but a 12 point version that has, in essence, been completely redrawn by METAFONT to preserve the slight but significant change in ascenders and descenders which typically occurs in traditional typefaces to enhance the legibility of text

faces. *Completely redrawn*, but done so from the same code; thus, METAFONT minimizes effort while maximizing effectiveness.

Widths and heights of component strokes and of the images themselves show subtle changes of their own. METAFONT, with its potential for attention to such necessary niceties, handles all these imaging considerations with far more finesse than is possible with any other digital design system.

All of these imaging adjustments are well and good; but as the reader knows, digital image information is not used by T<sub>E</sub>X at all. The details that T<sub>E</sub>X requires are all contained in separate files, the .t<sub>m</sub> files. For example, to set text, we need to have an idea of the amount of space between words; ideally, such space will grow or shrink within acceptable limits depending on the exigencies of the text being set; an error here can make a font the ultimate inconvenience rather than the ultimate in convenience.

It is imperative that the information contained herein be as precise as possible in order to make a good font, a font capable of rising to the highest level of digital typesetting quality, that is, a font suitable for T<sub>E</sub>X. If the amount of space at the side of each letter's image is ill-advised, typesetting quality deteriorates and legibility is sacrificed (Notice how dreadful the seven words preceding this parenthetical remark look.)

Moreover, the ability to let these spaces shrink or grow ever so slightly according to the letter's environment, called  *Kerning*, as in

Toffee<sub>not</sub> Toffee

or to substitute a slightly different image for a character pair or triple, called *ligatures*, as in

Toffee<sub>for</sub> Toffee

is a hallmark of the best typesetting. METAFONT has the ability to handle both kerns and ligatures and thus can produce fonts that rise above the level of the merely adequate.

In some cases, such information can be derived 'after the fact' (from existing images) and appended to a digital font; but it is not difficult to imagine that such a shotgun marriage of image information and .tfm information will tend not to result in a happy and harmonious union of the two. Quite simply, the simultaneous creation of image and .tfm information as done with METAFONT produces the best results.

More critically, there are some other subtler bits of information in font metric files that are much more nearly impossible to imagine creating with any tool other than METAFONT. Those are the various tidbits in the math/science and symbols fonts that are crucial to the fine setting of equations and formulæ that is one of T<sub>E</sub>X's strengths. These fonts must have a wealth of information that controls positioning and even the composition of certain characters (built up curly brackets, square brackets, integrals, radicals).

The modern typographer must now understand that his art has become an interdisciplinary pursuit and involves mathematics and programming skills as well as the traditional design concerns. While most current typographers will fail to adjust to this radically different method of type design, there will be many newcomers who will use METAFONT to contribute the beautiful digital typefaces that T<sub>E</sub>X needs for unprecedentedly superb typesetting.

Powell, Ohio  
24 May 1986

*Every character in this column was created using METAFONT version 0.81. Fonts used include a prototype sans serif in book and slant styles, a proto-prototype Century Schoolbook text style, and a chiseled-look headline font.*

*The original of this document was printed on a Canon LBP-CX with a resolution of 300 dpi.*

## The T<sub>E</sub>X Logo in Various Fonts

Donald E. Knuth

According to the plain T<sub>E</sub>X macro package described in *The T<sub>E</sub>Xbook*,

```
\def\TeX{T\kern-.1667em\lower.5ex
\hbox{E}\kern-.125emX}
```

is the "official" definition of T<sub>E</sub>X's logo. But the plain T<sub>E</sub>X macros are specifically oriented to the Computer Modern fonts. Other typefaces call for variations in the backspacing, in order to preserve the logo's general flavor.

The definition above seems to work satisfactorily with the main serifed fonts of Computer Modern (i.e., with all sizes of `cmr` and `cmsl` and `cmti` and `cmbx`); but sans-serif types are a different story. Indeed, *The T<sub>E</sub>Xbook* itself gives alternative definitions of '\TeX' on pages 418 and 419, one for the font `cmssdc10` at 40pt used in chapter titles (cf. page 36) and one for the `cmssq` fonts used in quotations at the ends of chapters (cf. page 337).

My purpose in this note is to record the various versions of '\TeX' that were actually used to typeset the books in the *Computers & Typesetting* series, so that it will be easy to make forgeries of the particular style used there.

In every case the 'E' has been lowered by .5ex (half of the x-height); the only variation is in the amount of backspacing represented by the

two \kern instructions. Let us therefore consider a "generic" T<sub>E</sub>X logo to be defined by

```
\def\TeX{T\kern\alpha em\lower.5ex
\hbox{E}\kern\beta emX}
```

for some  $\alpha$  and  $\beta$ . The following values of  $(\alpha, \beta)$  were actually used in the published volumes:

| font family         | $\alpha$ | $\beta$ |
|---------------------|----------|---------|
| <code>cmr</code>    | -.1667   | -.125   |
| <code>cmsl</code>   | -.1667   | -.125   |
| <code>cmti</code>   | -.1667   | -.125   |
| <code>cmbx</code>   | -.1667   | -.125   |
| <code>cmssdc</code> | -.2      | -.06    |
| <code>cmssq</code>  | -.2      | 0       |
| <code>cmssqi</code> | -.2      | 0       |
| <code>cmss</code>   | -.15     | 0       |
| <code>cmssi</code>  | -.2      | 0       |
| <code>cmssbx</code> | -.1      | 0       |

(The last three were used only to typeset the jacket copy, not the "real" texts inside. It took a bit of fiddling to get the spacing right.)

I've had little experience with other fonts, but they seem to respond to a similar treatment. For example, my paper on "Literate Programming" in *The Computer Journal* 27 (1984), 97-111, was typeset in a variant of Times Roman, and the standard T<sub>E</sub>X macro worked fine. The captions and references in that article were set in Univers; for that sans-serif font we used  $(\alpha, \beta) = (-.2, 0)$  as in `cmssq`.