

Mode_def's please

Doug Henderson

This article is a follow-up to the plea I made at the 1988 T_EX Users Group meeting in Montréal for people to provide me with METAFONT mode definition settings as they develop them. In order to help people who need to create special fonts or just use METAFONT to create the Computer Modern typefaces on new laser printers or typesetters, I am maintaining a list of these settings for the T_EX Users Group. I do not, however, have every new laser printer or typesetter at my disposal, and even if I did, I would not have the time to test them all. I am simply acting as a repository for the information sent to me by other members of the TUG community. I also spend a good deal of time (via email) explaining how to create new `mode_def` settings for various devices that I cannot test myself. In order for my efforts to benefit the entire TUG community, I need to have everyone report their findings to me after they have created their fonts.

I will include all `mode_def` settings given to me in the `mode_def` article that is published from time to time. Here is a little background on mode definitions. A mode definition (or `mode_def`) is a body of statements that define various things that the METAFONT program needs to know in order to create fonts for a particular printer or typesetter. Below is a typical example of a `mode_def` for an Epson FX-80 dot matrix printer:

```
mode_def epon = % Epson
proofing:=0;
fontmaking:=1;
tracingtitles:=0;
pixels_per_inch:=240;
blacker:=0;
fillin:=0;
o_correction:=.2;
aspect_ratio:=216/240;
endef;
```

The Parameters

The first parameter, *proofing*, determines whether a proof sheet is being created. Proof sheets are useful when you are creating or viewing new characters. Each character is generally 4 to 6 inches square, and fits quite nicely on an 8.5 × 11 inch sheet of paper. To create a proof mode character, you should set the parameter *proofing* to a positive number; 1 will do. In the example above, the value of 0 indicates that we do not wish to create character proof sheets.

The next parameter, *fontmaking*, is set to 1 to indicate that we are creating a font for use with an output device. This causes a TFM (T_EX Font Metrics) file to be created.

The *tracingtitles* parameter determines whether the character description, such as “The letter A”, will be written to the screen or the log file. Higher resolution METAFONT runs sometimes take quite a while, and it’s reassuring to see signs of life (like a title coming on the screen) during the process.

Next, we see the *pixels_per_inch* parameter, which sets (surprise) the number of pixels in an inch for this particular printer. This information is generally supplied by the manufacturer of your printer. Since an Epson FX-80 dot matrix printer has 240 dots per inch horizontally, the setting in the example is 240.

The *blacker* parameter determines how dark a METAFONT pen will stroke through a given character (since some pixels are “burned off” during the pixel rendering process). In general, the higher the resolution of the laser printer or typesetter, the less we need to correct for pixels being shaved off and, consequently, the lower the *blacker* value needs to be.

The *fillin* parameter determines how much “ink” needs to be taken out of the corners where two diagonal strokes meet in characters such as the V and M. Some filling in occurs here and makes certain characters appear darker than they should be. A positive *fillin* value will remove pixels where needed.

The *o_correction* determines how much the bottoms of characters such as O will overshoot the baseline. These three parameters, *blacker*, *fillin*, and *o_correction* are the ones that need to be experimented with to create a new mode definition for a laser printer or typesetter, since changing these values is what really tunes a font correctly for a specific printer.

The final parameter, *aspect_ratio*, corrects for nonsquare pixel ratios. The Epson FX-80, for instance, has 240 dots horizontally and 216 dots vertically. To let METAFONT know this, we supply the *aspect_ratio* setting of 216/240 (vertical/horizontal).

Font-Tuning Advice

Before I give my two cents worth of advice on font tuning, I would like to mention some other people who have done nice work of testing fonts with various engine types. In TUGboat Volume 8, Number 1 (pages 29–33), there is an excellent article entitled “Write-white printing engines and tuning fonts with

METAFONT" by Neenie Billawala. In the following TUGboat issue, Volume 8, Number 2 (pages 128-129) "Blacker Thoughts" by John Gourlay also has excellent advice on the specifics involved with testing fonts for a Xerox 2700 laser printer. John also has some interesting discoveries as to how the *blacker* parameter really works.

For my own first piece of advice, I strongly suggest you have at least version 1.3 of METAFONT, the most recent version being 1.5a. This is because Professor Knuth made some changes to the base files that specifically help write-white engine tuning. This is a good start.

To correctly tune a set of fonts for your printer, you should be altering the values to determine the best settings. Also, you should test your *blacker*, *fillin*, and *o_correction* settings on a number of different-sized fonts. I test with at least the fonts *cmr5*, *cmr10*, and *cmr17* so I can get a feel for what these values do at a very small point size (*cmr5*=5pt), at the largest point size (*cmr17*=17.28pt) and, of course, at the point size that Computer Modern was patterned after, *cmr10*.

The model for what Computer Modern Roman fonts should look like (and what we should be striving for as font tuners) can be found in any one of the five-volume set, *Computers and Typesetting*, by Donald E. Knuth. I highly recommend Volume E, "Computer Modern Typefaces", for viewing the complete set of all 75 Computer Modern fonts. These fonts were all tuned for typesetter quality, and we now have high-resolution character sets against which to compare our samples and tests.

Some generalities

My experience with dot matrix printers has been that they usually don't need any extra black placed in the characters, since they come out nice and black anyway. The *fillin* value works well at 0, and the *o_correction* need not be set very high since the print is fairly dark and overshoots some by bleeding on the paper.

With Canon LBP-CX-based engines, which include the Corona LP 300 and the Apple LaserWriter Plus, I have found that a little black(er) was needed (*corona* = .3; *laserwriter* = .5;) to make good looking character sets. Some people argue that only one value should be used per printer engine type, but I disagree. Small differences are introduced by the companies, which tune these engines, and I believe they are not all the same. Interestingly, I have noticed in working with these similar type print engines that even from engine to engine and from toner cartridge to toner cartridge there is enough

of a difference to warrant slightly different *blacker* settings.

On to new printers

Recently, I have had a rash of requests from people who have purchased new-generation laser printers, among them the HP LaserJet II, the DEC LPS 40, and other Ricoh 408X series or Canon LBP-SX-based engines. Unfortunately, I do not have access to any of these machines for testing.

So, to repeat myself, I would like to ask everyone to please turn in your findings for mode definitions for any new laser printers or phototype-setters you may be testing, and I will try and keep them in order for us in future issues of TUGboat.

Here are the various ways to send me these mode definitions:

bitnet address

DLATEX@CMSA.BERKELEY.EDU

physical mail address

Doug Henderson

Office of the President

Division of Library Automation

300 Lakeside Drive, Floor 8

Oakland, CA 94612-3550

direct hot line

(415) 987-0561