MFtool: A Description of a METAFONT Script-Driven Processing Facility

John M. Crawford
Ohio State University

I thought I might take a moment to describe some of the elements of an environment I've hacked together which helps me in generating fonts using METAFONT. I wanted to develop a mechanism which would allow me to generate new fonts with METAFONT, on demand, without much overhead on my part. What I've come up with is a file-driven facility which allows me to generate fonts from a set of what I'll call script files. Using the command procedure language available with my operating system (Primos), I've put together this processing facility which provides the ability to make repeated invocations of METAFONT while varying specified elements for each call. Perhaps a similar solution could be designed for any given operating environment. Particular elements of this system were added as individual needs arose, so the system design is not necessarily very elegant.

A major element of this scripting file definition describes the font and magnifications desired. This information is provided as a text prefix of the font to be generated, followed (optionally) by the magstep at which the font is to be generated. Multiple requests for a font at various magsteps can be scripted by specifying magsteps within parentheses. A file might then contain

```
      { tfm loaded by plain TeX }
cmr10 (0 0.5 1 2 3 4 5)
cmr12 (0 0.5 1 2)
cmr17 (0 0.5 1 2)
```

This example would invoke METAFONT fifteen times. We also see an example of a comment. Comments may be indicated in two ways: Text preceded by a left curly bracket is discarded. If any given line of text has a space in column 1, then that line is also treated as a comment and ignored.

This system also allows one to select specific base files, or include METAFONT specifications to be fed to METAFONT. For example,

```
spec:\mode=qms
base:&cm
```

could be specified; this must be done before the fonts are chosen. I've also included labels and an "ignore until label" goto facility. Additionally, to improve checkpointing of output, text can be displayed in screen traffic, with a "type" directive. With these elements, I've been able to build specification files and font family files which allow me to generate various sets of METAFONT fonts quickly.

As an example, I wanted to generate the METAFONT logo font for use with a personal computer preview package, mainly as a test. I chose to create a new base file, which specified some new *mode_def*'s I wanted to try. After creating with INIMF a base file called JMC (my initials) containing three specifications for preview fonts, I ran a file similar to the following:

```
base:&jmc
spec:\mode=preview
logo10 (0 0.5 1 2)
spec:\mode=previeww
logo10 (0 0.5 1 2)
spec:\mode=previewww
logo10 (0 0.5 1 2)
```

With that, I'd generated the GF files and TFM file which I'd need to later use the LOGO font on the PC.

In actuality, I can specify several script files to process. I generally divide files into FONT, BASE and SPEC script files. This allows greater flexibility when building a script for a new run. By extending and modifying this basic font generation scheme, I've been able to build various sets of fonts easily.