

Go to the [previous](#), [next](#) section.

SM's Fonts

These are a sub-set of the well-known Hershey fonts @footnote `{created by Dr. A. V. Hershey at the U. S. National Bureau of Standards and illustrated in National Bureau of Standards publication NBS SP-24. The format used in the hershey_oc.dat file was originally due to James Hurt at Cognition, Inc., 900 Technology Park Drive, Billerica, MA 01821. It may be converted to any format except the format distributed by the U. S. NTIS (whatever it may be). We have to tell you all this for copyright reasons, but as distributed the fonts are in the public domain.}` and the available characters are listed in the following table, which were generated from within SM by saying `load fonts fonts`.

For details on SM's implementation of TeX and the ``traditional'` style see the section in the main body of this manual (see section [Drawing Labels and SM's TeX Emulation](#)).

The characters in a font are specified using a programme `read_fonts` which you can use to make binary font files from the list of Hershey characters, using an index file to specify what character should go where. The binary fonts file also specifies which TeX ``definitions'` are available (e.g. `\alpha`). The first 4 bytes of the file are an integer (in binary) that specifies the format of the file; when the format is changed in an incompatible way this number is changed and you will have to rebuild your font files (``make fonts'`)/

The default font table is illustrated at the end of this appendix. Which font file you want to use is specified as the `fonts` entry in your ``.sm'` file. @footnote `{It's possible to resurrect the font table used by pre-2.0 versions of SM, using the index file `old_font_index'}. The fonts.bin fonts have been cleaned up a bit for version 2.0 of SM, although the order of characters in the greek and roman fonts is unchanged. There is a new font, `Old English' or \o or \oe, and a good number of new characters are provided. Neither of these fonts supports the `private' font, that is there in case users desperately need something, when they can make their own binary font file. For example, there is a set of Hershey oriental fonts that could be used (we have it somewhere).`

The complete list of (occidental) Hershey characters is given in a file called `hershey_oc.dat`, and is in the public domain. Each character is specified by a number in the first 5 columns, then a number of strokes in

the next 3, then pairs of letters in the remaining columns up to 72, and in as many 72 character lines as are needed. (Annoyingly, if a line consists of exactly 72 characters, the next must be left blank). Each pair of characters consists of a coordinate, with the origin at (R,R), and the y axis pointing *down*. A ` ' indicates that the next point is a move, otherwise just connect the dots. The very first pair is different, as it specifies the left and right spacing for the character. If this isn't clear, try drawing a few characters on graph paper, character 2001 (roman A) for example. There are a few characters that have traditionally been available in Mongo that are not in the Hershey set, these have been added to the end of the `hershey_oc.dat' file, plus a few that we thought deserved adding.

If you want to create your own characters, the macro `make_char` in `fonts' (i.e. `load fonts make_char`) might help. It uses the cursor to make a string that is (nearly) in the correct form for inclusion in `hershey_oc.dat'

The programme `read_fonts` reads this file, an `index' file that specifies the characters to be put into the fonts, and a list of TeX definitions. The index file consists of character numbers, or ranges consisting of two numbers separated by a minus sign. Comments go from the character `#` to end of line. Each font consists of 96 characters in ascii order, and fonts appear in the index in the order `rm`, `gr`, `sc`, `ti`, `oe`, and `pr`.

The format of the TeX definition file is that each definition has a line to itself, lines starting with a `#` are comments. A line consists of a name, some whitespace, the number of arguments (optional, defaults to zero), the name of the font to use, a single white-space character, and the value of the definition to the end of the line; you can continue onto another line by putting a `\` at the end of the line. You can use any of the normal font specifications, or `cu` which means use the current font.

For example

```
alpha      gr a
alsoalpha  0 cu \gr a
alphatoo   1 gr \#1a
```

defines `\alpha` the conventional way as the character `a` in the greek font, then defines `alsoalpha` in a less efficient way (by specifying the current font, then explicitly switching to greek), then defines `alphatoo` as a large `,` used as `\alphatoo5`. There's no reason why your definitions can't be reasonably complicated, see for example the definition of `\TeX`. The main Makefile prepares your binary font file for you.

Go to the [previous](#), [next](#) section.