

NAME

mf, inifm, mf-nowin – Metafont, a language for font and logo design

SYNOPSIS

mf [*options*] [*&base*] [*file* [*more-input*] | [*\more-input*]

DESCRIPTION

Metafont reads the source program in the specified files and outputs font bitmaps in *gf* (generic font) format and font metrics in *tfm* (T_EX font metric) format. The Metafont language is described in *The Metafontbook*. In general, Metafont programs describe how to draw characters in fonts, or other shapes, in an algorithmic way.

The normal usage is to run

```
mf '\mode=printengine; [mag=magstep(n);]’ input font
```

to process *font.mf* with Metafont.

mf’s command line processing is essentially the same as T_EX’*s*, except that the conventional assignment to *mode* on the command line means that it’s usual to invoke **mf** with several commands, as shown, and not just a filename. Without the initial backslash, Metafont would try to read a file name *mode*, just as T_EX would. On Unix, the single quotes are the simplest way of keeping the shell from (mis)interpreting the backslash, semicolons, parentheses, etc.

Alternatively, you can invoke *mf* without arguments and give the commands on the next line, in response to **mf**’s “***” prompt, without any quotes.

In either case, other Metafont commands, such as *batchmode* (for silent operation) can also appear. The default extension, “.mf”, can be overridden by specifying an extension explicitly.

The name *font* will be the “jobname”, and is used in forming output file names. If Metafont doesn’t get a file name in the first line, the jobname is *mfput*. A log of error messages goes into the file *jobname.log*. The output files are *jobname.tfm* (for the metrics) and *jobname.<number>gf* (for the bitmaps), where <number> depends on the resolution and magnification of the generated font.

Metafont can also be invoked by T_EX, via the (for example) **mktextfm**(1) script (q.v.), when T_EX cannot find a font.

Metafont’s output

Internally, Metafont computes Bezier cubic splines (and straight lines) from the Metafont source program. However, as mentioned above, the main output from Metafont is a bitmap file, that is, given the desired output device (see next section), Metafont does its own rasterization. (Why? Because Knuth’s goal in creating T_EX and Metafont was to completely control the typesetting of his own books, down to the pixel.)

On the other hand, nowadays it is almost universal to want outline fonts in the pdf (or other) output from T_EX (or other document processors). It is possible to extract the splines that Metafont computes from its log file, or to have Metafont generate a high-resolution bitmap and trace that (with, e.g., **autotrace**(1)), to get back to splines, but both of these are inconvenient, at best. It is easier to use MetaPost (**mpost**(1)), which outputs PostScript instead of bitmaps, and outputs the splines instead of their rasterizations. Metafont sources cannot generally be used without change with MetaPost, but they are close. A detailed article on this, in a relatively early project, is by Richard Kinch: MetaFog: Converting Metafont shapes to contours, *TUGboat* 16:3, <https://tug.org/TUGboat/tb16-3/tb48kinc.pdf>.

The *tfm* font metrics file that Metafont outputs contains character dimension (height, width, depth) information, along with ligatures, kerns, and so on; in short, everything needed to do typesetting with the font. This does not include the shapes themselves. OpenType and TrueType fonts, by contrast, incorporate both metric and shape information, along with (potentially) a whole lot more for typesetting in non-Latin scripts, which the original T_EX and Metafont do not attempt to handle, in modern generality. (There have been many projects to do typesetting with T_EX in non-Latin scripts, both predating Unicode and working with Unicode when it became available. Articles in *TUGboat* can be found at <https://tug.org/TUGboat/Contents/listkeyword.html#CatTAGMultilingualDocumentProcessing>.)

The technical specifications of the *gf* (bitmap) and *tfm* (metrics) formats are in the Metafont source program, or any of the utility programs working with the respective formats; see list at end.

Modes (device definitions)

The Metafont *mode* of *printengine* in the above invocation needs to be the name of an actual device (e.g., *ljfour*) or *localfont*. If a mode is not specified, or is not valid, Metafont will default to *proof* mode, which produces large character images for use in font design. Proof mode will generate an output file of .2602gf (36 pixels per point). A proof mode gf can be further processed into a printable document with **gftodvi**(1). This proof mode output is described further in *The Metafontbook*, Appendix H (Hardcopy Proofs), and examples for the entire Computer Modern typeface family are shown in *Computer Modern Typefaces* (Volume E of *Computers & Typesetting*).

Modes are defined with the *mode_def* macro, which adapts the font output to a particular output device, by defining such things as the device resolution. The file *modes.mf* (<https://ctan.org/pkg/modes>) is distributed with T_EX Live and is used by default; it describes the various parameters, defines more than 150 modes (more entries are welcome), and defines *localfont* as *ljfour*, a 600dpi HP LaserJet printer.

Nowadays, output is often viewed online, where high-resolution bitmaps are desirable. Several definitions in *modes.mf* can be used for this: the mode named *pdf* (or *dpdfezzz* for older versions) generates bitmaps at 8000 dpi (other parameters are canonical). If you want a lower resolution, similar canonical modes are *supre* at 2400 and *ultr* at 1200.

The Metafontbook gives some more details and examples of modes, particularly in chapter 11 (Magnification and Resolution).

Magsteps

The system of *magsteps* is identical to what is used in T_EX, with values of *n* in the above invocation generally one of 0.5, 1, 2, 3, 4 or 5, although in principle it can be anything. Magnification can also be specified not as a call to magstep but as an arbitrary value, such as 1.315, to create special character sizes. It can also be entirely omitted, as shown above.

Base (.base) selection

Like T_EX, Metafont is normally used with a large body of precompiled macros, called a "base"; the *plain* base is distributed as part of Metafont and is by far the most commonly used.

The Metafont implementation in T_EX Live (a.k.a. Web2C) looks at its command line to see what name it was called under, just as T_EX does. In short, The names **inimf** and **virmf** read *plain.base*, and any other name reads the base file by that name. See **tex**(1) for details. The invocation name **mf-nowin** is a case special to Metafont that uses *mf.base*; see below.

Error handling

See **tex(1)**; Metafont's error recovery is entirely analogous. The environment variable name for the editor is **MFEDIT**.

Online graphics output

Unlike **T_EX**, Metafont can show the results of its processing online, as it runs. Metafont can use most modern displays. Chapter 23 of *The Metafontbook* describes what you can do. Metafont uses environment variables to determine which display device to use.

First it looks for a variable **MFTERM**; if **MFTERM** is not set, and **DISPLAY** is set, the Metafont window support for X is used. (**DISPLAY** must be set to a valid X server specification, as usual.) If neither **MFTERM** nor **DISPLAY** is set, **TERM** is used to guess the window support to use. If it can't find any of these, you get no online output.

The supported values for **MFTERM** and **TERM** are listed at <https://tug.org/texinfohtml/web2c.html#Online-Metafont-graphics>. Since they are mostly old and rarely used, and not enabled by default in **T_EX Live**, the list is not repeated here.

The two **Kpathsea** configuration variables *screen_width* and *screen_depth* define the width and height of the window used by Metafont. They can be set in the environment or a **Kpathsea** configuration file (<https://tug.org/texinfohtml/kpathsea.html#Path-sources>). They default to 1664 and 1200, respectively.

By default, **T_EX Live** is distributed with two Metafont binaries, **mf** and **mf-nowin**. The **mf** binary supports graphics, while the **mf-nowin** binary does not. The **mf-nowin** binary is used by scripts like **mktexpk(1)**, where graphics support is a nuisance rather than something helpful.

OPTIONS

See **tex(1)** for both details of command-line parsing and the options available (apart from the **T_EX**-specific ones). The only Metafont-specific option name is **-base**, which is entirely analogous to **T_EX**'s **-format**.

ENVIRONMENT

See **tex(1)** for general information and more details. The terse descriptions here are just a hint. There are many more environment variables that can affect the path searching behavior.

TEXMFOUTPUT

If set, Metafont attempts to put its output files here, if they cannot be put in the current directory.

MFINPUTS

Search path for *input* files.

MFEDIT

Command template for switching to editor.

MFTERM

Determines the online graphics display. See above.

screen_width, screen_depth

Width and height of online graphics window.

FONT UTILITIES

Here is a selection of the core utility programs that accompany Metafont:

- gftopk** Takes a *gf* file and produces a smaller *pk* font file.
- gftodvi** Produces proof sheets for fonts.
- gftype** Displays the contents of a *gf* file in mnemonics and/or images.
- pktype** Mnemonically displays the contents of a *pk* file.
- mft** Formats a source file as shown in *Computer Modern Typefaces*.

FILES

- \$TEXMFMAIN/metafont/base/plain.mf*
The standard base.
- \$TEXMFMAIN/metafont/misc/modes.mf*
A file of *mode_defs* for many devices. See <https://ctan.org/pkg/modes>.
- *.base* Predigested Metafont base files.
- mf.pool*
Encoded text of Metafont's messages.

BUGS

Knuth still accepts new bug reports for Metafont, but only reviews them every several years. For a list of submitted reports and other information: <https://tug.org/texmfbug>.

HEALTH WARNING

From *The Metafontbook*: “Type design can be hazardous to your other interests. Once you get hooked, you will develop intense feelings about letterforms; the medium will intrude on the messages that you read. And you will perpetually be thinking of improvements to the fonts that you see everywhere, especially those of your own design.”

SEE ALSO

gftopk(1), **gftodvi(1)**, **gftype(1)**, **mft(1)**, **pktogf(1)**, **pltotf(1)**, **tftopl(1)**.
mktxmf(1), **mktxpk(1)**, **mktextfm(1)**.
mpost(1), **tex(1)**.

Donald E. Knuth, *The Metafontbook* (Volume C of *Computers & Typesetting*).

Donald E. Knuth, *Metafont: The Program* (Volume D of *Computers & Typesetting*).

Donald E. Knuth, *Computer Modern Typefaces* (Volume E of *Computers & Typesetting*).

Some online information:

Metafont for beginners, by Geoffrey Tobin: <https://ctan.org/pkg/metafont-beginners>

Metafont font implementations (nearly all descend from Computer Modern):

<https://ctan.org/topic/font-mf>

T_EX Live web pages: <https://tug.org/texlive>

Web2C Texinfo manual and web page: <https://tug.org/web2c>

Kpathsea Texinfo manual and web page: <https://tug.org/kpathsea>

Sources for the T_EX-world literate programs, as pdf: <https://ctan.org/pkg/knuth-pdf>

TUGboat (the journal of the T_EX Users Group, submissions welcome): <https://tug.org/TUGboat/>

Particularly the *Fonts* topic: <https://tug.org/TUGboat/Contents/listkeyword.html#CatTAGFonts>

AUTHORS

Metafont was created by Donald E. Knuth, who implemented it using his WEB system for Pascal programs. It was originally ported to Unix by Paul Richards at the University of Illinois at Urbana-Champaign. The version now included in T_EX Live, the current Unix T_EX distribution, is

generated by the Web2C system originally written by Tomas Rokicki and Tim Morgan. This man page was originally written by Pierre MacKay.

Bug reports in Metafont (exceedingly rare): <https://tug.org/texmfbug>

Public mailing list for implementation questions and reports: <https://lists.tug.org/tex-k>

Public discussion list for all things Metafont (and T_EX): <https://lists.tug.org/texhax>