armTEX 3.141

Reference manual

Compiled and edited by Mark J. Sinke

ABOUT THE SOFTWARE

The software described in this reference manual is a full port of TEX 3.141 and its friends. The TEX programs (**initex**(1), **virtex**(1), **bigvirtex**(1) and **bigvirtex**(1)) are "correct" in the sense that they've passed the *trip* test. They are 99% compatible with their UNIX counterparts, the only differences being the following:

- As FileCore (and so almost all RISC OS filing systems) does not support extensions, they are simulated using directories. This keeps the files corresponding to a document in the same directory, which is especially useful for LATEX. If a document with an extension is written and the corresponding directory doesn't exist, it is created, if possible. But this approach has a disadvantage: if a simple \input file must be read, this costs a full directory table of disc space. The solution is partly built into the UNIX TEX change file: a file wihout extension is tried both wtiout and with ".tex" appended. For another huge category of files support is built into the armTEX programs: files with a ".sty" extension are tried both with and without the extension.
- The UNIX version uses a directory search algorithm to allow subdirectories in a path to be search to any level. But as we cannot see if a directory is a pseudo-file or a "real" directory, the armTEX programs do a one-level search only.
- The UNIX system of links appears not to be working with the current possibilities for file linking. Support is however built in, so if there's a way to create links to the programs in the future, the link to the program reads by default the format file with the same name, e.g., the program "latex" reads "latex.fmt". As for now, there's a system of aliases that works just as well.
- There are two versions of armTEX available. First of all there are **bigvirtex**(1) and **biginitex**(1). These are fully compatible with their UNIX counterparts, but they eat up about 3600 kB of memory, so they can be run only from a virtual task window. As a swap file of about 4 MB will be created, their startup time is long, and the running time depends heavily on the harddisc speed. There are also cut-down versions, that have almost the same memory sizes as their UNIX small counterparts, but some areas have been cut down by quite an amount. Don't despair, however, as I've only needed a big TEX about twice in the four years I've used TEX.

ABOUT THE DESKTOP

As TEX is command-line oriented, I have not made hard attempts to use the desktop, but there are a few facilities:

- A number of filetypes are defined. They are hard-wired into the code, so they can be changed only by recompiling. As they have been chosen at random in the "user file type area", a file type clash may occur. The file type currently recognised are "TeX", "LaTeX", "METAFONT", "GF Font", "PK Font" and "TFM". The default run actions (e.g. on a double click) are resp. run *virtex &plain*, run *virtex &lplain*, run *mf &plain*, run *gftopk*, run *pktogf* and run *tftopl*. Their load actions are undefined for "TeX", "LaTeX" and "METAFONT", so shift-double clicking will result in loading them into the active editor. The load action for "GF Font" is to run **gftofnt**, a program not (yet) described in this manual, that (tries to) convert a GF file into a RISC OS compatible bitmap file. This program might become obsolete if **mf2ps** works ok (then we run **mf2ps** to create a PostScript font and then **!T1Font** to create an outline font, which is much smaller than any bitmap created by **gftofnt**.
- The concept of a "current directory" is built into any command-line program, and TEX is no exception. The workaround here is that the if first file loaded in has an absolute path, like in *SCSI::Disc4.$.TeX.mydoc.tex*, then the path to this file, *SCSI::Disc4.$.TeX*, is used as the "current directory". This way, TEX is independent of the current directory. See **tex**(1) for more details.
- TEX and friends are run in a task window. For the big versions of the programs, you need a "virtual task window", i.e., a task window having a virtual memory manager. **!Virtual**, supplied with the distribution does this job.

ABOUT THIS MANUAL

The rest of this manual consists of converted UNIX manual pages for each of the programs.

ABOUT THE AUTHOR

I am Mark Sinke, a student of the Eindhoven University of Technology. I can be reached at `marks@stack.urc.tue.nl` or by writing to:
Mark J. Sinke
Mendelssohnstraat 5
5144 GD WAALWIJK
The Netherlands

Please send me bug reports, comments, etc.

P.S. The word author refers to this introduction and some change files.

NAME

amslatex – structured text formatting and typesetting

SYNOPSIS

latex [first line]

DESCRIPTION

$\mathcal{AMS}$-LaTeX is an extension of LaTeX (see *LaTeX – A Document Preparation System*) that provides the powerful mathematical typesetting macros of $\mathcal{AMS}$-TeX (see *The Joy of TeX*) within the syntax of LaTeX. It should be contrasted with LA$\mathcal{MS}$-TeX (**lamstex**(1)), which extends $\mathcal{AMS}$-TeX with LaTeX-like features.

$\mathcal{AMS}$-LaTeX documents can be processed by **latex**(1), so a separate executable program named **amslatex** is not usually provided.

$\mathcal{AMS}$-LaTeX is a TeX macro package, not a modification to the TeX source program, so all the capabilities described in **tex**(1) are present.

The $\mathcal{AMS}$-LaTeX macros encourage writers to think about the content of their documents, rather than the form. The ideal, not always realized, is to have no formatting commands (like "switch to italic" or "skip 2 picas") in the document at all; instead, everything is done by specific markup instructions: "emphasize", "start a section".

$\mathcal{AMS}$-TeX is the official typesetting system of the American Mathematical Society, and nearly all of its publications are typeset using $\mathcal{AMS}$-TeX. The online version of AMS Math Reviews uses $\mathcal{AMS}$-TeX input syntax for display of mathematical material. AMS authors can provide editors with computer-readable $\mathcal{AMS}$-TeX or $\mathcal{AMS}$-LaTeX files.

FILES

TeXInputs:amslatex.tex Sample LaTeX document illustrating the *amsbook* document style.

SEE ALSO

amstex(1), **lamstex**(1), **latex**(1), **tex**(1), **slitex**(1).
Leslie Lamport, *LaTeX — A Document Preparation System*, Addison-Wesley, 1985,
ISBN 0-201-15790-X.
Michael Spivak, *The Joy of TeX*, 2nd edition, Addison-Wesley, 1990, ISBN 0-8218-2997-1.

NAME

amstex – structured text formatting and typesetting

SYNOPSIS

amstex [first line]

DESCRIPTION

The $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX language is described in the book *The Joy of TEX*. $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX is a TEX macro package, not a modification to the TEX source program, so all the capabilities described in **tex**(1) are present.

The $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX macros encourage writers to think about the content of their documents, rather than the form. The ideal, not always realized, is to have no formatting commands (like "switch to italic" or "skip 2 picas") in the document at all; instead, everything is done by specific markup instructions: "emphasize", "start a section".

$\mathcal{A}\mathcal{M}\mathcal{S}$-TEX is the official typesetting system of the American Mathematical Society, and nearly all of its publications are typeset using $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX. The online version of AMS Math Reviews uses $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX input syntax for display of mathematical material. AMS authors can provide editors with computer-readable $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX files.

For authors more familiar with LATEX, an AMS-supported variant called $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX is available.

SEE ALSO

amslatex(1), **latex**(1), **tex**(1), **slitex**(1).
Michael Spivak, *The Joy of TEX*, 2nd edition, Addison-Wesley, 1990, ISBN 0-8218-2997-1.

NAME

bibtex – make a bibliography for LaTeX and TeX

SYNOPSIS

bibtex [**-min-crossrefs**=*number*] [**-verbose**] [*auxname*]

DESCRIPTION

BibTeX reads the top-level auxiliary (*.aux*) file that was output during the running of **latex**(1) or **tex**(1) and creates a bibliography (*.bbl*) file that will be incorporated into the document on subsequent runs of LaTeX or TeX. The *auxname* on the command line must be given without the *.aux* extension. If you don't give the *auxname*, the program prompts you for it.

BibTeX looks up, in bibliographic database (*.bib*) files specified by the \bibliography command, the entries specified by the \cite and \nocite commands in the LaTeX or TeX source file. It formats the information from those entries according to instructions in a bibliography style (*.bst*) file (specified by the \bibliographystyle command, and it outputs the results to the *.bbl* file.

The LaTeX manual explains what a LaTeX source file must contain to work with BibTeX. Appendix B of the manual describes the format of the *.bib* files. The 'BibTeXing' document describes extensions and details of this format, and it gives other useful hints for using BibTeX.

OPTIONS

The **-min-crossrefs** option defines the minimum number of **crossref** required for automatic inclusion of the crossref'd entry on the citation list; the default is two. Without the **-verbose** option, BibTeX operates silently. With it, a banner and progress reports are printed on *stdout*.

ENVIRONMENT

BibTeX searches the directories in the path defined by the BstInputs$Path environment variable for for *.bst* files. If BstInputs$Path is not set, it uses TeXInputs$Path; and if TeXInputs$Path is not set, it uses the system default:

$.TeX.TeXInputs..

For *.bib* files, it uses the BibInputs$Path environment variable if that is set, otherwise the default:

$.TeX.BibInputs.

See **tex**(1) for the details of the searching.

If the environment variable TeXMFOutput$Dir is set, BibTeX attempts to put its output files in it, if they cannot be put in the current directory. Again, see **tex**(1). No special searching is done for the *.aux* file.

FILES

TeXInputs:.bst*	standard style files
btxdoc.tex	"BibTeXing" — LaTeXable documentation for general BibTeX users
btxhak.tex	"Designing BibTeX Styles" — LaTeXable documentation for style designers
btxdoc.bib	database file for those two documents
xampl.bib	database file giving examples of all standard entry types
btxbst.doc	template file and documentation for the standard styles

All those files should be available somewhere on your system.

The host math.utah.edu has a vast collection of *.bib* files available for anonymous ftp, including references for all the standard TeX books and a complete bibliography for TUGboat.

SEE ALSO

latex(1), **tex**(1).
Leslie Lamport, *LaTeX — A Document Preparation System*, Addison-Wesley, 1985, ISBN 0-201-15790-X.

AUTHOR

Oren Patashnik, Stanford University. This man page describes the web2c version of BibTeX. Other ports of BibTeX, such as Donald Knuth's version using the Sun Pascal compiler, do not have the same path searching implementation, or the command-line options.

NAME

dvitype – translate a dvi file for humans

SYNOPSIS

dvitype *dvi_file_name*

DESCRIPTION

The **dvitype** program translates a DVI (DeVice Independent) file output by (for example) **tex**(1) or **gftodvi**(1), to a file that humans can read. It also serves as a DVI file-validating program (i.e., if **dvitype** can read it, it's correct) and as an example of a DVI-reading program for future device drivers.

The output file can include all commands, just the important ones, or none at all (in which case only errors are reported). A subinterval of pages may be selected for transliteration; the magnification and resolution of the "output device" may be changed; and so on. All options are specified with an on-line dialog.

The *dvi_file_name* on the command line must be complete. The *.dvi* extension is not supplied if omitted. The output goes to *stdout*.

ENVIRONMENT

The environment variable TeXFonts$Path is used to search for the TFM files used in the DVI file. See **tex**(1) for the details of the searching. If TeXFonts$Path is not set, it uses the system default:

,$.TeX.Fonts..

SEE ALSO

gftype(1), **pktype**(1).
Donald E. Knuth, *TEXware*.

AUTHORS

Donald E. Knuth wrote the program. It was published as part of the *TEXware* technical report, available from the TEX Users Group. Howard Trickey and Pavel Curtis originally ported it to UNIX.

NAME

etex – extended plain TeX

SYNOPSIS

etex [first line]

DESCRIPTION

etex is an extension of plain TEX described in the book *TEX for the Impatient*. It adds these features (among others) to plain TEX:

- left-justified displays
- double-column output
- tables of contents
- *hrule* and *vrule* with a different default than 0.4pt
- time of day
- verbatim file listing
- generalized footnotes
- blank and black boxes
- citations using BIBTEX

FILES

eplain.doc.eplain.texi TEXable documentation of **etex** macros in **texinfo** format.

The absolute paths to these files may be different on your system.

The complete *eplain* distribution is available via anonymous ftp to the Internet host *ftp.cs.umb.edu* from the file *pub/tex/eplain.tar.Z* which is linked to another file containing the most recent version of the package. Major TEX archive sites should also have it.

SEE ALSO

emacs(1), **info**(1), **initex**(1), **tex**(1), **xinfo**(1).
Karl Berry, *eplain* , TUGboat 11(4), 571–572, 1990.
Paul W. Abrahams with Karl Berry and Kathryn A. Hargreaves, *TEX for the Impatient*, Addison-Wesley, 1990, ISBN 0-201-51375-7.

NAME

gftodvi – make proof sheets from generic font files

SYNOPSIS

gftodvi [–**overflow-label-offset**=*real*] [–**verbose**] [*gf_file_name*]

DESCRIPTION

The **gftodvi** program converts a generic font (*gf*) file output by, for example, **mf**(1), to a device independent (DVI) file (that can then be typeset using the same software that has already been written for TEX). The characters in the *gf* file will appear one per page, with labels, titles, and annotations as specified in Appendix H (Hardcopy Proofs) of *The METAFONTbook*.

gftodvi uses other fonts in addition to the main *gf* file. A 'gray' font is used to typeset the pixels that actually make up the character. (We wouldn't want all the pixels to be simply black, since then labels, key points, and other information would be lost.) A 'title' font is used for the information at the top of the page. A 'label' font is used for the labels on key points of the figure. A 'slant' font is used to typeset diagonal lines, which otherwise have to be simulated using horizontal and vertical rules. The default gray, title, and label fonts are *gray*, *cmr8*, and *cmtt10*, respectively; there is no default slant font.

To change the default fonts, you can give **special** commands in your METAFONT source file, or you can change the fonts online. An online dialog ensues if you end the *gf_file_name* with a '/'. For example,

```
gftodvi cmr10.300gf/
Special font substitution:  grayfont black
OK; any more?  grayfontarea SCSI::Disc4.$.TeX.TeXFonts
OK; any more?  slantfont SCSI::Disc4.$.TeX.TeXFonts.Grayfonts.slantim6
OK; any more?  ⟨RET⟩
```

will use *SCSI::Disc4.$.TeX.TeXFonts.black* as the 'gray' font and *SCSI::Disc4.$.TeX.TeXFonts.GrayFonts.slantim6* as the 'slant' font (this name indicates a font for lines with slope 1/6 at the resolution of an Imagen printer).

The *gf_file_name* on the command line must be complete. (The program prompts you for it if you don't give it.) Because the resolution is part of the extension, it would not make sense to append a default extension as is done with TEX or DVI-reading software. The output file name defaults to the same root as the *gf* file, with the *dvi* extension added. For example, the input file *cmr10.2602gf* would become *cmr10.dvi*.

OPTIONS

The argument to –**overflow-label-offset** specifies the distance from the right edge of the character bounding box at which the overflow equations (if any) are typeset. The value is given in TEX points. The default is a little over two inches.

Without the –**verbose** option, **gftodvi** operates silently. With it, a banner and progress report are printed on *stdout*.

ENVIRONMENT

gftodvi looks for *gf_file_name* using the environment variable GFFonts$Path. If that is not set, it uses the variable TeXFonts$Path. If that is not set, it uses the system defaults:

GFFonts$Path .,*$.TeX.Fonts..*
TeXFonts$Path .,*$.TeX.Fonts..*

See **tex**(1) for the details of the searching.

FILES

TeXFonts:gray.tfm, ...	the default fonts
MFInputs:gray.mf, ...	the METAFONT sources (perhaps)

SEE ALSO

tex(1), **mf**(1).
Donald E. Knuth, *The METAFONTbook* (Volume C of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13445-4.
Donald E. Knuth et al., *METAFONTware*.

AUTHORS

Donald E. Knuth wrote the program. It was published as part of the *METAFONTware* technical report, available from the TEX Users Group. Paul Richards ported it to UNIX.

NAME

gftopk – convert generic font files to packed font files

SYNOPSIS

gftopk [**−v**] *gf_file_name* [*pk_file_name*]

DESCRIPTION

The **gftopk** program converts a generic font file output by, for example, **mf**(1), to a packed font file for use by DVI-reading programs. Packed font files (*pk* files) are much smaller than the corresponding *gf* files, so they are generally the font format of choice.

The *gf_file_name* on the command line must be complete. Because the resolution is part of the extension, it would not make sense to append a default extension as is done with TEX or DVI-reading software. The *pk_file_name* defaults to the same (stripped) name as *gf_file_name*, and it is placed in the current working directory with the *pk* suffix replacing *gf*. For example, the input file *cmr10.300gf* would become *cmr10.300pk*.

OPTIONS

Unless the **−v** switch is given, **gftopk** operates silently. With **−v**, the version number of the program and statistics about the packing are output to *stdout*.

ENVIRONMENT

gftopk looks for *gf_file_name* first in the current directory. If it is not present there, it uses the environment variable GFFonts$Path. If that is not set, it uses the variable TeXFonts$Path. If that is not set, it uses the system defaults:

GFFonts$Path *,$.TeX.Fonts..*
TeXFonts$Path *,$.TeX.Fonts..*

See **tex**(1) for the details of the searching.

SEE ALSO

gftype(1), **pktogf**(1), **pktype**(1).
Donald E. Knuth et al., *METAFONTware*.

AUTHORS

Tomas Rokicki wrote the program. It was published as part of the *METAFONTware* technical report, available from the TEX Users Group. Paul Richards originally ported it to UNIX.

NAME

gftype – translate a generic font file for humans to read

SYNTAX

gftype [−**i**] [−**m**] *gf_file_name* [*output_file_name*]

DESCRIPTION

The **gftype** program translates a *gf* (generic font) file output by, for example, **mf**(1), to a file that humans can read. It also serves as a *gf* file-validating program (i.e., if **gftype** can read it, it's correct) and as an example of a *gf*-reading program for other software that wants to read *gf* files.

The *gf_file_name* on the command line must be complete. Because the resolution is part of the extension, it would not make sense to append a default extension as is done with TEX or DVI-reading software. If no *output_file_name* is specified, the output goes to *stdout*.

OPTIONS

The output file includes bitmap images of the characters, if −**i** is given on the command line, and a symbolic listing of the *gf* commands, if −**m** is specified.

ENVIRONMENT

gftype looks for *gf_file_name* using the environment variable GFFonts$Path. If that is not set, it uses the variable TeXFonts$Path. If that is not set, it uses the system defaults:

GFFonts$Path *,$.TeX.Fonts..*
TeXFonts$Path *,$.TeX.Fonts..*

SEE ALSO

dvitype(1), **pktype**(1).
Donald E. Knuth et al., *METAFONTware*.

AUTHORS

David Fuchs and Don Knuth wrote the program. It was published as part of the *METAFONTware* technical report, available from the TEX Users Group. Paul Richards originally ported it to UNIX.

NAME

initex – initial TeX

DESCRIPTION

initex is a special version of the TEX program that has no preloaded macro packages, but is capable of converting a macro package into a special preformatted binary file, called a *format* (*.fmt*) file. That format file can subsequently be read at high speed by **virtex**.

Major macro packages may require TEX to process many thousands of lines of macros, and open and read scores of font files, all of which would contribute to a sizable startup overhead if the job had to be done every time TEX was run. **initex** allows TEX to do the job once, and then save the results in a binary format file that can be later loaded more rapidly by **virtex**.

initex is normally required only at the time TEX is installed, or whenever major macro packages are updated. Thus, it will be rare for anyone but system installers to invoke it. Nevertheless, it is just a normal program without special privileges, so ordinary users can use it to prepare a private format file.

Here is how you can ask **initex** to prepare format files for several major packages:

For plain **tex**:

```
initex plain \dump
```

For **amstex**:

```
initex &plain amstex \dump
```

For **etex**:

```
initex &plain eplain \dump
```

For **lamstex**:

```
initex &plain amstexl \input lamstex \dump
```

For **latex**:

```
initex lplain \dump
```

For **slitex**:

```
initex splain \dump
```

For **texinfo**:

```
initex &plain texinfo @dump
```

The last argument is different from the previous examples because **texinfo** redefines the TEX escape character from backslash to at-sign. As for **amstex**, the file *plain.fmt* must already be available.

Note that several of these require that the format file for plain TEX be available, since the first argument word *&plain* asks for the loading of the file *plain.fmt*.

It does not matter whether the command-line arguments are passed as a single argument, or as separate arguments, since TEX reconstructs the command line anyway.

You must move the created *.fmt* file to a place where **tex**(1) can find it. See **tex**(1) for more details.

In order to be able to use **bigvirtex**(1) you need to create new format files, using **biginitex** instead of **initex**. The difference between **initex** and **biginitex** is internal memory capacity and speed. The **biginitex** program is slower because of disc swapping (it must be run in a virtual task window).

SEE ALSO

amslatex(1), **amstex**(1), **lamstex**(1), **latex**(1), **slitex**(1), **tex**(1), **virtex**(1).

NAME

lamstex – structured text formatting and typesetting

SYNOPSIS

lamstex [first line]

DESCRIPTION

The LA$\mathcal{M}\mathcal{S}$-TEX language is described in the book *LA$\mathcal{M}\mathcal{S}$-TEX: The Synthesis*. LA$\mathcal{M}\mathcal{S}$-TEX is a new system that augments A$\mathcal{M}\mathcal{S}$-TEX with LATEX-like features, but retains the syntactical flavor of A$\mathcal{M}\mathcal{S}$-TEX. It can be compared with A$\mathcal{M}\mathcal{S}$-LATEX, which augments LATEX with A$\mathcal{M}\mathcal{S}$-TEX-like features.

LA$\mathcal{M}\mathcal{S}$-TEX is a TEX macro package, not a modification to the TEX source program, so all the capabilities described in **tex**(1) are present.

The LA$\mathcal{M}\mathcal{S}$-TEX macros encourage writers to think about the content of their documents, rather than the form. The ideal, not always realized, is to have no formatting commands (like "switch to italic" or "skip 2 picas") in the document at all; instead, everything is done by specific markup instructions: "emphasize", "start a section".

SEE ALSO

amslatex(1), **amstex**(1), **latex**(1), **tex**(1), **slitex**(1).
Michael Spivak, *LA$\mathcal{M}\mathcal{S}$-TEX: The Synthesis*, The TEXplorators Corporation, 3701 W. Alabama, Suite 450-273, Houston, TX 77027, USA, 1989.

NAME

latex – structured text formatting and typesetting

SYNOPSIS

latex [first line]

DESCRIPTION

The LaTeX language is described in the book *LaTeX – A Document Preparation System*. LaTeX is a TeX macro package, not a modification to the TeX source program, so all the capabilities described in **tex**(1) are present.

The LaTeX macros encourage writers to think about the content of their documents, rather than the form. The ideal, not always realized, is to have no formatting commands (like "switch to italic" or "skip 2 picas") in the document at all; instead, everything is done by specific markup instructions: "emphasize", "start a section".

SEE ALSO

amslatex(1), **amstex**(1), **slitex**(1), **tex**(1),
Leslie Lamport, *LaTeX — A Document Preparation System*, Addison-Wesley, 1985,
ISBN 0-201-15790-X.

NAME

mf, inimf, virmf – Metafont, a language for alphabet design

SYNOPSIS

mf [first line]
inimf [first line] or **biginimf** [first line]
virmf [first line] or **bigvirmf** [first line]

DESCRIPTION

METAFONT reads the program in the specified files and outputs font rasters (in *gf* format) and font metrics (in *tfm* format). The METAFONT language is described in *The METAFONT book*.

Like TEX, METAFONT is normally used with a large body of precompiled macros, and font generation in particular requires the support of several macro files. The basic program as compiled is called **inimf**; it can be used to precompile macros into a *.base* file. The **virmf** variant is used to reload the *.base* files quickly. Typically, **virmf** isn't used directly. There is a set of aliases that automatically load a base file. You can load a different base by saying, e.g., *virmf &mybase*. The two programs **biginimf** and **bigvirmf** are versions of METAFONT that have much more internal memory, but they need to be run in a virtual task window and so they're slower because of disc swapping.

As described in *The METAFONT book*, the command line (or first input line) should otherwise begin with a file name or a `\`controlsequence. The normal usage is to say

mf `\mode=`⟨`printengine`⟩`; [mag=magstep(`*n*`);] input` *font*

to start processing *font.mf*. (Or you can just say *mf* and give the other stuff on the next line, without quotes.) Other control sequences, such as *batchmode* (for silent operation) can also appear. 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*. The default extension, *.mf*, can be overridden by specifying an extension explicitly.

A log of error messages goes into the file *jobname.log*. The output files are *jobname.tfm* and *jobname.*⟨*number*⟩*gf*, where ⟨number⟩ depends on the resolution and magnification of the font. The *mode* in this example is shown generically as ⟨printengine⟩, a symbolic term for which the name of an actual device or, most commonly, the name *localfont* (see below) must be substituted. If the mode is not specified or is not valid for your site, METAFONT will default to *proof* mode which produces large character images for use in font design and refinement. Proof mode can be recognized by the suffix *.2602gf* after the jobname. Examples of proof mode output can be found in *Computer Modern Typefaces* (Volume E of *Computers and Typesetting*). The system of *magsteps* is identical to the system used by TEX, with values generally in the range 0.5, 1.0, 2.0, 3.0, 4.0 and 5.0. A listing of *gf* numbers for 240-dpi, 300-dpi and 360-dpi fonts is shown below.

	240 dpi	**300 dpi**	**360 dpi**
mag:=magstep(0 1)	240	300	360
mag:=magstep(0.5 1)	263	329	394
mag:=magstep(1 1)	288	360	432
mag:=magstep(2 1)	346	432	518
mag:=magstep(3 1)	415	518	622
mag:=magstep(4 1)	498	622	746
mag:=magstep(5 1)	597	746	896

Magnification can also be specified not as a magstep but as an arbitrary value, such as 1.315, to create special character sizes.

Before font production can begin, it is necessary to set up the appropriate base files. The minimum set of components for font production for a given print-engine is the *plain.mf* macro file and the local

mode_def file. The macros in *plain.mf* can be studied in an appendix to the *METAFONT book*; they were developed by Donald E. Knuth, and this file should never be altered except when it is officially upgraded. Each *mode_def* specification helps adapt fonts to a particular print-engine. There is a regular discussion of *mode_def*s in *TUGboat*, the journal of the TEX Users Group. The local ones in use on this computer should be in *MFInputs:modes.mf*. With only *plain.mf* and the *modes* file loaded it is possible to create fonts of simple characters, such as those used for the METAFONT logo, and those used for the LATEX line and circle fonts, but the production of Computer Modern fonts would be facilitated by making a *cmmf.base* file (which includes the macros in *cmbase.mf* as well as those in *plain.mf*).

Several environment variables can be used to set up directory paths to search when METAFONT opens a file for input. For example, the command

```
Set MFInputs$Path ,SCSI::Disc4.$.TeX.MFInputs..,SCSI::Disc4.$.TeX.
```

would cause all invocations of METAFONT and its derivatives to look for \input files first in the current directory, then in a directory *SCSI::Disc4.£.TeX.MFInputs* and all its subdirectories (indicated by the double dots) user's , and finally in the directory *SCSI::Disc4.£.TeX*.

Normally, the user will place the command sequence which sets up the MFInputs$Path environment variable in his or her *!Boot* file or a file that's called by *!Boot*.

The *e* response to METAFONT 's error-recovery mode might invoke an editor at the erroneous line of the source file. There is an environment variable, MFEdit, that overrides the default editor. It should contain a string with "%s" indicating where the filename goes and "%d" indicating where the decimal linenumber (if any) goes. For example, an MFEdit string for the **vi** editor can be set with the command

```
Set MFEdit "vi +%d %s"
```

The **ENVIRONMENT** section below lists the relevant environment variables, and their defaults.

A convenient file in the library is *null.mf*, containing nothing. When **mf** can't find the file it thinks you want to input, it keeps asking you for another file name; responding 'null' gets you out of the loop if you don't want to input anything.

ONLINE GRAPHICS OUTPUT

METAFONT can use most modern displays, so you can see its output without printing. Chapter 23 of *The METAFONT book* describes what you can do. This implementation of METAFONT is not yet able to use the WIMP desktop, but a version that will do that is being developed.

ENVIRONMENT

The default values for all environment variables are set at the time of compilation in a file *paths.h*. See **tex**(1) for the details of the searching. If the environment variable TeXMFOuput is set, METAFONT attempts to put its output files in it, if they cannot be put in the current directory. Again, see **tex**(1).

MFInputs$Path	Search path for *input* and *openin* files. Default: *,$.TeX.MFInputs..*
MFBase$Path	Search path for base files. Default: *$.TeX.Formats.*
MFPool$Path	Search path for METAFONT internal strings. (Used by **inimf** only.) Default: *$.TeX.Formats.*
MFEdit	Command template for switching to editor. Default: *vi +%d %s*

FONT UTILITIES

A number of utility programs are available. The following is a partial list of available utilities and their purpose. Consult your local METAFONT guru for details.

- **gftopk** Takes a *gf* file and produces a more tightly packed *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

MFPool:mf.pool	Encoded text of METAFONT's messages
MFBases:.base*	METAFONT base files
MFInputs:plain.mf	The "standard" macro package
MFInputs:modes.mf	The file of *mode_def*s for your site's various printers
*MFInputs:CMSources.**	METAFONT sources for Computer Modern

SUGGESTED READING

Donald E. Knuth, *The METAFONT book* (Volume C of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13445-4.
Donald E. Knuth, *METAFONT: The Program* (Volume D of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13438-1.
Donald E. Knuth, *Computer Modern Typefaces* (Volume E of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13446-2.
TUGboat (the journal of the TEX Users Group).

COMMENTS

Warning: "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), **pltotf**(1), **tftopl**(1).

BUGS

On January 4, 1986 the "final" bug in METAFONT was discovered and removed. If an error still lurks in the code, Donald E. Knuth promises to pay a finder's fee which doubles every year to the first person who finds it. Happy hunting.

AUTHORS

METAFONT was designed 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. This page was mostly written by Pierre MacKay.

NAME

mft – translate Metafont code to TeX code for prettyprinting

SYNOPSIS

mft *mf_file_name[.mf]* [[−**c**] or [*change_file_name[.ch]*]] [[−**s**] or [*style_file_name[.mft]*]]

DESCRIPTION

The **mft** program creates a TEX file from a METAFONT program. It takes appropriate care of typographic details like page layout and the use of indentation, italics, boldface, etc., as illustrated in the book *Computer Modern Typefaces*. Special conventions in METAFONT comments allow you to control things that would not otherwise come out right; section 1 of the MFT source program in the *METAFONTware* report explains these rules.

The command line has one required file name and two optional file names. The required one is a METAFONT source file; there is also an optional change file (which works just as the change files to **tangle**(1) and **weave**(1) do) and an optional style file (which is prepended to everything).

A file name that doesn't contain a dot is always given an extension, either *.mf* (METAFONT) or *.ch* (change) or *.mft* (style). If no style file is specified, the style file *plain.mft* is automatically used. The *change_file_name* is not searched for using any paths. The *.mf* file is searched for using the MFINPUTS environment variable if you have set it, or else the system default:

,$.TeX.MFInputs..

The style file is searched for using the TeXInputs$Path environment variable; its default value is:

$.TeX.TeXInputs..

See **tex**(1) for the details of the searching.

The output TEX file name is formed by using *.tex* in place of the extension of *mf_file_name*.

OPTIONS

Specify command-line options −**c** to suppress the reading of a change file, and −**s** to suppress the reading of a style file.

FILES

TeXInputs:mftmac.tex	TEX macros used by **mft** output.
TeXInputs:plain.mft	Default style file.
TeXInputs:cmbase.mft	Style file for Computer Modern.

SEE ALSO

mf(1), **weave**(1).
Donald E. Knuth, *Computer Modern Typefaces* (Volume E of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13446-2.
Donald E. Knuth et al., *METAFONTware*.

AUTHORS

Donald E. Knuth wrote the program, and he ported it to UNIX with the help of Pierre MacKay and the UNIX port of **weave** by Howard Trickey and Pavel Curtis. The program is published in the *METAFONTware* technical report, available from the TEX Users Group.

NAME

patgen – generate patterns for TeX hyphenation

SYNOPSIS

patgen dictionary_file pattern_file log_file translate_file

DESCRIPTION

The **patgen** program reads the *dictionary_file* and produces the *pattern_file*, writing a transcript of its actions on the *log_file*. The *dictionary_file* should contain a list of hyphenated words and previously-generated patterns.

The patterns thus generated can be read by **initex** for use in hyphenating words. For a (very) long example of *patgen*'s output, see *TeXInputs:hyphen.tex*, which contains the patterns TEX uses for English. At some sites, patterns for several other languages may be available, and the local **tex** programs may have them preloaded; consult your *Local Guide* or your system administrator for details.

All filenames must be complete; no adding of default extensions or path searching is done.

FILES

TeXInputs:hyphen.tex Patterns for English.

SEE ALSO

Frank Liang, *Word hy-phen-a-tion by com-puter*, STAN-CS-83-977, Stanford University Ph.D. thesis, 1983.
Donald E. Knuth, *The TEXbook*, Addison-Wesley, 1986, ISBN 0-201-13447-0, Appendix H.

AUTHORS

Frank Liang wrote the first version of this program. Peter Breitenlohner made a substantial revision in 1991 for TEX 3. The first version was published as the appendix to the *TEXware* technical report, available from the TEX Users Group. Howard Trickey originally ported it to UNIX.

NAME

pktogf – convert packed font files to generic font files

SYNOPSIS

pktogf [**−v**] *pk_file_name* [*gf_file_name*]

DESCRIPTION

The **pktogf** program converts a packed font file (*pk*) to a generic font file (*gf*). Packed font files are much smaller than the corresponding *gf* files, but some DVI readers only understand *gf* files.

The *pk_file_name* on the command line must be complete. Because the resolution is part of the extension, it would not make sense to append a default extension as is done with TEX or DVI-reading software. The *gf_file_name* defaults to the same (stripped) name as *pk_file_name*, and it is placed in the current working directory with the *gf* suffix replacing *pk*. For example, the input file *io.300pk* would become *io.300gf*.

Double-clicking on a *pk* file also runs **pktogf**. The resulting *gf* file appears in the same directory as the original *pk file*.

OPTIONS

Unless the **−v** switch is given, **pktogf** operates silently. With **−v**, the version number of the program and statistics about the unpacking are output to *stdout*.

ENVIRONMENT

pktogf looks for *pk_file_name* using the environment variable PKFonts$Path. If that is not set, it uses the variable TeXPKs$Path. If that is not set, it uses TeXFonts$Path. If that is not set, it uses the system defaults:

PKFonts$Path *,$.TeX.Fonts..*
TeXFonts$Path *,$.TeX.Fonts..*

See **tex**(1) for the details of the searching.

SEE ALSO

gftopk(1), **gftype**(1), **pktype**(1).
Donald E. Knuth et al., *METAFONTware*.

AUTHORS

Tomas Rokicki wrote the program. Pierre MacKay adapted it for compilation with **web2c**.

NAME

pktype – translate a packed font file for humans to read

SYNTAX

pktype *pk_file_name* [*output_file_name*]

DESCRIPTION

The **pktype** program translates a packed font file (*pk*) (output by, for example, **gftopk**(1) to a file that humans can read. It also serves as a *pk* file-validating program (i.e., if **pktype** can read it, it's correct) and as an example of a *pk*-reading program for other software that wants to read *pk* files.

The *pk_file_name* on the command line must be complete. Because the resolution is part of the extension, it would not make sense to append a default extension as is done with TEX or DVI-reading software. If no *output_file_name* is specified, the output goes to the *stdout*.

The output file gives a compact encoding of the packed encoding, using conventions described in the source code. Run lengths of black pixels alternate with parenthesized run lengths of white pixels, and brackets are used to indicate when a row should be repeated.

ENVIRONMENT

pktype looks for *pk_file_name* using the environment variable PKFonts$Path. If that is not set, it uses the variable TeXPKs$Path. If that is not set, it uses TeXFonts. If that is not set, it uses the system defaults:

PKFonts$Path *,$.TeX.Fonts..*
TeXFonts$Path *,$.TeX.Fonts..*

See **tex**(1) for the details of the searching.

SEE ALSO

dvitype(1), **gftopk**(1), **gftype**(1).
Donald E. Knuth et al., *METAFONTware*.

AUTHORS

Tomas Rokicki wrote the program. Donald E. Knuth originally ported it to UNIX. Pierre MacKay adapted it for compilation with **web2c**.

NAME

pltotf – convert WEB property list files to TEX font metric (tfm) format

SYNOPSIS

pltotf [−**verbose**] *pl_file_name tfm_file_name*

DESCRIPTION

The **pltotf** program translates a (human-oriented) WEB property list file to a (program-oriented) TEX font metric file. Thus, after editing the property list file, a TFM file can be generated for use with, for example, **tex**(1).

Both the *pl_file_name* and the *tfm_file_name* must be complete; no adding of default extensions or path searching is done.

OPTIONS

Without the −**verbose** option, **pltotf** operates silently. With it, a banner and progress report are printed on *stdout*.

SEE ALSO

tftopl(1).
Donald E. Knuth, *TEXware*.
Donald E. Knuth, *The METAFONTbook*, (Volume C of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13445-4.

AUTHORS

Donald E. Knuth wrote the program. It was published as part of the *TEXware* technical report, available from the TEX Users Group. Howard Trickey, Pavel Curtis, and Richard Furuta originally ported it to UNIX.

NAME

pooltype – display a WEB pool file

SYNOPSIS

pooltype *pool_file_name*

DESCRIPTION

The **pooltype** program translates a (program-oriented) WEB string pool file to a (human-oriented) text file.

The *pool_file_name* must be complete; no adding of default extensions or path searching is done.

SEE ALSO

tangle(1).

AUTHORS

Donald E. Knuth wrote the program and originally ported it to UNIX. It was published as part of the *TEXware* technical report, available from the TEX Users Group.

NAME

slitex – make LaTeX slides

SYNOPSIS

slitex [first line]

DESCRIPTION

The **slitex** program is a variant of **latex**(1), designed for generating slides. It is described in Appendix A of *LaTeX – A Document Preparation System*. All the capabilities described in **tex**(1) are present.

SEE ALSO

latex(1), **tex**(1).
Leslie Lamport, *LaTeX – A Document Preparation System*, Addison-Wesley, 1985,
ISBN 0-201-15790-X.

NAME

tangle – translate WEB to Pascal

SYNOPSIS

tangle *webfile[.web]* [*changefile[.ch]*]

DESCRIPTION

The **tangle** program converts a WEB source document into a Pascal program that may be compiled in the usual way with the on-line Pascal compiler (e.g., **pc**(1)). The output file is all in lower case and packed into lines of 72 characters or less, with the only concession to readability being the termination of lines at semicolons when this can be done conveniently.

The WEB language allows you to prepare a single document containing all the information that is needed both to produce a compilable Pascal program and to produce a well-formatted document describing the program in as much detail as the writer may desire. The user of WEB must be familiar with both TEX and Pascal. WEB also provides a relatively simple, although adequate, macro facility that permits a Pascal program to be written in small easily-understood modules.

The command line should have either one or two names on it. The first is taken as the WEB file (and *.web* is added if there is no extension). If there is another name, it is a change file (and *.ch* is added if there is no extension). The change file overrides parts of the WEB file, as described in the WEB system documentation.

The output files are a Pascal file and a string pool file, whose names are formed by adding *.p* and *.pool* respectively to the root of the WEB file name.

SEE ALSO

pc(1), **pxp**(1) (for formatting **tangle** output when debugging), **tex**(1).
Donald E. Knuth, *The WEB System of Structured Documentation*.
Donald E. Knuth, *Literate Programming*, Computer Journal **27**, 97–111, 1984.
Wayne Sewell, *Weaving a Program*, Van Nostrand Reinhold, 1989, ISBN 0-442-31946-0.
Donald E. Knuth, *TEX: The Program* (Volume B of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13437-3.
Donald E. Knuth, *METAFONT: The Program* (Volume D of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13438-1.

These last two are by far the largest extant examples of WEB programs.

There is an active Internet electronic mail discussion list on the subject of literate programming; send a subscription request to `litprog-request@shsu.edu` to join.

AUTHORS

WEB was designed by Donald E. Knuth, based on an earlier system called DOC (implemented by Ignacio Zabala). The **tangle** and **weave** programs are themselves written in WEB. The system was originally ported to UNIX at Stanford by Howard Trickey, and at Cornell by Pavel Curtis.

NAME

tex, virtex – text formatting and typesetting

SYNOPSIS

tex [first line]
virtex [first line] or **bigvirtex** [first line]

DESCRIPTION

TEX formats the interspersed text and commands contained in the named files and outputs a typesetter independent file (called *DVI*, which is short for *DeVice Independent*). TEX capabilities and language are described in *The TEXbook*. The program **bigvirtex** is a version of TEX that has much more internal memory, but it needs to be run in a virtual task window and so it's slower because of disc swapping.

TEX is normally used with a large body of precompiled macros, and there are several specific formatting systems, such as LATEX, which require the support of several macro files. The basic programs as compiled are called **initex** and **virtex**, and are distinguished by the fact that **initex** can be used to precompile macros into a *.fmt* file, which is used by *virtex*. On the other hand, **virtex** starts more quickly and can read a precompiled *.fmt* file, but it cannot create one. It is the version of TEX which is usually invoked in production, as opposed to installation.

Any arguments given on the command line to the TEX programs are passed to them as the first input line. As described in *The TEXbook*, that first line should begin with a filename or a \controlsequence. The normal usage is to say

```
tex paper
```

to start processing *paper.tex.* The name *paper* will be the "jobname", and is used in forming output filenames. If TEX doesn't get a filename in the first line, the jobname is *texput.* The default extension, *.tex*, can be overridden by specifying an extension explicitly.

If there is no *paper.tex* in the current directory, TEX will look through a search path of directories to try to find it. If *paper* is the "jobname", a log of error messages, with rather more detail than normally appears on the screen, will appear in *paper.log,* and the output file will be in *paper.dvi.* The system library directories *TeXInputs:* contain the basic macro package *plain.tex,* described in *The TEXbook*, as well as several others. Except when *.fmt* files are being prepared it is unnecessary to *input plain*, since almost all instances of TEX begin by loading *plain.fmt.* This means that all of the control sequences discussed in *The TEXbook* are known when you invoke **tex**. For a discussion of *.fmt* files, see below.

The *e* response to TEX's error prompt might cause an editor to start up at the current line of the current file. The environment variable TeXEdit can be used to change the editor used. It can contain a string with "%s" indicating where the filename goes and "%d" indicating where the decimal line number (if any) goes. For example, a TeXEdit string for **vi** can be set with the command

```
Set TeXEdit vi +%d %s
```

A convenient file in the library is *null.tex*, containing nothing. When TEX can't find a file it thinks you want to input, it keeps asking you for another filename; responding 'null' gets you out of the loop if you don't want to input anything. You can also type your EOF character (control-D).

The **initex** and **virtex** programs can be used to create fast-loading versions of TEX based on macro source files. The **initex** program is used to create a *format (.fmt)* file that permits fast loading of fonts and macro packages. After processing the fonts and definitions desired, a *dump* command will create the format file. The format file is used by **virtex**. It needs to be given a format filename as the first thing it reads. A format filename is preceded by an &.

Fortunately, it is often not necessary to make explicit references to the format file. The present version of armTEX, has a file *set-tex*, that sets a host of useful aliases that run **virtex** with a suitable format file.

When looking for a font *f*, TEX (and its companion programs) first look for a file starting with *f* in the various font directories (see the next section). If no such file is found, it then looks for a file *texfonts.map* in each of the font directories in turn. Each non-blank non-comment line of *texfonts.map* specifies mappings from one name to another. (Comments start with % and continue to the end of the line.) The target name is the first word (words are separated by spaces or tabs) and the source name is the second. (Subsequent words are ignored, so that information intended for other programs can be given there.) Thus, going back to *f* for a moment, if TEX reads a *texfonts.map* entry that looks like *g f* it will then search for a font file starting with *g*.

ENVIRONMENT

The defaults for all environment variables are set at the time of compilation. All paths are comma-separated, just like RISC OS paths. But as an extra TEX, which does the path searching by itself, allows double dots to be at the end of a directory name. If one of the components in a search path ends with two such dots, e.g.,

SCSI::Disc4.$.TeX.TeXFonts..

then all (first-level) subdirectories of the given path are searched, instead of the directory itself.

All the programs in the **web2c** distribution (as well as some others) use this same search method.

Normally, TEX puts its output files in the current directory. If any output file cannot be opened there, it tries to open it in the directory specified in the environment variable TeXMFOutput$Dir. There is no default value for that variable. For example, if you say *tex paper* and the current directory is not writable, if TeXMFOutput$Dir has the value *SCSI::Disc4.tmp*, TEX attempts to create *SCSI::Disc4.tmp.paper.log* (and *SCSI::Disc4.tmp.paper.dvi*, if any output is produced.)

If an output directory does not exist (which is often the case for *texput*), TEX tries to create it.

When running TEX with an absolute path name (e.g., by double-clicking on a *tex* file, the path to this file is used as the "current directory". This way, TEX is completely independent of the real current directory.

TeXInputs$Path	Search path for *input* and *openin* files. This should probably start with ",", so that user files are found before system files. Default: *$.TeX.TeXInputs..*
TeXFonts$Path	Search path for font metric (*.tfm*) files. Default: *,$.TeX.Fonts..*
TeXFormats$Path	Search path for format files. Default: *$.TeX.Formats.*
TeXPool$Path	Search path for **initex** internal strings. Default: *$.TeX.Formats.*
TeXEdit	Command template for switching to editor. Default: *vi +%d %s*

FILES

TeXInputs:	TEX's library area
TeXPool:tex.pool	Encoded text of TEX's messages.
TeXFonts:texfonts.map	Filename mapping definitions.
TeXFonts:.tfm*	Metric files for TEX's fonts.
TeXFonts:.nnn{gf,pk}*	Character bitmaps for various devices. These files are not used by TEX.
TeXFormats:.fmt*	TEX *.fmt* files.

TeXInputs:plain.tex The "default" macro package.

SEE ALSO

mf(1).
Donald E. Knuth, *The TEXbook*, Addison-Wesley, 1986, ISBN 0-201-13447-0.
Leslie Lamport, *LATEX — A Document Preparation System*, Addison-Wesley, 1985, ISBN 0-201-15790-X.
Michael Spivak, *The Joy of TEX*, 2nd edition, Addison-Wesley, 1990, ISBN 0-8218-2997-1.
TUGboat (the journal of the TEX Users Group).

TRIVIA

TEX, pronounced properly, rhymes with "blecchhh." The proper spelling in typewriter-like fonts is "TeX" and not "TEX" or "tex."

AUTHORS

TEX was designed by Donald E. Knuth, who implemented it using his WEB system for Pascal programs. It was ported to UNIX at Stanford by Howard Trickey, and at Cornell by Pavel Curtis. The version now offered with the UNIX TEX distribution is that generated by the WEB to C system (**web2c**), originally written by Tomas Rokicki and Tim Morgan.

The Archimedes version of TEX (and all its friends) were derived using slightly modified versions of the UNIX change files. The port was done by Mark J. Sinke.

NAME

tftopl – convert TeX font metric (tfm) files to WEB property lists

SYNOPSIS

tftopl [−**charcode-format**=*format*] [−**verbose**] *tfm_file_name* [*pl_file_name*]

DESCRIPTION

The **tftopl** program translates a (program-oriented) font metric file to a (human-oriented) property list file. Thus, you can edit the contents of the TFM files, if the font designer has not done his or her job properly, or if you're encountering strange difficulties, or if you're just curious. It also serves as a TFM-file validating program, i.e., if no error messages are given, the input file is correct.

The *pl_file_name* must be complete; no adding of default extensions or path searching is done. If the *pl_file_name* is not given, standard output is used. The *tfm_file_name* is extended with *.tfm* if it lacks a suffix. Also, path searching is done for the *.tfm* file using the environment variable TeXFonts$Path. If that is not set, **tftopl** uses the system default:

,$.TeX.Fonts..

See **tex**(1) for the details of the searching.

OPTIONS

The argument *format* to −**charcode-format** specifies how character codes are output in the PL file. By default, only letters and digits are output using the **C** integer code (i.e., in ASCII); the others are output in octal. (Unless the font's coding scheme starts with **TeX math sy** or **TeX math ex**, in which case all character codes are output in octal.) If *format* is **ascii**, all character codes that correspond to graphic characters, besides the left and right parentheses, are output in ASCII. Finally, if *format* is **octal**, all character codes are output in octal.

Without the −**verbose** option, **tftopl** operates silently. With it, a banner and progress report are printed on *stdout*.

SEE ALSO

pltotf(1).
Donald E. Knuth, *TEXware*.
Donald E. Knuth, *The METAFONTbook* (Volume C of *Computers and Typesetting*), Addison-Wesley, 1986, ISBN 0-201-13445-4.

AUTHORS

Donald E. Knuth wrote the program. It was published as part of the *TEXware* technical report, available from the TEX Users Group. Howard Trickey and Pavel Curtis originally ported it to UNIX.

NAME

vftovp – convert virtual font (vf) files to virtual property lists

SYNOPSIS

vftovp [−**charcode-format**=*format*] [−**verbose**] *vf_file_name tfm_file_name* [*vpl_file_name*]

DESCRIPTION

The **vftovp** program translates a (program-oriented) virtual font file and its accompanying TEX font metric file to a (human-oriented) property list file. Thus, the virtual font can be edited, and its exact contents can be displayed mnemonically. It also serves as a VF-file validating program; if no error messages are given, the input files are correct.

The *vpl_file_name* must be complete; no adding of default extensions is done.

OPTIONS

The argument *format* to −**charcode-format** specifies how character codes are output in the PL file. By default, only letters and digits are output using the **C** integer code (i.e., in ASCII); the others are output in octal. (Unless the font's coding scheme starts with **TeX math sy** or **TeX math ex**, that is, when all are output in octal.) If *format* is **ascii**, all character codes that correspond to graphic characters, besides the left and right parentheses, are output in ASCII. On the other hand, if *format* is **octal**, all character codes are output in octal.

Without the −**verbose** option, **vftovp** operates silently. With it, a banner and progress report are printed on *stdout*.

ENVIRONMENT

vftovp looks for *vf_file_name* using the environment variable VFFonts$Path. If that is not set, it uses the variable TeXFonts$Path. If that is not set, it uses the system defaults:

VFFonts$Path *,$.TeX.Fonts..*
TeXFonts$Path *,$.TeX.Fonts..*

See **tex**(1) for the details of the searching.

SEE ALSO

pltotf(1), **tftopl**(1), **vptovf**(1).

AUTHORS

Donald E. Knuth wrote the program, based in part on an idea of David Fuchs, starting with the code for **tftopl**(1). Karl Berry adapted it for compilation with **web2c**.

NAME

virtex – virgin TeX

DESCRIPTION

virtex is a version of the TEX program that has no preloaded macro packages, but is capable of loading a special preformatted binary file, called a *format* (*.fmt*) file produced by the **initex**(1) program.

Preprocessing of large macro files and font definitions by **initex** makes it possible to rapidly load complex packages, like plain TEX, $\mathcal{AMS}$-TEX, $\mathcal{AMS}$-LATEX, or LATEX, without the overhead of macro parsing and file opening that would otherwise be experienced.

virtex can be invoked directly, providing it with a name of a format file specially prefixed by an ampersand:

```
virtex &plain filename
```

This causes it to load the binary file named *plain.fmt* from a directory in the standard TeXInputs$Path search path, and then load the text file *filename* and process it.

Because this need occurs frequently, armTeX comes with a file *set-tex* that sets up a system of aliases. See **tex**(1) for more details.

SEE ALSO

initex(1), **tex**(1).

NAME

vptovf – convert virtual property lists to virtual font metrics

SYNOPSIS

vptovf [−**verbose**] *vpl_file_name vfm_file_name tfm_file_name*

DESCRIPTION

The **vptovf** program translates a (human-oriented) property list file to a pair of (program-oriented) files in the VF (virtual font) and TFM (TEX font metric) formats. Thus, a virtual font file can be edited and its exact contents can be displayed mnemonically. New virtual fonts, which map characters as seen by TEX into an arbitrary sequence of low-level typesetting operations, can also be created in this way.

All three file names, *vpl_file_name, vf_file_name,* and *tfm_file_name,* must be complete; no adding of default extensions or path searching is done.

OPTIONS

Without the −**verbose** option, **vptovf** operates silently. With it, a banner and progress report are printed on *stdout*.

SEE ALSO

pltotf(1), **tftopl**(1), **vftovp**(1).

AUTHORS

Donald E. Knuth wrote the program, based in part on an idea of David Fuchs, starting with the code for **pltotf**(1). Karl Berry adapted it for compilation with **web2c**.

NAME

weave – translate WEB to TEX

SYNOPSIS

weave [**−x**] *webfile[.web]* [*changefile[.ch]*]

DESCRIPTION

The **weave** program is used to create a TEX file for viewing the WEB program. It takes appropriate care of typographic details like page layout and the use of indentation, italics, boldface, etc., and it supplies extensive cross-index information that it gathers automatically. The command line arguments are the same as for **tangle** except for the option: **−x** says to omit the index, module name list, and table of contents pages. (A *CONTENTS.tex* file will still be written when the TEX file is processed, however, unless some macros in *webmac.tex* are redefined.)

The output TEX file name is formed by adding *.tex* to the root of the WEB file name.

There are several macros that probably should be redefined by the programmer at the beginning of the WEB file. It is a good idea to set \title to the name of the program. And, to cause output of only changed modules, one can say \let\maybe=\iffalse (usually as the first change in the change file).

FILES

TeXInputs:webmac.tex TEX macros used by **weave** output.

SEE ALSO

tangle(1) for references, authors, and other information about WEB.