

NAME

`tex`, `initex` – text formatting and typesetting

SYNOPSIS

`tex` [*options*] [*&format*] [*file*|*commands*]

DESCRIPTION

Run the \TeX typesetter on *file*, usually creating *file.dvi*. If the file argument has no extension, ".tex" will be appended to it. Instead of a filename, a set of \TeX commands can be given, the first of which must start with a backslash. With a *&format* argument \TeX uses a different set of precompiled commands, contained in *format.fmt*; it is usually better to use the **-fmt** *format* option instead.

\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 's capabilities and language are described in *The \TeX book*. \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.

This version of \TeX looks at its command line to see what name it was called under. If they exist, then both **initex** and **virtex** are symbolic links to the **tex** executable. When called as **initex** (or when the **-ini** option is given) it can be used to precompile macros into a *.fmt* file. When called as **virtex** it will use the *plain* format. When called under any other name, \TeX will use that name as the name of the format to use. For example, when called as **tex** the *tex* format is used, which is identical to the *plain* format. The commands defined by the *plain* format are documented in *The \TeX book*. Other formats that are often available include *latex* and *amstex*.

The non-option command line arguments to the \TeX program are passed to it as the first input line. (But it is often easier to type extended arguments as the first input line, since UNIX shells tend to gobble up or misinterpret \TeX 's favorite symbols, like backslashes, unless you quote them.) As described in *The \TeX book*, that first line should begin with a filename, a *\controlsequence*, or a *&formatname*.

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*. When looking for a file, \TeX looks for the name with and without the default extension (*.tex*) appended, unless the name already contains that extension. 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*.

This version of \TeX can look in the first line of the file *paper.tex* to see if it begins with the magic sequence *%&*. If the first line begins with *%&format -translate-file tcxname* then \TeX will use the named format and translation table *tcxname* to process the source file. Either the format name or the **-translate-file** specification may be omitted, but not both. This overrides the format selection based on the name by which the program is invoked. The **-parse-first-line** option or the **parse_first_line** configuration variable controls whether this behaviour is enabled.

The *e* response to \TeX 's error prompt causes the system default 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 may 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 **emacs** can be

set with the **sh** command

```
TEXEDIT="emacs +%d %s"; export TEXEDIT
```

A convenient file in the library is *null.tex*, containing nothing. When T_EX 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 (usually control-D).

OPTIONS

This version of T_EX understands the following command line options.

- enc** Enable the encT_EX extensions. This option is only effective in combination with **-ini**. For documentation of the encT_EX extensions see <http://www.olsak.net/enctex.html>.
- file-line-error**
Print error messages in the form *file:line:error* which is similar to the way many compilers format them.
- no-file-line-error**
Disable printing error messages in the *file:line:error* style.
- file-line-error-style**
This is the old name of the **-file-line-error** option.
- fmt** *format*
Use *format* as the name of the format to be used, instead of the name by which T_EX was called or a %& line.
- halt-on-error**
Exit with an error code when an error is encountered during processing.
- help** Print help message and exit.
- ini** Start in *INI* mode, which is used to dump formats. The *INI* mode can be used for typesetting, but no format is preloaded, and basic initializations like setting catcodes may be required.
- interaction** *mode*
Sets the interaction mode. The mode can be either *batchmode*, *nonstopmode*, *scrollmode*, and *errorstopmode*. The meaning of these modes is the same as that of the corresponding \commands.
- ipc** Send DVI output to a socket as well as the usual output file. Whether this option is available is the choice of the installer.
- ipc-start**
As **-ipc**, and starts the server at the other end as well. Whether this option is available is the choice of the installer.
- jobname** *name*
Use *name* for the job name, instead of deriving it from the name of the input file.
- kpathsea-debug** *bitmask*
Sets path searching debugging flags according to the bitmask. See the *Kpathsea* manual for details.

-mktex *fnt*

Enable mktex *fnt*, where *fnt* must be either *tex* or *tfm*.

-mltex Enable ML_T_EX extensions. Only effective in combination with **-ini**.**-no-mktex *fnt***

Disable mktex *fnt*, where *fnt* must be either *tex* or *tfm*.

-output-comment *string*

Use *string* for the DVI file comment instead of the date.

-output-directory *directory*

Write output files in *directory* instead of the current directory. Look up input files in *directory* first, then along the normal search path. See also description of the TEXMFOUTPUT environment variable.

-parse-first-line

If the first line of the main input file begins with %& parse it to look for a dump name or a **-translate-file** option.

-no-parse-first-line

Disable parsing of the first line of the main input file.

-progrname *name*

Pretend to be program *name*. This affects both the format used and the search paths.

-recorder

Enable the filename recorder. This leaves a trace of the files opened for input and output in a file with extension *.fls*.

-shell-escape

Enable the `\write18{command}` construct. The *command* can be any shell command. This construct is normally disallowed for security reasons.

-no-shell-escape

Disable the `\write18{command}` construct, even if it is enabled in the *texmf.cnf* file.

-src-specials

Insert source specials into the DVI file.

-src-specials *where*

Insert source specials in certain places of the DVI file. *where* is a comma-separated value list: *cr*, *display*, *hbox*, *math*, *par*, *parent*, or *vbox*.

-translate-file *tcxname*

Use the *tcxname* translation table to set the mapping of input characters and re-mapping of output characters.

-default-translate-file *tcxname*

Like **-translate-file** except that a %& line can overrule this setting.

-version

Print version information and exit.

ENVIRONMENT

See the Kpathsearch library documentation (the ‘Path specifications’ node) for precise details of how the environment variables are used. The **kpsewhich** utility can be used to query the values of the variables.

One caveat: In most $\text{T}_{\text{E}}\text{X}$ formats, you cannot use `~` in a filename you give directly to $\text{T}_{\text{E}}\text{X}$, because `~` is an active character, and hence is expanded, not taken as part of the filename. Other programs, such as Metafont, do not have this problem.

TEXMFOUTPUT

Normally, $\text{T}_{\text{E}}\text{X}$ 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`. There is no default value for that variable. For example, if you say *tex paper* and the current directory is not writable, if `TEXMFOUTPUT` has the value */tmp*, $\text{T}_{\text{E}}\text{X}$ attempts to create */tmp/paper.log* (and */tmp/paper.dvi*, if any output is produced.) `TEXMFOUTPUT` is also checked for input files, as $\text{T}_{\text{E}}\text{X}$ often generates files that need to be subsequently read; for input, no suffixes (such as “.tex”) are added by default, the input name is simply checked as given.

TEXINPUTS

Search path for `\input` and `\openin` files. This probably start with “.”, so that user files are found before system files. An empty path component will be replaced with the paths defined in the *texmf.cnf* file. For example, set `TEXINPUTS` to “.:/home/user/tex:” to prepend the current directory and “/home/user/tex” to the standard search path.

TEXFORMATS

Search path for format files.

TEXPOOL

search path for **tex** internal strings.

TEXEDIT

Command template for switching to editor. The default, usually **vi**, is set when $\text{T}_{\text{E}}\text{X}$ is compiled.

TFMFORMATS

Search path for font metric (*.tfm*) files.

FILES

The location of the files mentioned below varies from system to system. Use the **kpsewhich** utility to find their locations.

texmf.cnf

Configuration file. This contains definitions of search paths as well as other configuration parameters like **parse_first_line**.

tex.pool

Text file containing $\text{T}_{\text{E}}\text{X}$ ’s internal strings.

texfonts.map

Filename mapping definitions.

**.tfm* Metric files for $\text{T}_{\text{E}}\text{X}$ ’s fonts.

**.fmt* Predigested $\text{T}_{\text{E}}\text{X}$ format (*.fmt*) files.

\$TEXMFMAIN/tex/plain/base/plain.tex

The basic macro package described in the *T_EXbook*.

NOTES

This manual page is not meant to be exhaustive. The complete documentation for this version of $\text{T}_{\text{E}}\text{X}$ can be found in the info manual *Web2C: A TeX implementation*.

BUGS

This version of T_EX implements a number of optional extensions. In fact, many of these extensions conflict to a greater or lesser extent with the definition of T_EX. When such extensions are enabled, the banner printed when T_EX starts is changed to print **TeXk** instead of **TeX**.

This version of T_EX fails to trap arithmetic overflow when dimensions are added or subtracted. Cases where this occurs are rare, but when it does the generated *DVI* file will be invalid.

SEE ALSO

mf(1),

Donald E. Knuth, *The T_EXbook*, Addison-Wesley, 1986, ISBN 0-201-13447-0.

Leslie Lamport, *L^AT_EX – A Document Preparation System*, Addison-Wesley, 1985, ISBN 0-201-15790-X.

K. Berry, *Eplain: Expanded plain T_EX*, <ftp://ftp.cs.umb.edu/pub/tex/eplain/doc>.

Michael Spivak, *The Joy of T_EX*, 2nd edition, Addison-Wesley, 1990, ISBN 0-8218-2997-1.

TUGboat (the journal of the T_EX Users Group).

TRIVIA

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

AUTHORS

T_EX was created 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 T_EX distribution is that generated by the WEB to C system (**web2c**), originally written by Tomas Rokicki and Tim Morgan.

The encT_EX extensions were written by Petr Olsak.