

NAME

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

SYNOPSIS

`tex` [*options*] [*&format*] [*file* [*more-input*] | [*\more-input*]

DESCRIPTION

Run the T_EX typesetter on *file* [.tex], usually creating *file.dvi*. If the file argument has no extension, ".tex" will be appended to it.

The normal usage is to run

```
tex paper
```

to process *paper.tex* with T_EX. The name *paper* will be the so-called "jobname", and is used in forming output filenames. If T_EX doesn't get a filename in the first line, the jobname is *texput*. When looking for a file, T_EX 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 be written to *paper.log*, and the output file will be *paper.dvi*. The **--jobname** option sets the jobname explicitly.

T_EX formats the interspersed text and commands contained in the input and outputs a so-called *DVI* file, which is short for *DeVice Independent*. DVI files are a device-independent binary representation of the pages in the output document. They can be displayed online (e.g., **xdvi**(1)), translated to PDF (**dvipdfmx**(1)), PostScript (**dvips**(1)), or otherwise manipulated. Other engines (**pdf_{te}x**(1), **lua_{te}x**(1)), descendants of original T_EX, can output PDF directly.

T_EX's capabilities and language are described in many places (see the references), including Donald Knuth's original book, *The T_EXbook*. T_EX is normally used with a large body of precompiled macros, called a format. Knuth's original format is named *plain*. Other formats, notably **l_atex**(1), are much larger, but still implemented entirely as T_EX macros.

Command line processing

To process the command line, after first reading any *option*(s) (listed below), if the first non-option argument does not begin with an escape character (normally a backslash), T_EX internally prepends **\input** to the non-option arguments. Whether the **\input** was prepended or not, T_EX then processes the text as normal T_EX input, including macro expansion.

With no non-option arguments, T_EX gives you a ** interactive prompt; whatever text you enter is processed in exactly the same way as non-option command line arguments.

Examples:

tex foo executes "**\input** foo".

tex foo bar

executes "**\input** foo bar", which inputs *foo.tex* and then typesets the string "bar".

tex \end

executes **\end**; here the double backslash is one common way of escaping the backslash for the shell. T_EX must see only one backslash.

To reiterate, the text of an input filename, as well as any text following a filename, is subject to T_EX's macro expansion. As a result, a ~ character is normally not usable within a filename, since ~ is conventionally defined to be an active character that expands into several primitive tokens,

which can't be part of a filename. You can usually use `\string~` to get a literal `~` character. More on input filenames: <https://tug.org/texinfohtml/web2c.html#Input-filenames>

Format (.fmt) selection

With an argument `&format`, T_EX reads the set of precompiled commands contained in the (binary) file `format.fmt` instead of the default `tex.fmt` (searched for using the `TEXFORMATS` path; see environment section below). This is sometimes more convenient than the `-fmt format` option, which is equivalent.

The T_EX implementation in T_EX Live (a.k.a. Web2C) looks at its command line to find the name it was invoked as. When called as **initex** (or when the `-ini` option is given, which is equivalent) it can be used to precompile macros into a `.fmt` file. When called as **virtex** it will use the *plain* format. Nowadays, if either or both of the executables **initex** and **virtex** exist, they are symbolic links to the **tex** executable; the **virtex** symlink is not made by default in T_EX Live.

When invoked as any other name, T_EX will use that name as the name of the format to use. Most notably, when invoked as **tex** the format file `tex.fmt` is read, which is identical to the *plain* format. The commands defined by the *plain* format are described in Appendix B of *The T_EXbook*, among other places (see references below). The other T_EX engines follow the same convention; for example, the **latex** command is a symlink to the **pdftex** binary, and reads `latex.fmt`.

This implementation of T_EX can look in the first line of the file `paper.tex` to see if it begins with the magic two-character sequence `%&`. If the first line begins with `%&format -translate-file tcxname` then T_EX will use the named format and (if specified) 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 behavior is enabled. It is disabled for the original T_EX, and enabled for all other engines.

Input encodings

T_EX, predating Unicode as it does, reads input byte by byte, and has no native recognition for UTF-8 or any other encoding. This is also the case for pdfT_EX. However, to a large extent, UTF-8 can be parsed at the T_EX macro level, and packages have been developed for this. For native recognition of Unicode, the engines **luatex**(1) and **xetex**(1) are available, with corresponding formats.

Error handling

T_EX's error messages can be rather unfortunately cryptic. As general advice, sometimes only the first error matters, so it's worth trying to fix that first. Cutting down the input file can be helpful if mysteries remain. For more advice, including on a few specific problems: <https://texfaq.org/index#errors>

A convenient file in most distributions 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).

When T_EX reports an error, by default it puts you into an interactive dialog with a `?` prompt. Responding with another `?` shows you the available responses, for example, `x` quits immediately. The `e` response to T_EX'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 specifying emacs to be run can be set like this (in Bourne shell syntax):

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

To debug problems with finding files, see the **--kpathsea-debug** option below.

OPTIONS

TEX (in T_EX Live and Web2C) understands the following command line options. Most of these options are also supported by the derived engines.

Options can be started with either - or --; in this and other documentation, we use "-" and "--" interchangeably. Options can be separated from their values with either an = character or whitespace. Put filenames or other non-options that start with a dash after an option --. All the programs in the T_EX family (and many non-T_EX programs; this command-line parsing comes originally from the GNU getopt_long_only(3) library function) follow these conventions.

In T_EX and the other engines, all options must precede any non-option arguments, due to the peculiarities of T_EX's first-line parsing described above. For the T_EX utilities, options and non-options can be given in any order.

-help Print help message and exit.

-cnf-line *string*

Parse *string* as a *texmf.cnf* configuration line. See the Kpathsea manual.

-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>.

-[no-]file-line-error

Print error messages in the form *file:line:error* which is similar to the way many other compilers format them. By default, or with **-no-**, T_EX's default error format is used, usually starting with an ! character.

-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. (See above and below for more on formats.)

-halt-on-error

Exit with an error code when an error is encountered during processing.

-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. (See above.)

-kpathsea-debug *bitmask*

Sets path searching debugging flags according to the bitmask. See the *Kpathsea* manual for details: <https://tug.org/texinfohtml/kpathsea.html#Debugging>

-[no-]mktex *fnt*

Enable (disable) mktex *fnt*, where *fnt* must be *tex*, or *tfm*. See the *Kpathsea* manual: <https://tug.org/texinfohtml/kpathsea.html#mktex-scripts>

-mltex Enable ML_T_EX (Multi-lingual T_EX) extensions. Only effective in combination with **-ini**. More info: <https://tug.org/texinfohtml/web2c.html#MLTeX>

-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 the description below of the TEXMFOUTPUT environment variable.

-[no-]parse-first-line

Enable (or disable) checking the first line of the main input file for %& and, if so, parsing it to look for a format name or a **-translate-file** option.

-progrname *name*

Pretend to be program *name* for purposes of *Kpathsea* lookups. This affects both the format used and the search paths.

-recorder

Enable the filename recorder. This outputs a machine-readable trace of all files opened for input and output in a file with the usual jobname and extension *.fls*.

-[no-]shell-escape

Allow (or disallow) documents to execute arbitrary shell commands via the `\write18{command}` construct. This feature is normally restricted for security reasons to a very few known-safe programs. You should use this option, which enables execution of any command at all, only for trusted documents. More on shell escapes: <https://tug.org/texinfohtml/web2c.html#Shell-escapes>

-src-specials

Insert source specials into the dvi file. This is used for **synctex**(1). In editors that support SyncT_EX, you can click in the output window and get back to the corresponding location in the source.

-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 translation table in the file *tcxname* 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 override this setting.

-8bit Make all characters printable by default.**-version**

Print version information and exit.

ENVIRONMENT

Numerous environment variables can be used to find files. A few are mentioned here, but see the Kpathsea(rch) library documentation (e.g., <https://tug.org/texinfohtml/kpathsea.html#TeX-support>) for a more comprehensive list, and how they are used. The **kpsewhich**(1) utility can be used to query the values of the variables. In Kpathsea specifications, `"/"` means to search subdirectories recursively.

Every variable setting in *texmf.cnf* can be overridden by the environment variable of the same (or associated) name, as explained tersely at the top of *texmf.cnf*, and more completely in the Kpathsea documentation.

TEXMFOUTPUT

Normally, T_EX 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*, T_EX attempts to create */tmp/paper.log* (and */tmp/paper.dvi*, if any output is produced.) TEXMFOUTPUT is also checked for input files, as T_EX often generates files that need to be subsequently read; for input, no suffixes (such as ".tex") are added; the input name is simply checked as given.

If the **-output-directory** option is given, T_EX sets TEXMFOUTPUT in the environment to its value, so that programs invoked by T_EX have it available.

More on output file location: <https://tug.org/texinfohtml/web2c.html#Output-file-location>

TEXINPUTS

Search path for `\input` and `\openin` files. By default, this starts with `.` (a period, meaning the current directory as usual), 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.

TEXMFDOTDIR

In T_EX Live's *texmf.cnf*, TEXINPUTS and other paths don't start literally with `"/`, but rather with `$TEXMFDOTDIR`, which in turn defaults to `"/`. This indirection makes it easy to, for example, search the current directory and all its subdirectories for everything: set TEXMFDOTDIR to `"/`.

TEXFORMATS

Search path for format files (*.fmt*). They are typically stored under the TEXMFSYSVAR directory. You can find the exact location of *tex.fmt* (for example) with this **kpsewhich** invocation:

`kpsewhich -engine=/ -all tex.fmt`

TEXPOOL

Search path for T_EX's internal strings (*.pool*); this is only relevant for original T_EX; the extended engines all have the strings compiled into the binary.

TFMFORMATS

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

TEXEDIT

Command template for switching to editor (see above). The default, usually **vi**, is set when T_EX is compiled.

FILES

The locations 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 T_EX's internal strings.

**.tfm* Metric files for T_EX's fonts.

**.fmt* Predigested T_EX format (*.fmt*) files.

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

The basic macro package described in *The T_EXbook*.

\$TEXMFMAIN/tex/plain/config/tex.ini

The driver file that builds the plain format file, *tex.fmt*, in T_EX Live.

BUGS

This version of T_EX implements a number of extensions, which technically conflict with the definition of original T_EX (although almost never make a difference in practice). 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 program may crash, and/or the generated DVI file will be invalid. Patches to catch such overflow cases would be welcome.

Knuth still accepts new bug reports for T_EX, but only reviews them every several years. They are vetted in advance by a small group of experienced volunteers. For a list of submitted reports and other information: <https://tug.org/texmfbug>.

Naturally, bugs in engines other than original T_EX should be directed to their respective mailing lists and maintainers.

SEE ALSO

mf(1), **pdfTeX(1)**, **latex(1)**, **luatex(1)**, and plenty more.

Donald E. Knuth, *The T_EXbook*, Addison-Wesley.

Leslie Lamport, *L^AT_EX: A Document Preparation System*, Addison-Wesley.

Some free documentation on the T_EX language:

Petr Olsak, T_EX in a Nutshell: <https://ctan.org/pkg/tex-nutshell>

Plain T_EX documentation topic on CTAN: <https://ctan.org/topic/tut-plaintex>

Victor Eijkhout, *T_EX By Topic*, A T_EXnician's reference: <https://www.eijkhout.net/tex/tex-by-topic.html>

Paul W. Abrahams, Kathryn Hargreaves, Karl Berry, *T_EX for the Impatient*, on T_EX, plain T_EX, and **plain**(1); available in French and Chinese translations: <https://ctan.org/pkg/impatient>

Some online information:

Getting Started with TeX, LaTeX, and friends, a short web page with selected resources:

<https://tug.org/begin>

David Bausum, T_EX primitive control sequences: <https://tug.org/utilities/plain/cseq.html>

Levels of T_EX, briefly describing the various engines and formats: <https://tug.org/levels>

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>

If you find T_EX useful, please consider joining or supporting the T_EX Users Group, or another user group in your area:

<https://tug.org>

<https://tug.org/usergroups.html>

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 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.

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

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

Public discussion list for T_EX Live: <https://lists.tug.org/tex-live>

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