

**NAME**

texfot – run TeX, filtering online transcript for interesting messages

**SYNOPSIS**

texfot [*option*]... *texcmd* [*texarg*...]

**DESCRIPTION**

texfot invokes *texcmd* with the given *texarg* arguments, filtering the online output for “interesting” messages. Its exit value is that of *texcmd*. Examples:

```
# Basic invocation:
texfot pdflatex file.tex

# Ordinarily the output is copied to /tmp/fot before filtering;
# omit that:
texfot --tee=/dev/null file.tex

# Example with more complex engine invocation:
texfot lualatex --recorder '\nonstopmode\input file'
```

Aside from its own options, described below, texfot just runs the given command with the given arguments (same approach to command line syntax as nice, time, timeout, etc.). Thus, texfot works with any engine and any command line options.

texfot does not look at the log file or any other possible output file(s); it only looks at standard output from the command.

The messages shown are intended to be those which probably deserve some action by the author: error messages, overfull and underfull boxes, undefined citations, missing characters from fonts, etc.

**FLOW OF OPERATION**

Here is the order in which lines of output are checked:

1. If the “next line” needs to be printed (see below), print it.
2. Otherwise, if the line matches the built-in list of regexps to ignore, or any user-supplied list of regexps to ignore (given with `--ignore`, see below), in that order, ignore it.
3. Otherwise, if the line matches the list of regexps for which the next line (two lines in all) should be shown, set the “next line” flag for the next time around the loop and show this line. Examples are the usual ! and `filename:lineno:` error messages, which are generally followed by a line with specific detail about the error.
4. Otherwise, if the line matches the list of regexps to show, show it.
5. Otherwise, ignore this line.

Once a particular check matches, the program moves on process the next line.

Don’t hesitate to peruse the source to the script, which is essentially a straightforward loop matching against the different regexp lists as above. You can see all the messages in the different categories in the source.

Incidentally, although nothing in this basic operation is specific to TeX engines, all the regular expressions included in the program are specific to TeX. So in practice the program isn’t useful except with TeX engines, although it would be easy enough to adapt it (if there was anything else

as verbose as TeX to make that useful).

## OPTIONS

The following are the options to `texfot` itself (not the TeX engine being invoked; consult the TeX documentation or the engine's `--help` output for that). The first non-option terminates `texfot`'s option parsing, and the remainder of the command line is invoked as the TeX command, without further parsing. For example, `texfot --debug tex --debug` will output debugging information from both `texfot` and `tex`.

Options may start with either `-` or `--`, and may be unambiguously abbreviated. It is best to use the full option name in scripts, though, to avoid possible collisions with new options in the future.

`--debug`

`--no-debug`

Output (or not) what is being done on standard error. Off by default.

`--ignore regexp`

Ignore lines in the TeX output matching (Perl) *regexp*. Can be repeated. Adds to the default set of ignore regexps rather than replacing. These regexps are not automatically anchored (or otherwise altered), just used as-is.

`--interactive`

`--no-interactive`

By default, standard input to the TeX process is closed so that TeX's interactive mode (waiting for input upon error, the `*` prompt, etc.) never happens. Giving `--interactive` allows interaction to happen.

`--quiet`

`--no-quiet`

By default, the TeX command being invoked is reported on standard output. `--quiet` omits that.

`--tee file`

By default, the output being filtered is tee-ed, before filtering, to `$TMPDIR/fot` (`/tmp/fot` if `TMPDIR` is not set), to make it easy to check the full output when the filtering seems suspect. This option allows specifying a different file. Use `--tee /dev/null` if you don't want the original output at all.

`--version`

Output version information and exit successfully.

`--help`

Display this help and exit successfully.

## RATIONALE

I wrote this because, in my work as a TUGboat editor (<<http://tug.org/TUGboat>>, submissions welcome), I end up running and rerunning many papers, many times. It was too easy to lose warnings I needed to see in the mass of unvarying and uninteresting output from TeX, such as all the style files being read and all the fonts being used. I wanted to see all and only those messages which actually needed some action by me.

I found some other programs of a similar nature, the `silence` LaTeX package, and plenty of other (La)TeX wrappers, but it seemed none of them did what I wanted. Either they read the log file (I wanted the online output only), or they output more or less than I wanted, or they required invoking TeX differently (I wanted to keep my build process exactly the same, definitely

including the TeX invocation, which can get complicated). Hence I finally went ahead and wrote this.

Here are some keywords if you want to explore other options: `texloganalyser`, `pydflatex`, `logfilter`, `latexmk`, `rubber`, `arara`, and searching for `log` at <http://ctan.org/search>.

`texfot` is written in Perl, and runs on Unix, and does not work on Windows. (If by some chance anyone wants to run this on Windows, please make your own fork; I'm not interested in supporting it.)

The name comes from the `trip.fot` and `trap.fot` files that are part of Knuth's trip and trap torture tests, which record the online output from the programs. I am not sure what "fot" stands for in trip and trap, but I can pretend that it stands for "filter online transcript" in the present case :).

## **AUTHORS AND COPYRIGHT**

This script and its documentation were written by Karl Berry and both are released to the public domain. Email [karl@freefriends.org](mailto:karl@freefriends.org) with bug reports. There is no home page beyond the package on CTAN: <http://www.ctan.org/pkg/texfot>.