texdoc

Find & view documentation in TEX Live

http://tug.org/texdoc/

Manuel Pégourié-Gonnard

v0.60rc 2009-11-18

1 Quick guide

1.1 Basics

Open a command line[1] and type `texdoc` ⟨*name*⟩: the documentation of the ⟨*name*⟩ package will pop up. Of course, replace ⟨*name*⟩ with the actual name of the package you want to learn about. You can also look for the documentation of more than one package at once: just give many names as arguments.

The rest of this section describes the most usual options, like how to see all documentation related to a package or use a different viewer.

1.2 Modes

texdoc has different modes that determine how results will be handled. The default mode, called "view" mode, is to open the first (supposedly the best) result in a viewer. It is rather handy when you know what you want to read, and want to access it quickly. On the other hand, there may be other relevant documents for the given ⟨*name*⟩, which are ignored in view mode.

The so-called "list mode" makes texdoc list all relevant documentation and ask you which one you want to view. It is useful when there a other interesting sources of information besides the package's main documentation.

There is also a "mixed" mode, intended to combine the best of view mode and list mode: if there is only one relevant result, then texdoc opens it in a viewer, else it offers you a menu.

Usually, texdoc shows you only results it considers relevant. If there are no "good" results, it falls back to less relevant results. You can force texdoc to show you also "bad"

[1] On windows, use "Execute" from the Start menu and type `cmd`. On Mac OS X, use the "terminal" icon on Mac OS X. If you are using another flavour of Unix, you probably know what to do.

results even when there are good ones by using the “showall” mode. (This implies using a menu rather than starting a viewer.)

You can select with command-line options: use `texdoc` ⟨*option*⟩ ⟨*name*⟩ with one of the following options: `-w` or `--view` for view mode, `-m` or `--mixed` for mixed mode, `-l` or `--list` for list mode, `-s` or `--showall` for showall mode.

If you always (or mostly) use the same mode, it is probably easier to select it in a configuration file than to always use the command-line option. For this, see the next section.

1.3 Configuration files

texdoc use various configuration files, which you can see using the `-f` or `--files` options. The second entry in this list is marked with a star `(*)`: it is the file you should use for your personal preferences as a user; you may need to create it (and the parents directories).

In order to select you favorite mode, just insert a line `mode =` ⟨*yourmode*⟩ in this files, where ⟨*yourmode*⟩ is one of `view`, `mixed`, `list` or `showall`.

The configuration files can be used to tweak texdoc in many ways, the most useful of which is probably the selection of the viewers for various types of documents, explained in the next section.

1.4 Viewers

texdoc’s mechanism for choosing a viewer varies according to your platform. On Windows and MacOS, it uses your file associations like when you double-click files in the Explorer or the Finder (except for the text viewer, which is always a pager). On Unix, it tries to find a viewer in the path from a list of “known’ viewers.

You may want to select a different viewer for some kind of file. This is achieved by the various `viewer_`⟨*ext*⟩ configuration options, where ⟨*ext*⟩ is the extension corresponding to the file type. For example, if you want to set xpdf as your default PDF viewer, and run it in the background, insert the line `viewer_pdf = xpdf %s &` in your configuration file. Here, `%s` stands for the name of the file to view.

1.5 Conclusion

We have now covered the most common needs. The next part explains how texdoc proceeds to find the best results. The default configuration file tries hard to set appropriate values so that you have a good out-of-the-box experience, but you may want to understand the underlying mechanisms and adapt them to your needs. The final part is a full reference for configuration options, including points omitted in the present part.

Your feedback is very welcome of the texdoc mailing list. Feel free to post comments, bug reports, suggestions for improvements (inc. new aliases), even without subscribing.

2 File search, aliases, score

2.1 An overview of how texdoc works

When you type `texdoc` ⟨*keyword*⟩, texdoc browses the tree containing documentation (given by the kpathsea variable `TEXDOCS`), lists all files containing ⟨*keyword*⟩ in their name (including the directory name) and give them a score based on some simple heuristics. For example, a file named ⟨*keyword*⟩`.pdf`, will get a high score, ⟨*keyword*⟩`-doc` will be preferred over ⟨*keyword*⟩`whatever`, files in a directory named exactly ⟨*keyword*⟩ get a bonus, etc.

There is also some filtering based on extensions: only files with known extensions are listed, and some extensions get a lesser score. Also, there is some score adjustments based on keywords; by default, Makefile's get a very bad score since they are almost never documentation.[2]

Then, depending on the mode, the file with the highest score is opened in a viewer, or the list of results is shown. Usually, only results with a positive score are displayed, except in showall mode. Result with very bad score (-100 and below) are never displayed.

However, this model works only if the documentation for ⟨*keyword*⟩ has ⟨*keyword*⟩ in its name, which is not always true. The documentation of the memoir class is in `memman.pdf`, for example, but it will be found anyway since it is in a directory named memoir. But there are more complicated cases: the documentation for mathptmx is in `psnfss2e.pdf`, which is quite more complicated to guess.[3]

Here comes the notion of *alias*: in the default configuration file, mathptmx is aliased to psnfss2e, so that when you type `texdoc mathptmx`, texdoc knows it has to look also for psnfss2e. Note that texdoc will also look for the original name, and that a name can be aliased to more than one new name (this is new in texdoc 0.60).

We will soon see how you can configure this, but let's give a few definitions about how a file can match keyword first (all matching is case-insensitive):

1. The keyword is a substring of the file name.
2. The keyword is a "subword" of the file name; words are defined as a sequence of alphanumeric characters delimited by punctuation characters (there is no space in file names in TeX Live) and a subword is a substring both ends of which are a word boundary.
3. The keyword can match "exactly" the file name: that is, the file name is the keyword + and extension.

[2] They often end up in the doc tree, since the source of documentation is often in the same directory as the documentation itself in TeX Live. Other source files are discriminated by extension.

[3] But not totally impossible: using the TeX Live database, texdoc can know which documentation files are associated with a given `.sty` file. Here, it could guess that the documentation is either `psfonts.pdf` of `psnfss2e.pdf`. This will probably be done in a future version.

2.2 Alias directives

`alias` ⟨*original keyword*⟩ `=` ⟨*name*⟩

You can define your own aliases in texdoc's configuration files (see 1.3 or 3.1). For example, insert

```
alias mathptmx = psnfss2e
```

in order to alias mathptmx to psnfss2e. Precisely, it means that files matching exactly psnfss2e will be added to the result list when you look for mathptmx, and get a score of 10 (default score for alias results). This is greater than the results of heuristic scoring: it means that results found via aliases will always rank before results associated to the original keyword. Please note than aliasing is case-insensitive, and the aliases don't cascade: only aliases associated to the original keyword are used.

You can have a look at the configuration file provided (the last shown by `texdoc -f`) for examples. If you feel one of the aliases you defined locally should be added to the default configuration, please share it on the texdoc mailing list.

If, for some reason, you want to prevent aliases from begin used for one session of texdoc, you can use the `-A` or `--noalias` command-line option (see 3.2.5).

2.3 Score directives

`setscore` ⟨*pattern*⟩ `=` ⟨*final score*⟩
`adjscore` ⟨*pattern*⟩ `=` ⟨*score adjustment*⟩

There are two ways of changing the score of a result: you can either set it to a fixed score, or adjust it by some quantity. This is done in the configuration file (1.3 or 3.1) using the directives `setscore` and `adjustscore`. They only affect results for which ⟨*pattern*⟩ matches as a subword.

```
setscore /Makefile = -100
adjscore /tex-virtual-academy-pl/ = -10
```

In the above example (from the default configuration file), all files named `Makefile` are killed (also files names `Makefile-foo` if there are any): since the get a score of exactly -100, they will never be displayed. Files from the `tex-virtual-academy-pl` directory, on the other hand, are not killed but just get a malus, since they are a common source of "fake" matches which hide better results (even for the lucky ones who can read polish).

2.4 File extensions

The allowed file extensions are defined by the configuration item `ext_list` (default: pdf, html, txt, ps, dvi, no extension). You can configure with a line `ext_list =` ⟨*your, list*⟩ in a configuration file. Be aware that it will completely override the default list, not add to

it. An empty string in the list means files without extension (no dot in the name), while a star means any extension.

For scoring purposes, there is also a `badext_list` parameter: files whose extension is "bad" according to this list will get a lesser score (currently 0). This only affect heuristic scoring (results found from the original keyword, not from aliases).

3 Full reference

3.1 Precedence of configuration sources

Values for a particular setting can come from several sources. They are treated in the following order, where first value found is always used:

1. Command-line options.
2. Environment variables ending with `_texdoc`.
3. Other environment variables.
4. Values from configuration files, read in the following order:
   a) `TEXMFHOME/texdoc/texdoc-`⟨*platform*⟩`.cnf`
   b) `TEXMFHOME/texdoc/texdoc.cnf`
   c) `TEXMFHOME/texdoc/texdoc-dist.cnf`
   d) `TEXMFLOCAL/texdoc/texdoc-`⟨*platform*⟩`.cnf`
   e) `TEXMFLOCAL/texdoc/texdoc.cnf`
   f) `TEXMFMAIN/texdoc/texdoc.cnf`
5. Hard-coded defaults that may depend on the current platform or the current value of another setting.

For the configuration files, ⟨*platform*⟩ stands for the name of the current platform, with names matching those of the directories in `TEXLIVEROOT/bin`, and `TEXMFHOME` and others are the kpse's values, see the kpathsea and web2c manuals. The name with ⟨*platform*⟩ can be used on installation shared between many machines where, for example, not the same viewers are available. However, their use is not recommended in other situations. Finally, the special file `texdoc-dist.cnf` allows you to install a newer version of texdoc (including its default config file) in your home: see the web page for instructions on running texdoc from the SVN.

3.2 Command-line options

All command-line options (except the first three below) correspond to configuration item that can be set in the configuration files: we refer the reader to the corresponding section for the meaning of this configuration item.

3.2.1 `-h, --help` — Shows a quick help message (namely a list of command-line options) and exits successfully.

3.2.2 `-V, --version` — Show the current version of the program and exits successfully.

3.2.3 `-f, --files` — Shows the list of the configuration files for the current installation and platform, with their status (active, not found, or disabled (see 3.4.11)) and exits successfully.

3.2.4 `-w, --view, -l, --list, -m, --mixed, -s, --showall, -r, --regex` — Set `mode` to the given value, see 3.4.2.

3.2.5 `-a, --alias, -A, --noalias` — Set `alias_switch` true (resp. false), see 3.4.4

3.2.6 `-i, --interact, -I, --nointeract` — Set `interact_switch` to true (resp. false), see 3.4.3.

3.2.7 `-e=`⟨*l*⟩`, --extensions=`⟨*l*⟩ — Set `ext_list`, see 3.4.5. *But* be aware that on the command line there should be no space at all, neither in the list (unless quoted according to you shell's convention) not between the `-e` or `--extension` option, the equal sign, and the list. Also take care to quote the special value `*` if necessary. The equal sign is optional.

3.2.8 `-v=`⟨*n*⟩`, --verbosity=`⟨*n*⟩ — Set `verbosity_level` to ⟨*n*⟩, see 3.4.8. Be aware that you must avoid spaces on the command line, and the `=` sign is optional.

3.2.9 `-d, -d=`⟨*list*⟩`, --debug, --debug=`⟨*list*⟩ — Set `debug_list`, see 3.4.9. If not list is given, activates all available debug items.

3.2.10 `-M, --machine` — Set `machine_switch` to true, see 3.4.10.

3.3 Environment variables

They all correspond to some `viewer_`⟨*ext*⟩ setting, and the reader is referred to 3.4.7 for details. Also, environment variables used by older versions of texdoc are accepted. You can append `_texdoc` to every name in the first column: this wins over every other name.

New name	Old name 1	Old name 2	Config. item
`PAGER`	`TEXDOCVIEW_txt`	`TEXDOC_VIEWER_TXT`	`viewer_txt`
`BROWSER`	`TEXDOCVIEW_html`	`TEXDOC_VIEWER_HTML`	`viewer_html`
`DVIVIEWER`	`TEXDOCVIEW_dvi`	`TEXDOC_VIEWER_DVI`	`viewer_dvi`
`PSVIEWER`	`TEXDOCVIEW_ps`	`TEXDOC_VIEWER_PS`	`viewer_ps`
`PDFVIEWER`	`TEXDOCVIEW_pdf`	`TEXDOC_VIEWER_PDF`	`viewer_pdf`

3.4 Configuration items

3.4.1 *Structure of configuration files.* — Configuration files are line-oriented text files. Comments begin with a `#` and run to the end of line. Lines containing only space are ignored. Space at the beginning or end of a line, as well as around an `=` sign, is ignored. Apart from comments and empty lines, each line must be of one of the following forms.

⟨*configuration item*⟩ = ⟨*value*⟩
alias ⟨*original keyword*⟩ = ⟨*name*⟩
setscore ⟨*pattern*⟩ = ⟨*final score*⟩
adjscore ⟨*pattern*⟩ = ⟨*score adjustment*⟩

We will concentrate on the «configuration items» here, other directives have already be presented (2.2 and 2.3).

In the above, ⟨*value*⟩ is usually a free string, but of course not every ⟨*value*⟩ is valid for every ⟨*configuration item*⟩ and nothing in it need not be quoted: quotes would be interpreted as part of the value, not as quotation marks (this also holds for the other directives).

Lines which do not obey these rules raise a warning, as well as unrecognised values of ⟨*configuration item*⟩ raise no warning at the moment.

The ⟨*value*⟩ is usually interpreted as a string, except when ⟨*configuration item*⟩ ends with:

1. `_list`, then ⟨*value*⟩ is a coma-separated list of strings. Space around commas is ignored. Two consecutive comas or a coma at the beginning or end of the list means the empty string at the corresponding place.
2. `_switch`, then ⟨*value*⟩ must be either `true` or `false` (lowercase).
3. `_level`, then ⟨*value*⟩ is an integer.

3.4.2 `mode` = ⟨*view, list, mixed, showall, regex*⟩ — Set the mode to the given value. Default is `view`. All mode except regex have been presented in 1.2.

In regex mode, the given name is interpreted as a Lua regex, and only files whose full name (including path) matches it are found. Alias and score mechanisms are disabled. **Warning**: this mode may disappear in a future version of texdoc. If you want me to keep it, please tell me so!

3.4.3 `interact_switch` = ⟨*true, false*⟩ — Turn on or off interaction. Default is on. Turning interaction off prevents texdoc to ask you to choose a file to view when there are multiple choices, so it just prints the list of files found.

3.4.4 `alias_switch` = ⟨*true, false*⟩ — Turn on or off aliasing. Default is on.

3.4.5 `ext_list` = ⟨*list*⟩ — Set the list of recognised extensions to ⟨*list*⟩. Default is

```
pdf, html, txt, dvi, ps,
```

This list is used to filter and sort the results that have the same score(with the default value: pdf first, etc). Two special values are recognised:

- *The empty element*. This means files without extensions, or more precisely without a dot in their name. This is meant for files like `README`, etc. The file is assumed to be plain text for viewing purpose.
- `*` means any extension. Of course if it is present in the list, it can be the only element!

There is a very special case: if the searched ⟨*name*⟩ has `.sty` extension, texdoc enters a special search mode for `.sty` files (not located in the same place as real doc-

umentation files) for this ⟨*name*⟩, indepandantly of the current value of `ext_list` and `mode` (unless it is the `regex` mode). In an ideal world, this wouldn't be necessary since every sty file would have a proper documentation in pdf, html or plain text, but...

For each ⟨*ext*⟩ in `ext_list` there should be a corresponding `viewer_`⟨*ext*⟩ value set. Defaults are defined corresponding to the default `ext_list`, but you can add values if you want. For example, if you want texdoc to be able to find man pages and display them with the `man` command, you can use

```
ext_list = pdf, html, 1, 5, txt, dvi, ps,
viewer_1 = man
viewer_5 = man
```

3.4.6 `badext_list =` ⟨*list*⟩ — Set the list of "bad" extensions to ⟨*list*⟩. Default is

```
txt,
```

Files with those extensions get a heuristic score of 0.

3.4.7 `viewer_`⟨*ext*⟩ `=` ⟨*cmd*⟩ — Set the viewer command for files with extension ⟨*ext*⟩ to ⟨*cmd*⟩. For files without extension, `viewer_txt` is used, and there's no `viewer_` variable. In ⟨*cmd*⟩, `%s` can be used as a placeholder for the file name, which is otherwise inserted at the end of the command. The command can be a arbitrary shell construct.

3.4.8 `verbosity_level =` ⟨*n*⟩ — Set the verbosity level to ⟨*n*⟩. 3 means errors, warnings and informational message will be printed (on stdout); 2 means only errors and warnings, 1 only errors and 0 nothing (not recommended).

3.4.9 `debug_list =` ⟨*list*⟩ — Set the list of activated debug items (default: none). For a list of available items, see the source code, file `constants.tlu`, variable `known_debugs`.

3.4.10 `machine_switch =` ⟨*true, false*⟩ — Turn on or off machine-readable output (default: off). In machine-readable mode, each line of output is

⟨*argument*⟩`\t`⟨*score*⟩`\t`⟨*filename*⟩

where ⟨*argument*⟩ is the name of the argument to which the results correspond (mainly useful if there were many arguments), `\t` is the tab (ascii 9) character, and the other entries are pretty self-explanatory. Nothing else is printed on stdout, except if a internal error occurs (in which case exit code will be 1). In the future, more tab-separated fields may be added at the end of the line, but the first 3 fields will remain unchanged.

3.4.11 `lastfile_switch =` ⟨*true, false*⟩ — If set to true, prevents texdoc from reading any other configuration file after this one (they will be reported as "disabled" by `texdoc -f`). Mainly useful for installing a newer version of texdoc in your home and preventing the default configuration file from older versions to be used(see the web site for instructions on how to do so).

3.5 Exit codes

The current exit code are as follow:

0. Success.
1. Syntax error.
2. Documentation not found for at least one argument.

4 Licence

Happy TEXing!