What's a configuration file, and where should it go?

Frank Küster

February 2007

There are two questions with respect to configuration files which have been raised over and over in Debian TeX packaging: 1. Which files should be treated as configuration files and kept in the hierarchy below `/etc/texmf`? 2. Should the TeX packages make use of the `TEXMFSYSCONFIG` tree? This document tries to summarizes the arguments that have been given in earlier discussions, in the hope that it will help us to make a good decision for lenny, and serve as a reference in the future when these issues come up again.

Contents

1. Which files should be treated as a configuration file?

1.1. General rules in Debian

The Debian policy defines configuration files in 10.7.1:

> configuration file
> A file that affects the operation of a program, or provides site- or host-specific information, or otherwise customizes the behavior of a program. Typically, configuration files are intended to be modified by the system administrator (if needed or desired) to conform to local policy or to provide more useful site-specific behavior.

This definition is not as clear-cut as it may seem. Two aspects that are worth remembering is that it talks about customizing *programs*, and about host- or *site*-specific behavior.

1.2. Document-specific vs. site-specific settings

The purpose of a TeX system is to typeset documents. Although a machine may be set up to generate many version of a particular type of document (e.g. timetables, product lists etc. from a database), in general it is not specified which type of documents are typeset in terms of content, layout and output format.

Therefore we regard all aspects of customization that affect the document layout, autogenerated content and output format as **document-specific**. Files which allow to change these things are not to be treated as configuration files. If a particular site only creates documents of a certain type at some point in time, this is a special case which doesn't invalidate the rule.

On the other hand, customization that affects the behavior of programs (search paths, available fonts, preferred font formats, etc.) is regarded as belonging to a configuratio file.

1.3. A survey of candidate files in TeXLive

In order to get an overview about the number of files affected, I conducted an overview over the files that are possible candidates for configuration files.

1.3.1. Configuration files in current packages

tex-common tex-common currently installs the configuration files `/etc/texmf/fmt.d/00tex.cnf`, `/etc/texmf/language.d/00tex.cnf`,

`/etc/texmf/web2c/mktex.cnf`, the empty directory `/etc/texmf/updmap.d`, and handles `/etc/texmf/texmf.cnf` and `/etc/texmf/texmf.d/*` (with `ucf`). No debate here, I guess; `mktex.cnf` affects the behavior of the font generation programs, namely the location of the font cache, and without doubt contains a site-wide, document-independent setting.

Sum: 9 files, but only 2 in kpathsea-searched directories.

teTEX tetex-base installs `/etc/texdoctk/texdoctk.dat` and configuration snippets in `updmap.d` and `language.d`. `/etc/texdoctk/` is accessible via a symlink in `$TEXMFDIST`, and contains also the `ucf`-managed `texdocrc`. However, `texdoctk.dat` is not in fact a configuration file, it depends on the contents of the `doc` hierarchies in the `TEXMF` trees, and should either be installed in `TEXMFDIST` or even be generated.

tetex-bin brings in an ucf-managed file in `fmt.d` plus `xdvi.cfg`, the latter handled with a symlink.

tetex-extra only adds a file to `updmap.d`[1]

Sum: 2 files in kpathsea-searched dirs, 4 other

texlive The texlive packages install files into `language.d`, `updmap.d`, and `fmt.d`. Additionally, texlive-base-bin installs files into `/etc/texmf/texlive`. These are accessed from `TEXMFDIST` via symlinks and comprise `dvipdfm.cfg`, `texdocrc.defaults`, `xdvi.cfg`, and the following files in `/etc/texmf/texlive/dvips`:

config.bakoma	config.ibmvga	config.ot2	alt-rule.pro	epson.cfg
config.canonex	config.ljfour	config.ps	canonex.cfg	ibmvga.cfg
config.cms	config.luc	config.qms	cx.cfg	ljfour.cfg
config.cx	config.mbn	config.toshiba	deskjet.cfg	qms.cfg
config.deskjet	config.mga	config.unms	dfaxhigh.cfg	toshiba.cfg
config.dvired	config.mirrorprint	config.xyp	dvired.cfg	
config.epson				

I'm not sure about the purpose or use of the `*.cfg` files.

Sum: 34 files outside the `foo.d` directories.

1.3.2. Candidate configuration files

Discarded candidates There are loads of TEX input files that have an extension `.cfg`. Most of them are not intended to be changed at all, not even

[1] The files with extension `md5sum` in `/usr/share/tetex-extra` are a leftover and not actually used for `ucf` handling.

for document-specific customization. As an example, take `listings.cfg`: The documentation says:

> Never modify a file from the listings package, in particular not the configuration file. Each new installation or new version overwrites it. The software license allows modification, but I can't recommend it. It's better to create one or more of the files
>
> | | | |
> |---|---|---|
> | `lstmisc0.sty` | for | local add-ons (see developer's guide), |
> | `lstlang0.sty` | for | local language definitions (see 4.17), and |
> | `lstlocal.cfg` | as | local configuration file |
>
> and put it/them to the other listings files. These three files are not touched by a new installation except you remove them. If `lstlocal.cfg` exists, it is loaded after `listings.cfg`. You might want to change one of the following parameters.

Those files that *are* intended for configuration do in most cases affect document-specific settings, and do therefore not qualify as configuration files in the Debian sense.

Possibly valid candidates

TeX input files used at documentation creation time There are some files in the teTEX and TEXLive distributions that have been changed by the distribution developer(s) compared to the versions on CTAN. The most well-known example is `hyperref.cfg`. It could be argued that if it makes sense to make a distribution-wide decision to change this file, a site-specific change by the local admin is just as reasonable.

`hyperref.cfg` is special because it contains code to detect the engine and output format that is used, in other words: It is related to behavior of the programs on the system. There is no one-fits-all default, and it may choose a setting that doesn't fit the local user's or admin's wishes. Therefore I think it qualifies as a Debian configuration file. The same is true for `geometry.cfg`, and maybe others.

TeX input files used for format creation While it is legally possible to modify LATEX components like `texsys.cfg`, or customize `fonttext.ltx` as `fonttext.cfg`, I still do not think that we should make them configuration files.

First of all, modifying `texsys.cfg` doesn't make sense on a Debian system, the path searching works well. Second, changing the font setup (or hyphenation

with `hyphen.cfg` instead of the files in `language.d`, shiver...) leads to an unsupported "LaTeX" format and should be very much discouraged; and for anyone who is able to do it, it's easy to create a new renamed format instead.

On the other hand, files like `pdftexconfig.tex` that are read by a particular engine are a different thing. `pdftexconfig.tex` contains the settings for the PDF paper size (and the default compression) and to me seems to be a configuration file for pdfTeX. Furthermore, it is handled by `texconfig-sys`.

1.4. A proposal: Listening to upstream

I propose to treat the following two types of files as configuration files:

1. Files that are modified by upstream's configuration scripts
2. TeX input files that are marked by upstream as being intended for configuration *and* that try to tweak the document creation depending on the detected engine or output format

1.4.1. Upstream configuration files

With the files that are handled by `texconfig-sys` and other scripts, we hardly have any choice, because these tools will create copies in TEXMFSYSCONFIG, anyway, if there isn't already one – except if we would decide to no longer support TEXMFSYSCONFIG, and patch the scripts accordingly.

Because of our setup with `{fmt,updmap,language}.d`, files normally handled by `fmtutil-sys` and `updmap-sys` are already covered, and we only need to care about `texconfig-sys`. It is able to handle the following files:

```
TEXMF/tex/generic/config/pdftexconfig.tex
TEXMF/web2c/mktex.cnf
TEXMF/dvipdfm/config/config
TEXMF/dvips/config/config.ps
TEXMF/xdvi/XDvi
TEXMF/web2c/texmf.cnf
TEXMF/dvips//config.<printer>
TEXMF/metafont/misc/modes.mf
```

1.4.2. Output format/engine-dependent files

Files that interact with the engine/output format at typesetting time have a problem: There is no one-fits-all default for all of them, and they may choose

the wrong one. This *can* be circumvented by including driver settings in the document, but this reduces "portability" in the sense that it is hard to write documents that can be processed to different formats and with different engines without changing the source.

However, instead of making all those files configuration files, we can also consider an alternative approach: All of them could read a central configuration file in which the site admin can specify the defaults they prefer, in particular for DVI[2]. We should ask Heiko Oberdiek and Martin Schröder on his opinion about this.

`grep`ping through `Master/texmf{,-dist}/tex` in TEXLive 2007 yields a lot of files which contain the string `pdfoutput`. Among these, I assume that only files in `generic` and `latex` are interesting (the others are `*.ini` files or ConTeXt files). Furthermore, in most cases the current usage of `\pdfoutput` is probably a bug, anyway, in particular in not-so-well maintained specialized LATEX classes.

Here's a list of upstream-declared configuration files which interact with the system/select drivers:

```
latex/pict2e/pict2e.cfg
latex/contour/contour.cfg
texmf/tex/latex/config/color.cfg
texmf/tex/latex/config/graphics.cfg
texmf/tex/latex/config/hyperref.cfg
```

These files use `\pdfoutput` in an attempt to detect the driver/output format and could use our central configuration file:

```
latex/crop/crop.sty
latex/geometry/geometry.sty
```

(Actually, there was a geometry.cfg in TEXLive 2005, but it was removed because it contained only autogenerated comments).

1.4.3. Consequences

Including possible configuration files for `crop.sty` and `geometry.sty`, this would give 7 additional configuration files, compared to 43 in a current TEXLive system – not a large increase. The list in the appendix contains 10 more packages that could possibly profit from configurable driver detections. However, since this requires upstream changes, I think we will only include these if there's going to be a central "divps or dvipdfm" configuration file.

[2] Does XeTEX introduce additional options here?

2. Should we use the TEXMFSYSCONFIG tree?

This question can be divided in two parts:

1. Should tex-common provide support for the TEXMFSYSCONFIG tree?
2. Should configuration files of the Basic TEX Packages, or of any TEX package, be installed in this tree? Or instead, should they be installed in a subdirectory /etc/texmf/<package>, with symlinks from the TEXMFDIST or TEXMFMAIN trees pointing to them?

The first questions seems to be settled and undisputed, the answer is "yes". The second one is currently under debate (see http://bugs.debian.org/403026). However, the reasons given in earlier discussions for supporting TEXMFSYSCONFIG also affect the decision about using it.

2.1. Reasons for supporting TEXMFSYSCONFIG

- no patching of upstream scripts needed
- less user-visible differences to upstream
- why use complicated ways to achieve what upstream already supports (namely, configuration files in /etc)?
- We do it in etch, changing that would be a pain

2.2. Arguments for per-package directories and symlinks

- No moving around necessary

2.3. Arguments for installing conffiles in TEXMFSYSCONFIG

- It's the logical, simple solution and uses upstream's provided facilities for conffiles
- If users install files in their "ordinary" locations below TEXMFSYSCONFIG, these will override the copies in /etc/texmf/texlive, although these are also in TEXMFSYSCONFIG, and it needs a kpathsea expert to know the difference.

  This is bound to become a source of confusion: People use texconfig once, but manually manipulate the symlink-managed files later, and are surprised that the manual change to a file below /etc/texmf/texlive doesn't have any effect.

I think the clean method for coexistence of two TEX systems is to use separate TEXMFSYSCONFIG trees, e. g. /etc/texmf-texlive/ and /etc/texmf-miktex/.

Part I.
Appendix

A. Files that use \pdfoutput

From my grepping through Master, here's an overview of non-configuration files using \pdfoutput

A.1. Possible (wishlist/normal) bugs

These files try to detect the output format and have a hardcoded switch to dvips - they are not configuration files, but could be enhanced to load one instead. Some of the checks are not ideal.

generic/pgf/utilities/pgfutil-plain.def
generic/pgf/utilities/pgfutil-context.def
generic/thumbpdf/thumbpdf.sty
generic/thumbpdf/thumbpdf.tex
latex/pict2e/p2e-pdftex.def

Hardcoded options for \RequirePackage, or restricts things to PDF/DVI output that probably shouldn't be:

latex/ifmslide/ifmslide.sty
latex/base/ltnews.cls
latex/breakurl/breakurl.sty
latex/ucs/ucshyper.sty
latex/iso10303/stepman.tex

Could use a driver/output format detection, but in fact requires driver options and only uses \pdfoutput for safety checking:

latex/changebar/changebar.sty

These files contain the string pdfoutput, and there's indication that they might use it inappropriately (e. .g. assuming that we're producing PDF when it is defined).

latex/IEEEtran/IEEEtran.cls
latex/hepparticles/hepparticles.sty
latex/teubner/teubner.sty
latex/scientificpaper/science.sty
latex/mla-paper/mla.sty
latex/fmp/fmp.sty
latex/hc/hcart.cls
latex/hc/hcreport.cls
latex/hc/hcslides.cls
latex/iso/isov2.cls
latex/pgf/utilities/pgfwriteexternal.sty
latex/beamer/emulation/beamerprosper.sty
latex/sciposter/sciposter.cls
latex/pbsheet/pbsheet.cls
latex/pracjourn/pracjourn.cls
latex/hepthesis/hepthesis.cls

A.2. Generally okay, but check for side effects

These files seem to (in some cases I have checked it) use `\pdfoutput` as intended by the pdfTeX developers[3].

generic/context/mptopdf.tex
generic/mfpic/mfpic.tex
generic/oberdiek/ifpdf.sty
generic/oberdiek/pdfcrypt.sty
latex/cyrillic/ot2wncyr.fd
latex/cyrillic/ot2wncyss.fd
latex/cyrillic/ot2wlcyss.fd
latex/cyrillic/ot2wlcyr.fd
latex/juramisc/jurabook.cls
latex/leaflet/leaflet.cls
latex/microtype/microtype.sty
latex/hyperref/hpdftex.def
latex/hyperref/hyperref.sty
latex/pdfcprot/pdfcprot.sty
latex/cmap/cmap.sty

[3] A surprising number, however, does it wrong AFAIK (am I right?) and uses `\@ifundefined` without any grouping. This has the consequence that it's impossible to detect whether pdfTeX is used with this primitive later on.

latex/listings/lstdoc.sty
latex/oberdiek/pdflscape.sty
latex/oberdiek/bmpsize-test.tex
latex/beamer/multimedia/multimedia.sty
latex/beamer/beamer.cls
latex/pdfpages/pdfpages.sty
latex/pdfpages/pppdftex.def
latex/memoir/memoir.cls
latex/koma-script/scrkbase.sty
latex/petri-nets/pndraw.sty
latex/texpower/fixseminar.sty
latex/texpower/texpower.sty
latex/pdftex-def/pdftex.def
latex/preview/preview.sty
latex/preview/prtightpage.def
latex/latexconfig/latex.ini
latex/stellenbosch/usthesis.cls
latex/vpe/vpe.sty
latex/marginnote/marginnote.sty
latex/pst-pdf/pst-pdf.sty

A.3. No problem at all

False positives (`pdfoutput` only in comment etc.):

latex/ppower4/texpause.sty
latex/sugconf/sugconf.cls