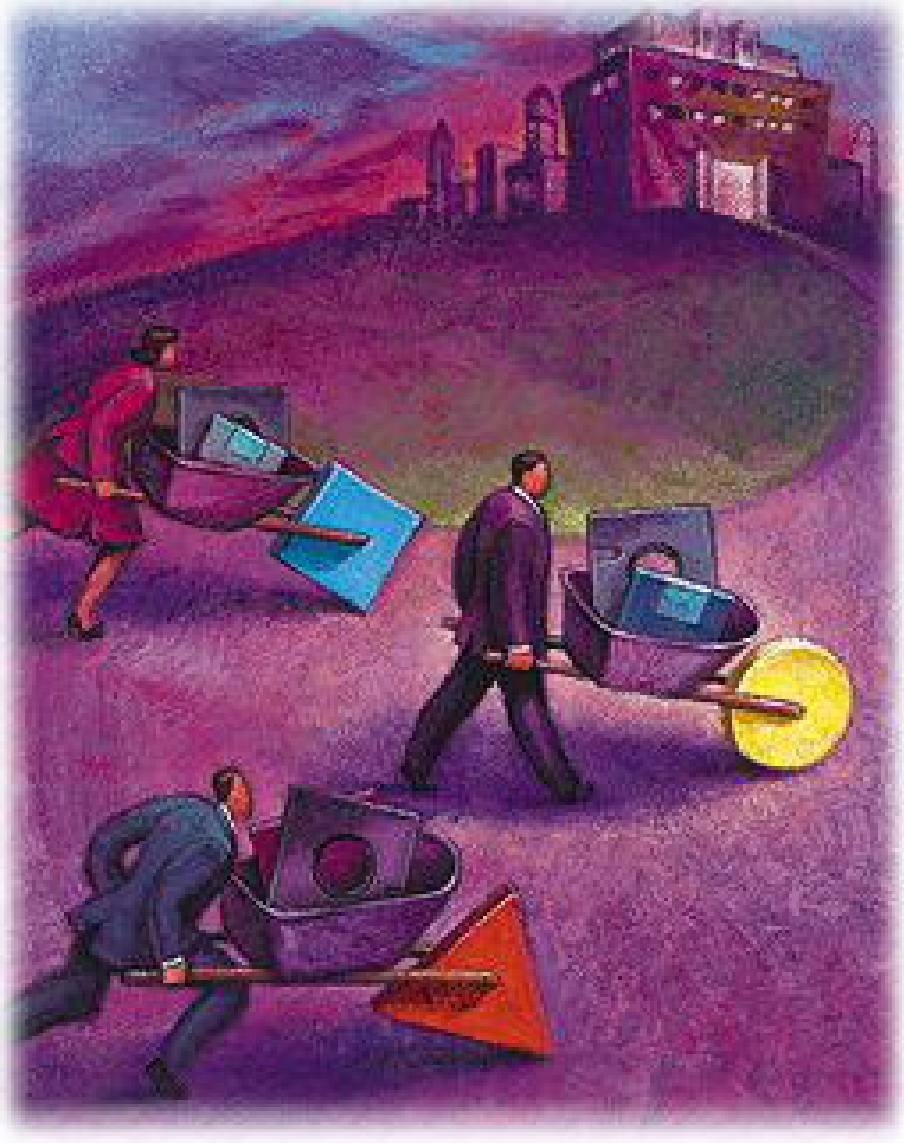

Distributing TEX and Friends

Norbert Preining

JAIST
TEX Live Team
Debian TEX Team

TUG 2013, Tokyo, October 2013

Overview

- introduction to TEX Live
- important configuration files
- infrastructure and package hierarchy
- packaging paradigma
- distribution breakdown
- possible problems and warnings
- recommendations

History

- late 1993 Dutch TEX Users Group, 4AllTEX CD, TDS
- 1995 Unix-based TDS CD based on teTEX
- 1996 first edition, Sebastian Rahtz

History

- late 1993 Dutch TEX Users Group, 4AllTEX CD, TDS
- 1995 Unix-based TDS CD based on teTEX
- 1996 first edition, Sebastian Rahtz
- 2000 5th edition, non-free software removed
- 2002 7th edition: Mac OS X support
- 2005 addition of the -sys scripts

TeX Live 2013

History

- late 1993 Dutch TeX Users Group, 4AllTeX
- 1995 Unix-based TDS CD based on teTeX
- 1996 first edition, Sebastian Rahtz
- 2000 5th edition, non-free software remov
- 2002 7th edition: Mac OS X support
- 2005 addition of the -sys scripts
- 2006–09 XeTeX addition, end of teTeX development, TeX Works additions, `tlmgr` introduction, Karl Berry
- 2009– Japanese TeX support ((e)pTeX, (e)upTeX)
- 2012 `updmap` goes multi-input
- 2013 `texmf` and `texmf-dist` merge

Features

- ‘complete’ – all the free stuff from CTAN
  (currently 2547 packages)
- multi-platform
  (11 archs, 10 os, 21 combinations, plus external)
- uniform across platforms
  (Windows is the hard part)
- own package manager `tlmgr`
  (responsible for update, backup, configuration, …)
- practically daily updates
- DFSG free with a few exceptions

Distribution methods

DVD

once a year around June/July, available permanently on CTAN

`tlnet` distribution

daily updates of packages, `tlmgr` is managing updates (and much more)

stacked versus non-stacked files

non-stacked

more prominent (higher up in the hierarchy of various trees) located files override less prominent ones (same name)

stacked versus non-stacked files

non-stacked

more prominent (higher up in the hierarchy of various trees) located files override less prominent ones (same name)

stacked

all files of the same name are evaluated

Configuration files for (any) TEX system

non-stacked

fmtutil.cnf definition of formats (memory dumps)

language.dat/def/def.lua specification of available hyphenation patterns for different engines

stacked

texmf.cnf configuration of search paths

updmap.cfg meta-listing of available fonts

texmf.cnf – available trees (increasing order)

TEXMFDIST	all the of the TEX Live files
TEXMFLOCAL	system wide additions (R/../texmf-local)
TEXMFSYSVAR	generated data (R/texmf-var)
TEXMFSYSCONFIG	system wide configuration (R/texmf-config)
TEXMFHOME	user tree (~/texmf)
TEXMFVAR	user generated data (~/.texlive2013/texmf-var)
TEXMFCONFIG	user configurations (~/.texlive2013/texmf-config)
VARTEXFONTS	location of generated fonts (TEXMFVAR/fonts)

Possible additions to `texmf.cnf`

- adjustment of tree locations
  e.g., `TEXMFSYSCONFIG = /etc/texmf`
- addition of trees
- anything else should not be necessary!

Possible additions to `texmf.cnf`

- adjustment of tree locations
  e.g., `TEXMFSYSCONFIG = /etc/texmf`
- addition of trees
- anything else should not be necessary!

How to

Since *all* the `texmf.cnf` files are read and evvaluated, please only set the necessary changes in

`TEXMFSYSCONFIG/web2c/texmf.cnf`

Configuration of fonts `updmap.cfg`

- input file for `updmap(-sys)` which generating font configuration files for `dvipdfm(x)`, `pdftex`, `dvips`
- lists configuration options (e.g., selection of font embedding) and font map definition files
- every change of availability of fonts (installation/removal) needs adaption of this file and re-run of `updmap-sys`
- since it is stacked, local system adaptions (adaptions by the admin) can easily be managed by adding one more config file in `TEXMFLOCAL`

New operation mode

all `updmap.cfg` are read in a stacked mode list in order of decreasing priority

System mode

TEXMFSYSCONFIG	$TEXLIVE/YYYY/texmf-config/web2c/updmap.cfg
TEXMFSYSVAR	$TEXLIVE/YYYY/texmf-var/web2c/updmap.cfg
TEXMFLOCAL	$TEXLIVE/texmf-local/web2c/updmap.cfg
TEXMFDIST	$TEXLIVE/temxf-dist/web2c/updmap.cfg

User mode

TEXMFCONFIG	$HOME/.texliveYYYY/texmf-config/web2c/updmap.cfg
TEXMFVAR	$HOME/.texliveYYYY/texmf-var/web2c/updmap.cfg
TEXMFHOME	$HOME/texmf/web2c/updmap.cfg
TEXMFSYSCONFIG	$TEXLIVE/YYYY/texmf-config/web2c/updmap.cfg
TEXMFSYSVAR	$TEXLIVE/YYYY/texmf-var/web2c/updmap.cfg
TEXMFLOCAL	$TEXLIVE/texmf-local/web2c/updmap.cfg
TEXMFDIST	$TEXLIVE/temxf-dist/web2c/updmap.cfg

format definitions `fmtutil.cnf`

- TEX traditionally reads in a precompiled memory dump for faster execution (back in the 80ies...)
- input file for `fmtutil(-sys)` which generates these memory dumps
- not-stacked!
- every change of availability of formats should trigger an adaption of this file, plus a re-run of `fmtutil-sys`

format definitions `fmtutil.cnf`

- TEX traditionally reads in a precompiled memory dump for faster execution (back in the 80ies...)
- input file for `fmtutil(-sys)` which generates these memory dumps
- not-stacked!
- every change of availability of formats should trigger an adaption of this file, plus a re-run of `fmtutil-sys`

adaptions have to be made in the `fmtutil.cnf` file itself

hyphenation patterns `language.dat/.def/.dat.lua`

- these files are read during memory dump generation by `fmtutil`, and include hyphenation patterns in the dump.
- every change of availability of hyphenation patterns should trigger an adaption of these files, plus a re-run of `fmtutil-sys`
- (exception: language.dat.lua is run-time file, no action but updating the file itself necessary)

Structure of TEX Live

Hierarchy of packages

Scheme (currently 9) top most level, overlapping contents

Collection (currently 45) non-overlapping, i.e., partition of the contents, examples: `collection-latex`, `collection-langcjk`

Package (currently ~2500) smallest unit, relates to one item on CTAN, examples: `beamer`, `koma-script`

TEX Live Database `texlive.tlpdb`

One installation or media is now completely described by one file, the TEX Live Database:

- simple text file – easily parseable
- revision number for the single packages
- generated from static content (the tlpsrc files)
- enriched with information from the TEX Catalogue
- format documented in detail (POD documentation in the respective perl module)
- perl modules for the whole db, for each package, lots of scripts using it to take examples

How does `texlive.tlpdb` look like

texlive.tlpdb

```
name abbr
...

name memoir
...
```

- sequence of `key value` pairs
- separated by an empty line (or more)
- one group per package
- some ‘meta’-packages for configuration options

The single package: `tlpobj` by example I

aoposter.tlpobj

```
name a0poster
category Package
revision 15878
shortdesc Support for designing posters on large paper.
longdesc Provides fonts in sizes of 12pt up to 107pt and also makes sure
longdesc that in math formulas the symbols appear in the right size. Can
longdesc also create a PostScript header file for dvips which ensures
longdesc that the poster will be printed in the right size. Supported
longdesc sizes are DIN A0, DIN A1, DIN A2 and DIN A3.
containersize 3320
containermd5 444d5f9f8da1a7812ac40da67d79e4df
doccontainersize 118128
doccontainermd5 1d764851470aab66e5872ecf64ac6bc9
docfiles size=47
 texmf-dist/doc/latex/a0poster/a0.pdf details="Package documentation (German)" language="de"
 texmf-dist/doc/latex/a0poster/a0.tex
 texmf-dist/doc/latex/a0poster/a0_eng.pdf details="Package documentation (English)" language="en"
 texmf-dist/doc/latex/a0poster/a0_eng.tex
runfiles size=4
 texmf-dist/tex/latex/a0poster/a0poster.cls
 texmf-dist/tex/latex/a0poster/a0size.sty
catalogue-ctan /macros/latex/contrib/a0poster
catalogue-date 2006-11-28 22:38:04 +0100
catalogue-license lppl
catalogue-version 1.22b
```

The execute statements

addMap, addMixedMap, addKanjiMap lists entries that have to be added to updmap.cfg

```
execute addMap grotesqvn.map
```

AddFormat specifies one entry for fmtutil.cnf

```
execute AddFormat name=pdflatex engine=pdftex
patterns=language.dat
options="-translate-file=cp227.tcx *pdflatex.ini"
```

AddHyphen specifies one entry for the files language.*

```
execute AddHyphen name=basque lefthyphenmin=2
righthyphenmin=2 file=loadhyph-eu.tex
file_patterns=hyph-eu.pat.txt file_exceptions=
```

Packaging paradigma

How to package about 4Gb of data?

Packaging paradigma

How to package about 4Gb of data?

all-or-nothing one dist-package which contains everything

collection-splitting one dist-package per collection

single-package one dist-package per TL-package

mixed-mode some intermediate mode

All-or-nothing packaging

Advantages

easy, no need for special dealing with config files, just use what we ship

All-or-nothing packaging

Advantages

easy, no need for special dealing with config files, just use what we ship

Disadvantages

big … be warned – there will be *many* complaints

All-or-nothing packaging

Advantages

easy, no need for special dealing with config files, just use what we ship

Disadvantages

big … be warned – there will be *many* complaints

I don't know of any distribution using this approach (MacTEX?)

All-or-nothing packaging

collection-splitting

Advantages

package manager friendly since the collections are disjoint, conceptually close tl-packages are in the same collection, thus in the same dist-package, not too many individual packages

collection-splitting

Advantages

package manager friendly since the collections are disjoint, conceptually close tl-packages are in the same collection, thus in the same dist-package, not too many individual packages

Disadvantages

complex packaging since the execute statements have to be collected and evaluated, dependencies are sometimes missing

collection-splitting

Advantages

package manager friendly since the collections are disjoint, conceptually close tl-packages are in the same collection, thus in the same dist-package, not too many individual packages

Disadvantages

complex packaging since the execute statements have to be collected and evaluated, dependencies are sometimes missing

Debian/Ubuntu uses tha apprach

collection-splitting

package splitting

Advantages

staight forward, fine-grained installations possible

package splitting

Advantages

staight forward, fine-grained installations possible

Disadvantages

huge number of packages, no way to treat manually, missing dependencies

package splitting

Advantages

staight forward, fine-grained installations possible

Disadvantages

huge number of packages, no way to treat manually, missing dependencies

Fedara, SuSE are using this approach.

package splitting

mixed-mode

Advantages

???

Disadvantages

???

mixed-mode

Advantages

???

Disadvantages

???

some of the BSD variants seem to use that approach

Distribution breakdown – overview

Debian	since etch, collection-splitting
Fedora	since FC6, package-splitting
SuSE	since SLE11, package-splitting
FreeBSD	since TL2009, package-splitting
OpenBSD	since TL2007, mixed-splitting, very large
NetBSD	work in progress
MacOS	gwTEX since 2007, MacTEX (based on TEX Live)
Windows	proTEXt, upstream, MikTEX (independent)

Debian (and Ubuntu)

- woody: teTEX 1; sarge: teTEX 2; etch: teTEX 3, TL 2005; lenny: TL2007; squeeze: TL2009; wheezy: TL2012; next: TL2013+
- hardy: TL2007; lucid, natty, oneiric, precise: TL2009; quantal: TL2012; 13.10: TL2013
- one package per one collection
- arch-dependent build from separate sources (texlive-binaries)
- TEXLIVEROOT = /usr/share/texlive
- additional tree `TEXMFDEBIAN = /usr/share/texmf`
- persistency of admin changes to format/hyphen configuration in `/etc/texmf`

Fedora

- Fedora till 5 and RHEL 5: teTEX; TL2007 since FC6; TL2010 for F13, F14; TL2011 for F13-F17; TL2012 for F16-, TL2013 for F20- (?)
- singe-package splitting
- very detailed license check (thanks!)
- `TEXLIVEROOT = /usr/share/texlive`
- no persistency of admin adaptions, postinst changes config files below `TEXLIVEROOT`

SuSE

- SLE11: TL2010; openSuSE 11.4: TL2010 and TL2011; 12.1: TL2011; 12.2: TL2011 and TL2012; 12.3: TL2012, 13.3: TL2013 (?)
- single-package splitting
- `texmf-dist` is becoming `/usr/share/texmf`
- config files are kept in `/etc/texmf`, but no persistency

Other OS

- FreeBSD since TL2009, currently TL2012, one-package splitting, hard to see how formats etc are handled
- OpenBSD since TL2007, mixed splitting into four: base, minimal, full, docs; currently TL2012
- NetBSD: work in progress
- MacOS: very nice wrap up of TEX Live into a package with a bit of front end and configuration: MacTEX
- Windows: TEX Live upstream, repackaged and slightly adapted: W32TEX, independent MikTEX, proTEXt

Warnings and common pitfalls

I never used TEX Live and I don't know what TEX does, but I package it! – please no!

Warnings and common pitfalls

- I never used TEX Live and I don't know what TEX does, but I package it! – please no!
- improper configuration file handling by far the biggest problem (shipping parts of TEX Live with the full `updmap.cfg` does *not* work

Warnings and common pitfalls

- I never used TEX Live and I don't know what TEX does, but I package it! – please no!
- improper configuration file handling by far the biggest problem (shipping parts of TEX Live with the full `updmap.cfg` does *not* work
- what is 'upstream' only one fixed release, other than that daily updates with no way to fetch the packages of a specific day (especially the BSD ports have problems with that)

Warnings and common pitfalls II

binaries and source TEX Live does *not* update binaries over the year, but the sources changes in svn, source building is complex, loads of scripts that are partially linked

Warnings and common pitfalls II

binaries and source TEX Live does *not* update binaries over the year, but the sources changes in svn, source building is complex, loads of scripts that are partially linked

shipping the `tlmgr` Only do it fixed in *user mode*. One cannot mix package managers, you will not want to have another program changing the files under the nose of the main distribution package manager

Recommendations

Get to know the system — use the normal installer and learn what is going on in TEX Live

learn Perl — there are many perl modules to make programming easier, parsing of executes, generation of proper configuration files, etc., all done already

look around — TEX Live has now been packaged several times, starting with Debian in 2005

select a paradigm — fitting to distribution's need

contact us — we have our own mailing list for distributors

Contact

Web page	`http://tug.org/texlive`
Main coordination ML	`texlive@tug.org`
Distributors ML	`tldistro@tug.org`
myself	`norbert@preining.info`

Contact

Web page	`http://tug.org/texlive`
Main coordination ML	`texlive@tug.org`
Distributors ML	`tldistro@tug.org`
myself	`norbert@preining.info`

Thanks for the attention