

# *The T<sub>E</sub>X Live Guide*

T<sub>E</sub>X Live 2008

Karl Berry, editor

<http://tug.org/texlive/>

July 2008

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	T <sub>E</sub> X Live and the T <sub>E</sub> X Collection . . . . .	3
1.2	Operating system support . . . . .	3
1.3	Basic installation of T <sub>E</sub> X Live . . . . .	3
1.4	Getting help . . . . .	3
<b>2</b>	<b>Overview of T<sub>E</sub>X Live</b>	<b>4</b>
2.1	The T <sub>E</sub> X Collection: T <sub>E</sub> X Live, proT <sub>E</sub> Xt, MacT <sub>E</sub> X . . . . .	4
2.2	Top level T <sub>E</sub> X Live directories . . . . .	5
2.3	Overview of the predefined texmf trees . . . . .	5
2.4	Extensions to T <sub>E</sub> X . . . . .	6
2.5	Other notable programs in T <sub>E</sub> X Live . . . . .	7
2.6	Fonts in T <sub>E</sub> X Live . . . . .	7
<b>3</b>	<b>Installation</b>	<b>8</b>
3.1	Starting the installer . . . . .	8
3.1.1	Unix . . . . .	8
3.1.2	Mac OS X . . . . .	8
3.1.3	Windows . . . . .	9
3.1.4	The text installer . . . . .	10
3.1.5	The graphical installer . . . . .	10
3.2	Running the installer . . . . .	11
3.2.1	Binary systems menu (Unix only) . . . . .	11
3.2.2	Selecting what is going to be installed . . . . .	11
3.2.3	Directories . . . . .	12
3.2.4	Options . . . . .	13
3.2.5	Set up for running from DVD (text mode only) . . . . .	13
3.3	Command-line install options . . . . .	14
3.3.1	The <code>location</code> parameter . . . . .	15
3.4	Post-install actions . . . . .	15
3.4.1	Windows . . . . .	15
3.4.2	If symlinks were created . . . . .	15
3.4.3	Environment variables for Unix . . . . .	15
3.4.4	Environment variables: global configuration . . . . .	16

3.4.5	When running from DVD	16
3.4.6	ConT <sub>E</sub> Xt Mark IV	16
3.4.7	Integrating local and personal macros	16
3.4.8	Integrating third-party fonts	17
3.5	Testing the installation	17
<b>4</b>	<b>Network installations</b>	<b>18</b>
<b>5</b>	<b>Maximally portable T<sub>E</sub>X Live on DVD and USB</b>	<b>19</b>
<b>6</b>	<b>tlmgr: Managing your installation</b>	<b>19</b>
6.1	tlmgr GUI mode	19
6.2	Sample tlmgr command-line invocations	19
<b>7</b>	<b>Notes on Windows</b>	<b>22</b>
7.1	Windows-specific features	22
7.2	Additional included software	23
7.3	The Dviout previewer	23
7.4	Links for additional downloadable software	24
7.5	User Profile is Home	24
7.6	The Windows registry	24
7.7	Windows permissions	25
<b>8</b>	<b>A user's guide to Web2C</b>	<b>25</b>
8.1	Kpathsea path searching	26
8.1.1	Path sources	27
8.1.2	Config files	27
8.1.3	Path expansion	28
8.1.4	Default expansion	28
8.1.5	Brace expansion	28
8.1.6	Subdirectory expansion	29
8.1.7	List of special characters and their meaning: a summary	29
8.2	Filename databases	29
8.2.1	The filename database	30
8.2.2	kpsewhich: Standalone path searching	30
8.2.3	Examples of use	31
8.2.4	Debugging actions	32
8.3	Runtime options	35
<b>9</b>	<b>Acknowledgements</b>	<b>35</b>
<b>10</b>	<b>Release history</b>	<b>37</b>
10.1	Past	37
10.1.1	2003	38
10.1.2	2004	38
10.1.3	2005	40
10.1.4	2006–2007	41
10.2	Present	41
10.3	Future	42

## 1 Introduction

### 1.1 T<sub>E</sub>X Live and the T<sub>E</sub>X Collection

This document describes the main features of the T<sub>E</sub>X Live software distribution—T<sub>E</sub>X and related programs for GNU/Linux and other Unix flavors, Mac OS X, and Windows systems.

You may have acquired T<sub>E</sub>X Live by downloading, or on the T<sub>E</sub>X Collection DVD, which T<sub>E</sub>X usergroups distribute among their members. Section 2.1 briefly describes the contents of this DVD. Both T<sub>E</sub>X Live and the T<sub>E</sub>X Collection are cooperative efforts by the T<sub>E</sub>X user groups. This document mainly describes T<sub>E</sub>X Live itself.

T<sub>E</sub>X Live includes executables for T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, ConT<sub>E</sub>Xt, METAFONT, MetaPost, BibT<sub>E</sub>X and many other programs; an extensive collection of macros, fonts and documentation; and support for typesetting in many different scripts from around the world.

T<sub>E</sub>X Live packages are regularly updated from CTAN: <http://www.ctan.org>.

For a brief summary of the major changes in this edition of T<sub>E</sub>X Live, see the end of the document, section 10 (p. 37).

### 1.2 Operating system support

T<sub>E</sub>X Live contains binaries for many Unix-based architectures, including Mac OS X. The included sources should enable installation on platforms for which we do not have binaries.

As to Windows: only Windows 2000 and later are supported. Windows 9x, ME and NT have been dropped. Because of this change, Windows requires much less special treatment compared to the Unix systems. There are no 64-bit executables for Windows, but the 32-bit executables should run on 64-bit systems.

See section 2.1 for alternate solutions for Windows and Mac OS X.

### 1.3 Basic installation of T<sub>E</sub>X Live

You can install T<sub>E</sub>X Live either from DVD or over the Internet. The net installer itself is small, and downloads everything requested from the Internet. The net installer is an attractive option if you need only a fraction of the complete T<sub>E</sub>X Live.

The DVD installer lets you install to a local disk, but you can also run T<sub>E</sub>X Live directly from the DVD (or from a DVD image, if your system supports that).

This is described in the installation sections following, but here is a quick start:

- The installation script is named `install-tl`. It can operate in both a GUI mode (default for Windows) and a text mode (default for everything else). You can force one mode or the other with the options `-gui` and `-no-gui`, respectively.
- One of the installed items is a new program, the ‘T<sub>E</sub>X Live Manager’, named `tlmgr`. Like the installer, it can be used in both GUI mode and in text mode. You can use it to install and uninstall packages and do various configuration tasks.

### 1.4 Getting help

The T<sub>E</sub>X community is both active and friendly, and most serious questions end up getting answered. However, the support is informal, done by volunteers and casual readers, so it’s especially important that you do your homework before asking. (If you prefer guaranteed commercial support, you can forego T<sub>E</sub>X Live completely and purchase a vendor’s system; <http://tug.org/interest.html#vendors> has a list.)

Here is a list of resources, approximately in the order we recommend using them:

**Getting Started** If you are new to T<sub>E</sub>X, the web page <http://tug.org/begin.html> gives a brief introduction to the system.

**T<sub>E</sub>X FAQ** The T<sub>E</sub>X FAQ is a huge compendium of answers to all sorts of questions, from the most basic to the most arcane. It is included on T<sub>E</sub>X Live in <texmf-doc/doc/english/FAQ-en/html/index.html>, and is available on the Internet through <http://www.tex.ac.uk/faq>. Please check here first.

**T<sub>E</sub>X Catalogue** If you are looking for a specific package, font, program, etc., the T<sub>E</sub>X Catalogue is the place to look. It is a huge collection of all T<sub>E</sub>X-related items. See <http://www.ctan.org/tex-archive/help/Catalogue/index.html>.

**T<sub>E</sub>X Web Resources** The web page <http://tug.org/interest.html> has many T<sub>E</sub>X-related links, in particular for numerous books, manuals, and articles on all aspects of the system.

**support archives** The two principal support forums are the Usenet newsgroup <news:comp.text.tex> and the mailing list [texhax@tug.org](mailto:texhax@tug.org). Their archives have years of past questions and answers for your searching pleasure, via <http://groups.google.com/groups?group=comp.text.tex> and <http://tug.org/mail-archives/texhax>, respectively. And a general web search, for example on <http://google.com>, never hurts.

**asking questions** If you cannot find an answer, you can post to [comp.text.tex](news:comp.text.tex) through Google or your newsreader, or to [texhax@tug.org](mailto:texhax@tug.org) through email. But before you post, *please* read this FAQ entry, to maximize your chances of getting a useful answer: <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=askquestion>.

**T<sub>E</sub>X Live support** If you want to report a bug or have suggestions or comments on the T<sub>E</sub>X Live distribution, installation, or documentation, the mailing list is [tex-live@tug.org](mailto:tex-live@tug.org). However, if your question is about how to use a particular program included in T<sub>E</sub>X Live, please write to that program's maintainer or mailing list. Often running a program with the `--help` option will provide a bug reporting address.

The other side of the coin is helping others who have questions. Both [comp.text.tex](news:comp.text.tex) and [texhax](mailto:texhax) are open to anyone, so feel free to join, start reading, and help out where you can.

## 2 Overview of T<sub>E</sub>X Live

This section describes the contents of T<sub>E</sub>X Live and the T<sub>E</sub>X Collection of which it is a part.

### 2.1 The T<sub>E</sub>X Collection: T<sub>E</sub>X Live, proT<sub>E</sub>Xt, MacT<sub>E</sub>X

The T<sub>E</sub>X Collection DVD comprises the following:

**T<sub>E</sub>X Live** A complete T<sub>E</sub>X system which can be run live or installed to disk.

**MacT<sub>E</sub>X** for Mac OS X, this adds a native Mac OS X installer and other Mac applications to T<sub>E</sub>X Live. Its home page is <http://www.tug.org/mactex/>.

**proT<sub>E</sub>Xt** An enhancement of the MiK<sub>T</sub><sub>E</sub>X distribution for Windows, ProT<sub>E</sub>Xt adds a few extra tools to MiK<sub>T</sub><sub>E</sub>X, and simplifies installation. It is entirely independent of T<sub>E</sub>X Live, and has its own installation instructions. The ProT<sub>E</sub>Xt home page is <http://tug.org/protext>.

**CTAN** A snapshot of the CTAN repository.

**texmf-extra** A directory with assorted additional packages.

CTAN, protext, and **texmf-extra** do not necessarily follow the same copying conditions as T<sub>E</sub>X Live, so be careful when redistributing or modifying.

## 2.2 Top level T<sub>E</sub>X Live directories

Here is a brief listing and description of the top level directories in the T<sub>E</sub>X Live distribution. On the live DVD, the entire T<sub>E</sub>X Live hierarchy is in a subdirectory `texlive`, not at the top level of the disc.

- `bin` The T<sub>E</sub>X system programs, arranged by platform.
- `readme.html` Web pages with a quick introduction and useful links, in various languages.
- `source` The source to all included programs, including the main Web2C T<sub>E</sub>X and METAFONT distributions.
- `support` Various auxiliary packages and programs. These are *not* installed automatically. This includes assorted editors and T<sub>E</sub>X shells.
- `texmf` See `TEXMFMAIN` below.
- `texmf-dist` See `TEXMFDIST` below.
- `texmf-doc` Tree for self-contained pure documentation, arranged by language.
- `tlpkg` Scripts, programs and data for managing the installation. It also contains private copies of Perl and Ghostscript for Windows, which are invisible outside T<sub>E</sub>X Live, and a new Windows PostScript viewer PSView.

In addition to the directories above, the installation scripts and `README` files (in various languages) are at the top level of the distribution.

The `texmf-doc` directory contains documentation which does not belong to any particular package or program. The documentation for the programs (manuals, man pages, Info files) is in `texmf/doc`. The documentation for T<sub>E</sub>X packages and formats is in `texmf-dist/doc`. You can use the `texdoc` or `texdoctk` programs to find any documentation wherever it is located. The comprehensive links in the top-level file `doc.html` may also be helpful.

This T<sub>E</sub>X Live documentation itself is in `texmf-doc` and is available in several languages:

- Chinese: `texmf-doc/doc/chinese/texlive-zh-cn`
- Czech/Slovak: `texmf-doc/doc/czechslovak/texlive-cz`
- English: `texmf-doc/doc/english/texlive-en`
- French: `texmf-doc/doc/french/texlive-fr`
- German: `texmf-doc/doc/german/texlive-de`
- Polish: `texmf-doc/doc/polish/texlive-pl`
- Russian: `texmf-doc/doc/russian/texlive-ru`

## 2.3 Overview of the predefined texmf trees

This section lists all predefined variables specifying the `texmf` trees used by the system, and their intended purpose, and the default layout of a T<sub>E</sub>X Live system. The command `texconfig conf` shows you the values of these variables, so that you can easily find out how they map to directory names in your installation.

- `TEXMFMAIN` The tree which holds vital parts of the system such as helper scripts (e.g., `web2c/mktextdir`) and other support files and their documentation. Does not include T<sub>E</sub>X formats and packages.
- `TEXMFDIST` The tree which holds the main set of macro packages, fonts, etc., as originally distributed.
- `TEXMFLocal` The tree which administrators can use for system-wide installation of additional or updated macros, fonts, etc.

**TEXMFHOME** The tree which users can use for their own individual installations of additional or updated macros, fonts, etc. The expansion of this variable dynamically adjusts for each user to their own individual directory.

**TEXMFCONFIG** The tree used by the utilities `texconfig`, `updmap`, and `fmtutil` to store modified configuration data. Under **TEXMFHOME** by default.

**TEXMFSYSCONFIG** The tree used by the utilities `texconfig-sys`, `updmap-sys`, and `fmtutil-sys` to store modified configuration data.

**TEXMFVAR** The tree used by `texconfig`, `updmap` and `fmtutil` to store (cached) runtime data such as format files and generated map files. Under **TEXMFHOME** by default.

**TEXMFSYSVAR** The tree used by `texconfig-sys`, `updmap-sys` and `fmtutil-sys`, and also by `tlmgr`, to store (cached) runtime data such as format files and generated map files.

The default layout is:

**system-wide root** can span multiple T<sub>E</sub>X Live releases:

2007 A previous release.

...

2008 The current release.

`bin`

`i386-linux` GNU/Linux binaries

...

`universal-darwin` Mac OS X binaries

`win32` Windows binaries

`texmf` This is **TEXMFMAIN**.

`texmf-dist` **TEXMFDIST**

`texmf-doc` **TEXMFDOC**

`texmf-var` **TEXMFSYSVAR**

`texmf-config` **TEXMFSYSCONFIG**

`texmf-local` **TEXMFLOCAL**, intended to be retained from release to release.

**user's home** (`$HOME` or `%USERPROFILE%`)

`.texlive2007` Privately generated and configuration data for a previous release.

`.texlive2008` Privately generated and configuration data for the current release.

`texmf-var` **TEXMFVAR**

`texmf-config` **TEXMFCONFIG**

`texmf` **TEXMFHOME** Personal macros. *etc.*

## 2.4 Extensions to T<sub>E</sub>X

T<sub>E</sub>X itself is frozen, apart from rare bug fixes. It is still present in T<sub>E</sub>X Live as the program `tex`, and will remain so in the foreseeable future. T<sub>E</sub>X Live contains several extended versions of T<sub>E</sub>X:

**ε-T<sub>E</sub>X** adds a set of new primitives (related to macro expansion, character scanning, classes of marks, additional debugging features, and more) and the T<sub>E</sub>X-X<sub>E</sub>T extensions for bidirectional typesetting. In default mode, ε-T<sub>E</sub>X is 100% compatible with ordinary T<sub>E</sub>X. See [texmf-dist/doc/etex/base/etex\\_man.pdf](http://texmf-dist/doc/etex/base/etex_man.pdf).

**pdfT<sub>E</sub>X** builds on the  $\varepsilon$ -T<sub>E</sub>X extensions, adding support for writing PDF output as well as DVI. This is the program invoked for most formats, e.g., `etex`, `latex`, `pdflatex`. See <http://www.pdfTeX.org/.texmf/doc/pdftex/manual/> for the manual, and [texmf/doc/pdftex/manual/samplepdf/samplepdf.tex](http://www.pdfTeX.org/.texmf/doc/pdftex/manual/samplepdf/samplepdf.tex) for example usage of some of its features.

**LuaT<sub>E</sub>X** is the designated successor of pdfT<sub>E</sub>X, with which it is intended to be backward-compatible. It should also supersede Aleph, see below. The incorporated Lua interpreter (see <http://www.lua.org/>) enables elegant solutions for many thorny T<sub>E</sub>X problems. When called as `texlua`, it functions as a standalone Lua interpreter, and is already used as such within T<sub>E</sub>X Live. See <http://www.luatex.org/> and [texmf-dist/doc/luatex/luatexref-t.pdf](http://www.texmf-dist/doc/luatex/luatexref-t.pdf).

**XeT<sub>E</sub>X** adds support for Unicode input and OpenType- and system fonts, implemented using standard third-party libraries. See <http://scripts.sil.org/xetex>.

**Ω (Omega)** is based on Unicode (16-bit characters), thus supports working with almost all the world's scripts simultaneously. It also supports so-called 'Ω Translation Processes' (OTPs), for performing complex transformations on arbitrary input. See [texmf-dist/doc/omega/base/doc-1.8.tex](http://www.texmf-dist/doc/omega/base/doc-1.8.tex) (not completely up-to-date).

**Aleph** combines the Ω and  $\varepsilon$ -T<sub>E</sub>X extensions. See [texmf-dist/doc/aleph/base](http://www.texmf-dist/doc/aleph/base) for some minimal documentation.

## 2.5 Other notable programs in T<sub>E</sub>X Live

Here are a few other commonly-used programs included in T<sub>E</sub>X Live:

`bibtex` bibliography support.

`makeindex` index support.

`dvips` convert DVI to PostScript.

`xdvi` DVI previewer for the X Window System.

`dvilj` DVI drive for the HP LaserJet family.

`dv2dt`, `dt2dv` convert DVI to/from plain text.

`dviconcat`, `dviselect` cut and paste pages from DVI files.

`dvipdfmx` convert DVI to PDF, an alternative approach to pdfT<sub>E</sub>X (mentioned above). See the `ps4pdf` and `pdftricks` packages for still more alternatives.

`psselect`, `psnup`, ... PostScript utilities.

`lacheck` L<sup>A</sup>T<sub>E</sub>X syntax checker.

`texexec`, `texmfstart` ConT<sub>E</sub>Xt and PDF processor.

`tex4ht` T<sub>E</sub>X to HTML (and XML and more) converter.

## 2.6 Fonts in T<sub>E</sub>X Live

T<sub>E</sub>X Live comes with many high-quality scalable fonts. See <http://tug.org/fonts> and [texmf-dist/doc/english/free-math-fonts-survey](http://www.texmf-dist/doc/english/free-math-fonts-survey).

## 3 Installation

### 3.1 Starting the installer

To start with, get the T<sub>E</sub>X Collection DVD or download the T<sub>E</sub>X Live net installer, and locate the installer script: `install-tl` for Unix, `install-tl.bat` for Windows.

**Net installer:** Download from CTAN, under `systems/texlive/tlnet`; the url <http://mirror.ctan.org/systems/texlive/tlnet> will automatically redirect to a nearby, up-to-date, mirror. You can retrieve either `install-tl.zip` which can be used under Unix and Windows and `install-unx.tar.gz` which is considerably smaller, for Unix only. After unpacking, `install-tl` and `install-tl.bat` will be in the `install-tl` subdirectory.

**T<sub>E</sub>X Collection DVD:** go to its `texlive` subdirectory. Under Windows, the installer should start automatically when you insert the DVD. You can get the DVD by becoming a member of a T<sub>E</sub>X user group (highly recommended, <http://tug.org/usergroups.html>), or purchasing it separately (<http://tug.org/store>), or burning your own from the ISO image (<http://tug.org/texlive/acquire.html>).

The following sections explain in more detail.

#### 3.1.1 Unix

Below, `>` denotes the shell prompt; user input is **bold**. Starting in a terminal window on a Unix-compatible system, the simplest way:

```
> cd /path/to/installer
> perl install-tl
```

(Or you can invoke `perl /path/to/installer/install-tl`, or `./install-tl` if it stayed executable, etc.; we won't repeat all these variations.)

To install in GUI mode (you'll need the Perl/TK module), use:

```
> perl install-tl -gui
```

For a complete listing of the various options:

```
> perl install-tl -help
```

**Warning about Unix permissions:** Your `umask` at the time of installation will be respected by the T<sub>E</sub>X Live installer. Therefore, if you want your installation to be usable by users other than you, make sure your setting is sufficiently permissive, for instance, `umask 002`. For more information about `umask`, consult your system documentation.

#### 3.1.2 Mac OS X

As mentioned in section 2.1, a separate distribution is prepared for Mac OS X, named MacT<sub>E</sub>X (<http://tug.org/mactex>). We recommend using the native MacT<sub>E</sub>X installer instead of the T<sub>E</sub>X Live installer on Mac OS X, because the native installer can make a few Mac-specific adjustments, in particular to allow easily switching between the various T<sub>E</sub>X distributions for Mac OS X (MacT<sub>E</sub>X, gwT<sub>E</sub>X, Fink, MacPorts, ...).

MacT<sub>E</sub>X is firmly based on T<sub>E</sub>X Live, and the main T<sub>E</sub>X trees are precisely the same. It does add a few extra folders with Mac-specific documentation and applications.



```

===== TeX Live installation procedure =====
===== Note: Letters/digits in <angle brackets> indicate menu items =====
===== for commands or configurable options =====

Detected platform: Intel x86 with GNU/Linux

<B> binary systems: 1 out of 16

<S> Installation scheme (scheme-full)

Customizing installation scheme:
<C> standard collections
<L> language collections
83 collections out of 84, disk space required: 1666 MB

<D> directories:
  TEXDIR (the main TeX directory):
    !! default location: /usr/local/texlive/2008
    !! is not writable, please select a different one!
  TEXMFLOCAL (directory for site-wide local files):
    /usr/local/texlive/texmf-local
  TEXMFSYSVAR (directory for variable and automatically generated data):
    /usr/local/texlive/2008/texmf-var
  TEXMFSYSCONFIG (directory for local config):
    /usr/local/texlive/2008/texmf-config
  TEXMFHOME (directory for user-specific files):
    ~/texmf

<O> options:
  [ ] use letter size instead of A4 by default
  [X] create all format files
  [X] install macro/font doc tree
  [X] install macro/font source tree
  [ ] create symlinks in standard directories

<V> Install for running from DVD

Other actions:
<I> start installation to HD
<H> help
<Q> quit

Enter command:

```

Figure 1: Main text installer screen (GNU/Linux)

### 3.1.3 Windows

If you are using the net installer, or the DVD installer failed to start automatically, double-click `install-tl.bat`.

You can also start the installer from the command-prompt. Below, `>` denotes the prompt; user input is **bold**. If you are in the installer directory, run just:

```
> install-tl
```

Or you can invoke it with an absolute location, such as:

```
> D:\texlive\install-tl
```

for the T<sub>E</sub>X Collection DVD, supposing that D: is the optical drive.

To install in text mode, use:

```
> install-tl -no-gui
```

For a complete listing of the various options:

```
> install-tl -help
```

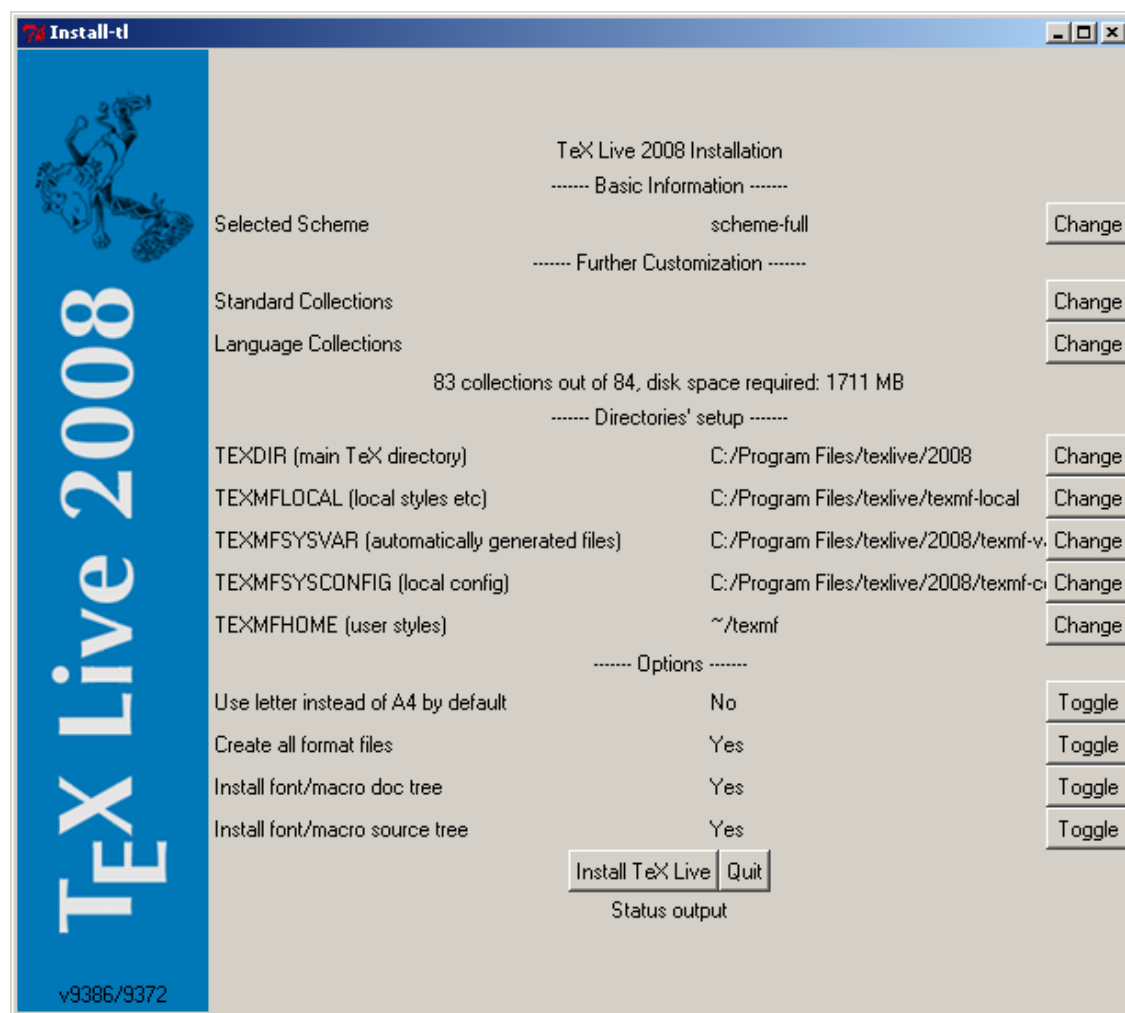


Figure 2: Main GUI installer screen (Windows)

### 3.1.4 The text installer

Figure 1 displays the main text mode screen under Unix.

This is only a command-line installer; there is no cursor support at all. For instance, you cannot tab around checkboxes or input fields. You just type something (case-sensitive) at the prompt and press the Enter key, and then the entire terminal screen will be rewritten, with adjusted content.

The text installer interface is this primitive for a reason: it is designed to run on as many platforms as possible, even with a very barebones Perl (for Windows, a Perl subset is included).

### 3.1.5 The graphical installer

Figure 2 displays the graphical installer under Windows. Most differences in available options are due to this screenshot being from Windows, and the previous text screen being from GNU/Linux.

## 3.2 Running the installer

The installer is intended to be mostly self-explanatory. Still, a few quick notes about the various options and submenus:

### 3.2.1 Binary systems menu (Unix only)

```
Available sets of binaries:
=====
a [ ] alpha-linux      DEC Alpha with GNU/Linux
b [ ] amd64-freebsd    x86_64 with FreeBSD
c [ ] hppa-hpux        HP-UX
d [ ] i386-freebsd     Intel x86 with FreeBSD
e [X] i386-linux       Intel x86 with GNU/Linux
f [ ] i386-openbsd     Intel x86 with OpenBSD
g [ ] i386-solaris     Intel x86 with Sun Solaris
h [ ] mips-irix        SGI IRIX
i [ ] powerpc-aix      PowerPC with AIX
j [ ] powerpc-linux    PowerPC with GNU/Linux
k [ ] sparc-linux      Sparc with GNU/Linux
l [ ] sparc-solaris    Sparc with Solaris
m [ ] universal-darwin universal binaries for MacOSX/Darwin
o [ ] win32            Windows
p [ ] x86_64-linux     x86_64 with GNU/Linux
```

Figure 3: Binaries menu

Figure 3 displays the text mode binaries menu. By default, only the binaries for your current platform will be installed. From this menu, you can select installation of binaries for other architectures as well. This can be useful if you are sharing a T<sub>E</sub>X tree across a network of heterogeneous machines, or for a dual-boot system.

### 3.2.2 Selecting what is going to be installed

```
Select a scheme:
=====
a [ ] TeX Live basic scheme
b [ ] TeX Live scheme for ConTeXt
c [X] TeX Live full scheme
d [ ] GUST TeX Live scheme
e [ ] GUTenberg TeX Live scheme
f [ ] TeX Live medium scheme
g [ ] TeX Live minimal scheme
h [ ] TeX Live for Omega users
i [ ] teTeX scheme
j [ ] XML typesetting scheme
```

Figure 4: Scheme menu

Figure 4 displays the T<sub>E</sub>X Live scheme menu; from here, you choose a “scheme”, which is an overall set of package collections. The default **full** scheme installs everything available, but you can also choose the **basic** scheme for a small system, **minimal** for testing purposes, and **medium** or **teTeX** to get something in between. There are also various specialized and country-specific schemes.

You can refine your scheme selection with the ‘standard collections’ and ‘language collections’ menus (figure 5, shown in GUI mode for a change).



Figure 5: Collections menu (Linux)

Collections are one level more detailed than schemes—in essence, a scheme consists of several collections, a collection consists of one or more packages, and a package (the lowest level grouping in  $\text{\TeX}$  Live) contains the actual  $\text{\TeX}$  macro files, font files, and so on.

If you want more control than the collection menus provide, you can use the `tlmgr` program after installation (see section 6); using that, you can control the installation at the package level.

### 3.2.3 Directories

The default layout is described in section 2.3, p. 5. The default location of `TEXDIR` is different between Windows (`%PROGRAMFILES%\texlive\2008`) and Unix (`/usr/local/texlive/2008`).

The main reason to change this default is if you lack write permission for the default location. You don't have to be root or administrator to install  $\text{\TeX}$  Live, but you do need write access to the target directory.

A logical alternate choice is a directory under your home directory, especially if you will be the sole user. Use `~` to indicate this, as in `~/texlive/2008`.

We recommend including the year in the name, to enable keeping different releases of  $\text{\TeX}$  Live side by side. (You may wish to make a version-independent name such as `/usr/local/texlive-cur` via a symbolic link, which you can then update after testing the new release.)

Changing `TEXDIR` in the installer will also change `TEXMFLOCAL`, `TEXMFSYSVAR` and `TEXMFSYSCONFIG`.

`TEXMFHOME` is the recommended location for personal macro files or packages. The default value is `~/texmf`. In contrast to `TEXDIR`, here a `~` is preserved in the newly-written configuration files, since it usefully refers to the home directory of each individual running `TEX`. It expands to `$HOME` on Unix and `%USERPROFILE%` on Windows.

### 3.2.4 Options

```

<P> use letter size instead of A4 by default: [ ]
<F> create all format files:                  [X]
<D> install font/macro doc tree:              [X]
<S> install font/macro source tree:           [X]
<L> create symlinks in standard directories: [ ]
      binaries to:
      manpages to:
      info to:

```

Figure 6: Options menu (Unix)

Figure 6 displays the text mode options menu. Three of the options here need mentioning:

**use letter size instead of A4 by default:** The usual default paper size selection. Of course, individual documents can specify a specific paper size, if desired.

**create all format files:** Although unnecessary format files take time to generate and disk space to store, it is still recommended to leave this option checked: if you don't, then format files will be generated in people's private `TEXMFVAR` tree as they are needed. In that location, they will not be updated automatically if (say) binaries or hyphenation patterns are updated in the installation, and thus could end up with incompatible format files.

**create symlinks in standard directories (Unix only):** This option bypasses the need to change environment variables. Without this option, `TEX Live` directories have to be added to `PATH`, `MANPATH` and `INFOPATH`. You will need write permissions to the target directories. It is not advisable to overwrite a `TEX` system that came with your system with this option. It is intended primarily for creating the links in standard directories that are known to users, such as `/usr/local/bin`, which don't already contain any `TEX` files.

Now when all these settings are to your liking, you can type 'I' to start the installation process. When it is done, skip to section 3.4 to read what else needs to be done, if anything.

### 3.2.5 Set up for running from DVD (text mode only)

Type 'V' to select this option. This changes the main menu into something like figure 7.

Note the changes: all options about what to install have disappeared, and the directories section now talks about `TEXDIRW` or writable root. The symlinks option has also disappeared.

The installer will still create various directories and configuration files, but won't copy `texmf`, `texmf-dist` or `texmf-doc` to hard disk.

Post-install configuration for Unix will be slightly more complicated, because now the directory layout deviates from the default; see section 3.4.

Due to time constraints, this option didn't make it into the GUI installer, but it is available both for Unix and for Windows. Windows users have to start the installer from a command prompt, see section 3.3.

Section 5 describes a more strictly portable way to run `TEX Live`, which doesn't make or require changes in the system's configuration, but doesn't allow any configuration either.

```

===== TeX Live installation procedure =====
===== Note: Letters/digits in <angle brackets> indicate menu items =====
===== for commands or configurable options =====

<D> directories:
  TEXDIRW (Writable root):
    !! default location: /usr/local/texlive/2008
    !! is not writable, please select a different one!
  TEXMFLOCAL (directory for site-wide local files):
    /usr/local/texlive/texmf-local
  TEXMFSYSVAR (directory for variable and automatically generated data):
    /usr/local/texlive/2008/texmf-var
  TEXMFSYSCONFIG (directory for local config):
    /usr/local/texlive/2008/texmf-config
  TEXMFHOME (directory for user-specific files):
    ~/texmf

<O> options:
  [ ] use letter size instead of A4 by default
  [X] create all format files

<V> set up for installing to hard disk

Other actions:
<I> start installation for running from DVD
<H> help
<Q> quit

```

Figure 7: The main menu with `from DVD` set

### 3.3 Command-line install options

Type

```
> install-tl -help
```

for a listing of command-line options. Either `-` or `--` can be used to introduce option names. These are the more interesting ones:

- `-gui` If possible, use the GUI installer. This requires the Perl/Tk module (<http://tug.org/texlive/distro.html#perlTk>); if Perl/Tk is not available, installation continues in text mode.
- `-no-gui` Force using the text mode installer, even under Windows; you'll need this if you want a 'from DVD' installation, since that is not available in GUI mode.
- `-lang LL` Specify the installer interface language as its standard two-letter code *LL*. Currently supported languages: English (**en**, default), German (**de**), French (**fr**), Dutch (**nl**), Polish (**pl**), Slovenian (**sl**) and Vietnamese (**vi**). The installer tries to determine the right language itself but if it fails, or if the right language is not available, then it uses English as a fallback.
- `-profile profile` The installer always writes a file `texlive.profile` to the `2008/tlpkg-` or `tlpkg` subdirectory of your installation. This option tells the installer to re-use such a profile file. Here's an excerpt:

```

# texlive.profile written on Wed Jul  9 16:13:44 2008 UTC
# It will NOT be updated and reflects only the
# installation profile at installation time.
selected_scheme scheme-basic
TEXDIR /usr/local/texlive/2008
TEXDIRW /usr/local/texlive/2008
...
TEXMFSYSVAR /usr/local/texlive/2008/texmf-var
collection-basic 1

```

```
...
option_symlinks 0
```

Thus, using this option you can install in batch mode on subsequent systems, reproducing the choices you made for the original installation.

**-location** *url/path* Specify a non-default source for packages; see below.

### 3.3.1 The location parameter

The location value can be a url starting with **ftp:**, **http:**, or **file:/** url, or a plain path. If the given argument is on a local drive (either a path or a **file:/** url), the installation type (compressed or live) is automatically determined: if an **archive** subdirectory with **.tar.lzma** files is found then it will be used, even if uncompressed files are available as well.

When giving an **http:** or **ftp:** location, trailing **'/** characters and/or a trailing **'/t1pkg'** component are ignored. For example, you could choose a particular CTAN mirror with something like: <http://ctan.example.org/tex-archive/texlive/tlnet/2008>, substituting of course a proper hostname and its particular top-level CTAN path (the list of CTAN mirrors is maintained at <http://ctan.org/mirrors>) for [ctan.example.org/tex-archive](http://ctan.example.org/tex-archive). The default is to pick a mirror automatically, using <http://mirror.ctan.org>; in any case, the chosen host is used for the entire download.

## 3.4 Post-install actions

Some post-install may be required.

### 3.4.1 Windows

But on Windows, the installer takes care of everything.

### 3.4.2 If symlinks were created

If you elected to create symlinks in standard directories (mentioned in section 3.2.4), then there is no need to edit environment variables.

### 3.4.3 Environment variables for Unix

The directory of the binaries for your platform must be added to the search path. Each supported platform has its own subdirectory under **TEXDIR/bin**. See figure 3 for the list of subdirectories and corresponding platforms.

You can also add the documentation man and Info directories to their respective search paths, if you want the system tools to find them.

For Bourne-compatible shells such as **bash**, and using Intel x86 GNU/Linux and a default directory setup as an example, the file to edit might be **\$HOME/.profile** (or another file sourced by **.profile**, and the lines to add would look like this:

```
PATH=/usr/local/texlive/2008/bin/i386-linux:$PATH; export PATH
MANPATH=/usr/local/texlive/2008/texmf/doc/man:$MANPATH; export MANPATH
INFOPATH=/usr/local/texlive/2008/texmf/doc/info:$INFOPATH; export INFOPATH
```

For **csh** or **tcsh**, the file to edit is typically **\$HOME/.cshrc**, and the lines to add might look like:

```
setenv PATH /usr/local/texlive/2008/bin/i386-linux:$PATH
setenv MANPATH /usr/local/texlive/2008/texmf/doc/man:$MANPATH
setenv INFOPATH /usr/local/texlive/2008/texmf/doc/info:$INFOPATH
```

Of course, you may well already have settings somewhere in your “dot” files, and so then the T<sub>E</sub>X Live directories should simply be merged in as appropriate.

### 3.4.4 Environment variables: global configuration

If you want to make these changes globally, or for a user newly added to the system, then you are on your own; there is just too much variation between systems in how and where these things are configured.

Our two hints are: 1) you may want to check for a file `/etc/manpath.config` and, if present, add lines such as

```
MANPATH_MAP /usr/local/texlive/2008/bin/i386-linux \
            /usr/local/texlive/2008/texmf/doc/man
```

And 2) check for a file `/etc/environment` which may define the search path and other default environment variables.

### 3.4.5 When running from DVD

Normally, a T<sub>E</sub>X Live program consults a file `texmf.cnf` for the location of the various trees. It looks for this file in a series of locations relative to its own location. However, this scheme breaks down when a program is run from DVD: the DVD is read-only. Some of the paths to be recorded in `texmf.cnf` are only known at installation time, so this file cannot be on the DVD and must be placed somewhere else. This makes it necessary to define an environment `TEXMFCONF` variable which tells T<sub>E</sub>X Live programs in what directory to find this `texmf.cnf`. It is also still necessary to modify the `PATH` environment variable, as described before.

At the end of the installation, the installer should have printed a message giving the value to which `TEXMFCONF` should be set. In case you missed it: this value is `$TEXMFSYSVAR/web2c`. For the default, `/usr/local/texlive/2008/texmf-var/web2c`, you need the lines

```
TEXMFCONF=/usr/local/texlive/2008/texmf-var/web2c; export TEXMFCONF
```

or, for `[t]csh`:

```
setenv TEXMFCONF /usr/local/texlive/2008/texmf-var/web2c
```

### 3.4.6 ConT<sub>E</sub>Xt Mark IV

The ‘old’ ConT<sub>E</sub>Xt should run out of the box. The new ‘Mark IV’ ConT<sub>E</sub>Xt will require manual setup; see [http://wiki.contextgarden.net/Running\\_Mark\\_IV](http://wiki.contextgarden.net/Running_Mark_IV).

### 3.4.7 Integrating local and personal macros

This is already mentioned implicitly in section 2.3: `TEXMFLOCAL` (by default, `/usr/local/texlive/texmf-local` or `C:\Program Files\texlive\texmf-local`)

is intended for system-wide local fonts and macros; and `TEXMFHOME` (by default, `$HOME/texmf` or `%USERPROFILE%\texmf`), is for personal fonts and macros. These directories are intended to stick around from release to release, and have their content seen automatically by a new T<sub>E</sub>X Live release. Therefore, it is best to refrain from changing the definition of `TEXMFLOCAL` to be too far away from the main T<sub>E</sub>X Live directory, or you will need to manually change future releases.

For both trees, files should be placed in their proper subdirectories; see <http://tug.org/tds> or consult `texmf/web2c/texmf.cnf`. For instance, a L<sup>A</sup>T<sub>E</sub>X class file or package should be placed in `TEXMFLOCAL/tex/latex` or `TEXMFHOME/tex/latex`, or a subdirectory thereof.

`TEXMFLOCAL` requires an up-to-date filename database, or files will not be found. You can update it with the command `mktextlsr` or use the ‘Reinit file database’ button on the configuration tab of `tlmgr` in GUI mode.



### 3.4.8 Integrating third-party fonts

This is unfortunately a very complicated and messy topic. Forget about it unless you really know your way around blindfolded in a T<sub>E</sub>X installation. Don't forget to check first what you get for free: see section 2.6.

A possible alternative is to use XeT<sub>E</sub>X (see section 2.4), which lets you use operating system fonts without any installation in T<sub>E</sub>X.

## 3.5 Testing the installation

After installing T<sub>E</sub>X Live as best you can, you naturally want to test it out, so you can start creating beautiful documents and/or fonts.

This section gives some basic procedures for testing that the new system is functional. We give Unix commands here; under Mac OS X and Windows, you're more likely to run the tests through a graphical interface, but the principles are the same.

1. Make sure that you can run the `tex` program in the first place:

```
> tex --version
TeX 3.1415926 (Web2C 7.5.7)
kpathsea version 3.5.7
...
```

If this comes back with 'command not found' instead of version and copyright information, or with an older version, most likely you don't have the correct `bin` subdirectory in your `PATH`. See the environment-setting information on p. 15.

2. Process a basic L<sup>A</sup>T<sub>E</sub>X file:

```
> latex sample2e.tex
This is pdfTeXk, Version 3.1415926...
...
Output written on sample2e.dvi (3 pages, 7484 bytes).
Transcript written on sample2e.log.
```

If this fails to find `sample2e.tex` or other files, most likely you have interference from old environment variables or configuration files; we recommend unsetting all T<sub>E</sub>X-related environment variables for a start. (For a deep analysis, you can ask T<sub>E</sub>X to report on exactly what it is searching for, and finding; see "Debugging actions" on page 32.)

3. Preview the result online:

```
> xdvi sample2e.dvi
```

(Under Windows, the analogous command is `dviout`.) You should see a new window with a nice document explaining some of the basics of L<sup>A</sup>T<sub>E</sub>X. (Well worth reading, by the way if you're new to the system.) You do have to be running under X for `xdvi` to work; if you're not, or your `DISPLAY` environment variable is set incorrectly, you'll get an error 'Can't open display'.

4. Create a PostScript file for printing or display:

```
> dvips sample2e.dvi -o sample2e.ps
```

5. Create a PDF file instead of DVI; this processes the `.tex` file and writes PDF directly:

```
> pdflatex sample2e.tex
```

6. Preview the PDF file:

```
> gv sample2e.pdf
or:
> xpdf sample2e.pdf
```

Unfortunately neither `gv` nor `xpdf` are currently included in T<sub>E</sub>X Live, so you must install them separately. See <http://www.gnu.org/software/gv> and <http://www.foolabs.com/xpdf>, respectively. (There are plenty of other PDF viewers, too.)

7. Standard test files you may find useful in addition to `sample2e.tex`:

`small2e.tex` A simpler document than `sample2e`, to reduce the input size if you're having troubles.

`testpage.tex` Test if your printer introduces any offsets.

`nfssfont.tex` For printing font tables and tests.

`testfont.tex` Also for font tables, but using plain T<sub>E</sub>X.

`story.tex` The most canonical (plain) T<sub>E</sub>X test file of all. You must type ‘`\bye`’ to the `*` prompt after ‘`tex story.tex`’.

If you are new to T<sub>E</sub>X, or otherwise need help with actually writing T<sub>E</sub>X or L<sup>A</sup>T<sub>E</sub>X documents, please visit <http://tug.org/begin.html> for some introductory resources.

## 4 Network installations

T<sub>E</sub>X Live has been designed to be shared between different users, and even different systems on a network. With a standard directory layout, no hard paths are configured: the locations for files needed by T<sub>E</sub>X Live programs are defined relative to the places of those programs. You can see this at work in the configuration file `$TEXMFMAIN/web2c/texmf.cnf`, which contains lines such as

```
TEXMFMAIN = $SELFAUTOPARENT/texmf
...
TEXMFLOCAL = $SELFAUTOPARENT/./texmf-local
```

This means that other systems or other users merely need to add the directory for T<sub>E</sub>X Live binaries for their platform to their search path.

By the same token, you can also install T<sub>E</sub>X Live locally and then move the entire hierarchy afterwards to a network location.

Of course, Windows users have different expectations. The directory `texmf-doc/source/texlive/texlive-common/w32client` contains a sample script `install-w32client` which creates settings and menu shortcuts for using an existing T<sub>E</sub>X Live installation on a LAN. It is to be invoked through a batch file `install-w32client.bat` in the same directory.

You can read more about T<sub>E</sub>X Live configuration and path searching in section 8.1.

## 5 Maximally portable T<sub>E</sub>X Live on DVD and USB

The ‘running from DVD’ option described in section 3.2.5 is fine for your own system, but if you are a guest on somebody else’s system then you would probably like something with minimal side effects.

In the root of the T<sub>E</sub>X Live DVD, or the `texlive` subdirectory of the T<sub>E</sub>X Collection DVD, are a `tl-portable` script (Unix) and a `tl-portable.bat` batchfile (Windows) which start up a secondary shell/command prompt with environment settings suitable for directly accessing the T<sub>E</sub>X Live on the DVD.

When it runs for the first time, some files will be generated in a directory `~/tlportable2008`, which will take a little time. On subsequent runs, though, it will start almost instantaneously.

The rest of the system will be unaware of T<sub>E</sub>X Live. If you want your editor to be aware of this T<sub>E</sub>X Live, then you can start it from a second, parallel, such `tl-portable` session.

You can also use `tl-portable` to run T<sub>E</sub>X Live on a USB stick. In this case, copy all the top-level files and the contents of the `bin`, `texmf`, `texmf-dist`, and `tlpkg` directories (at least) to the stick. This may take quite a while! If you are copying to a FAT32-formatted USB stick, make sure to dereference symbolic links (`cp -L`). A `texmf-local` directory on the stick will also be found and used.

Then run `tl-portable` from the root of the stick, as above. In this case, the script will notice that the stick is writable and use it for generated files. You could burn the resulting contents of the stick back to DVD if that is more convenient to (for example) give to others.

## 6 tlmgr: Managing your installation

The installer installs a program named `tlmgr` for managing T<sub>E</sub>X Live after the initial installation. The programs `updmap`, `fmtutil` and `texconfig` are still included and will be retained in the future, but `tlmgr` is now the preferred interface. Its capabilities include:

- listing schemes, collections and packages;
- installing, updating, backing up, restoring and uninstalling individual packages, optionally taking dependencies into account;
- searching for packages;
- listing and adding architectures;
- changing installation options such as paper size and source location (see section 3.3.1).

*Warning:* `tlmgr` has not been designed for or tested with installations which run from DVD.

### 6.1 tlmgr GUI mode

`tlmgr` can be started in GUI mode with

```
> tlmgr -gui
```

or in Windows via the Start menu: **Start, Programs, TeX Live 2008, TeX Live Manager**. After clicking ‘Load’ it displays a list of available and installed packages—the latter prepended with ‘(i)’. This assumes of course that the installation source is valid and reachable.

Figure 9 shows the configuration tab.

### 6.2 Sample tlmgr command-line invocations

This example adds several packages, including a new ‘engine’, from a non-default location:

```
> tlmgr -location http://172.16.18.1/tldev install collection-xetex xkeyval
```

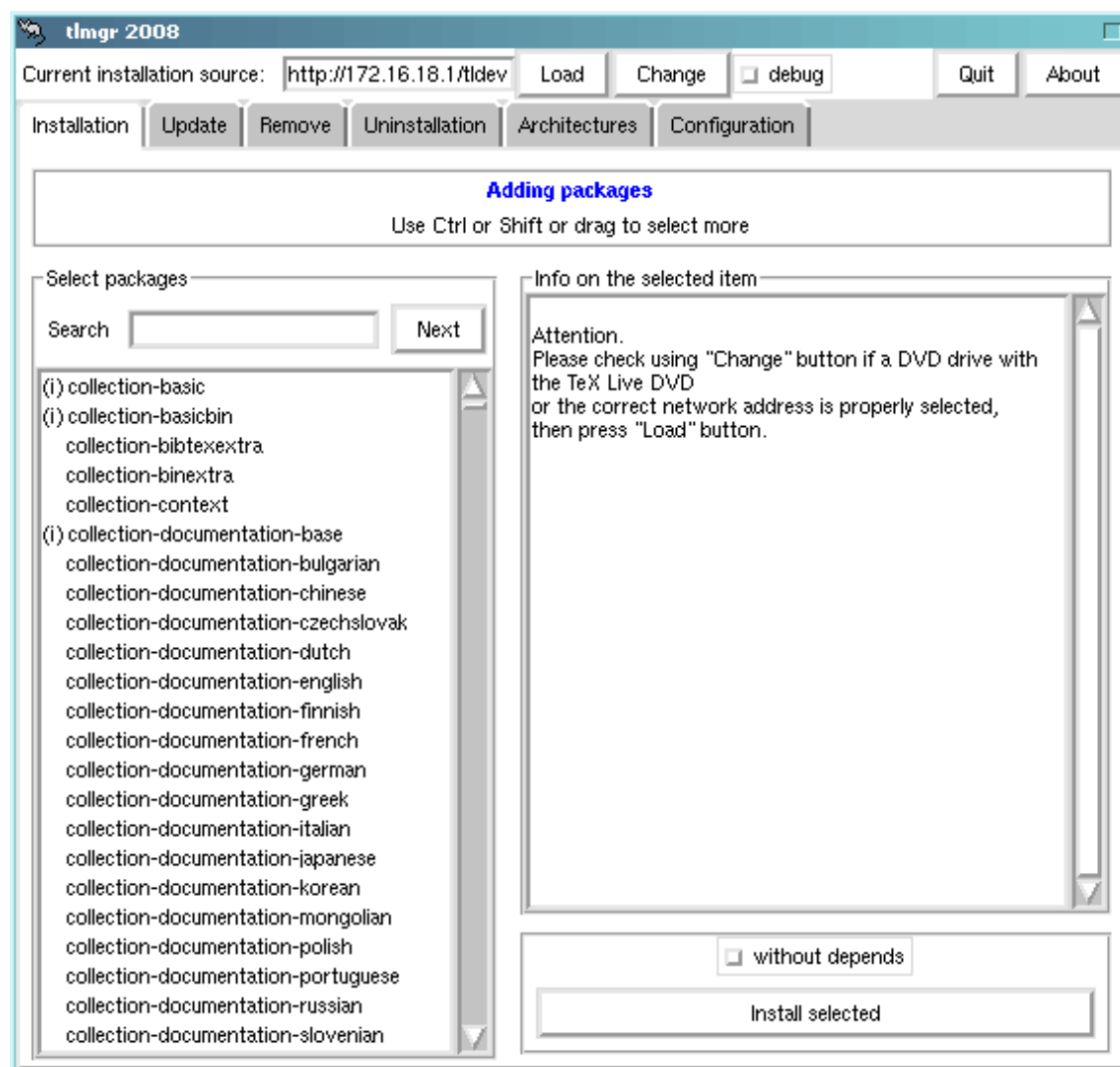


Figure 8: tlmgr in GUI mode. The list of packages/collections/schemes only shows up after clicking the ‘Load’ button.

It generates the following output:

```
install: collection-xetex
install: arabxetex
install: euenc
install: bidi
install: fontspec
install: fontwrap
install: ifxetex
install: philokalia
install: polyglossia
install: xecyr
install: xecyr.i386-linux
install: xepersian
install: xetex
install: xetexconfig
install: xetex.i386-linux
running post install action for xetex
```

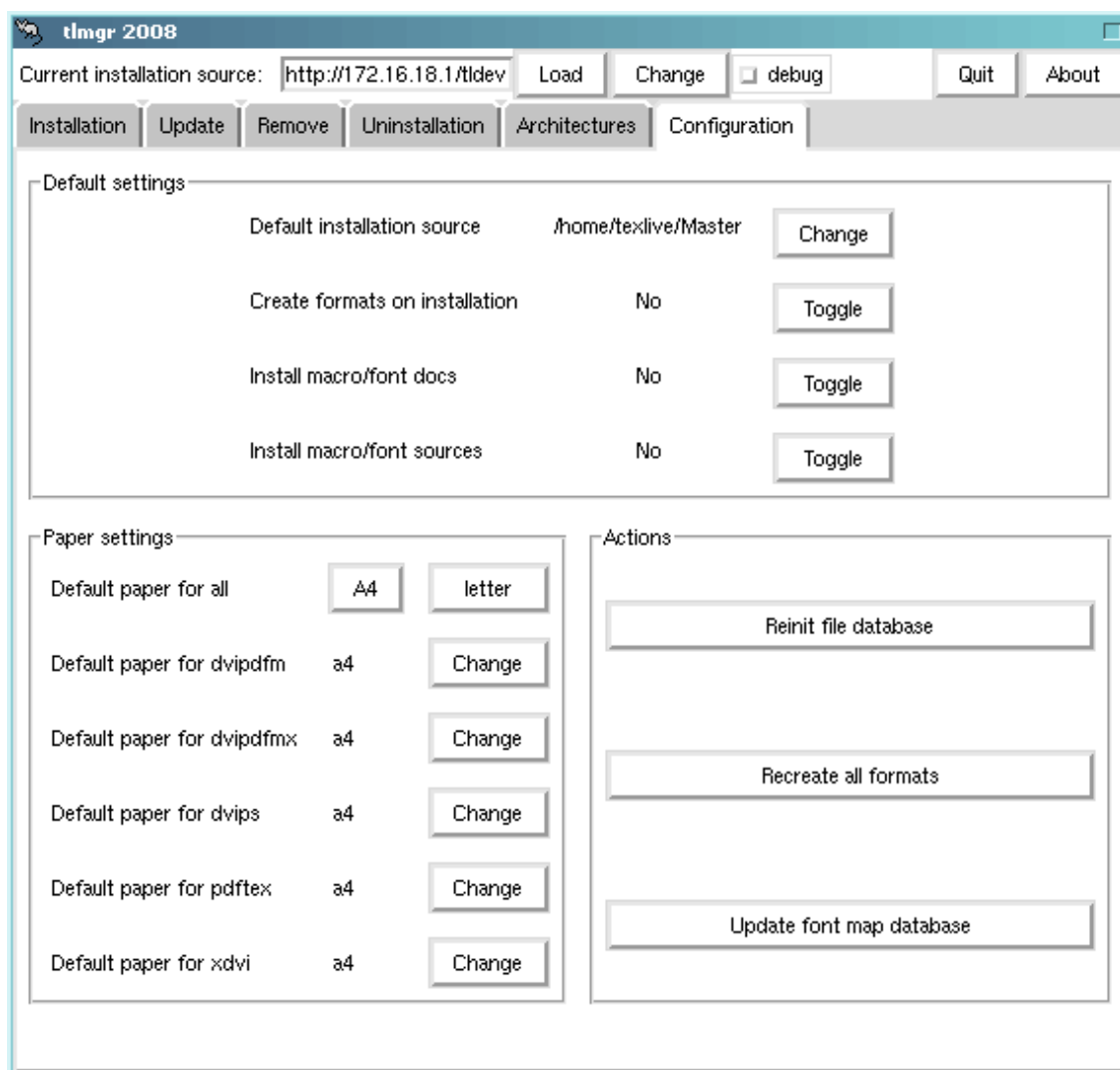


Figure 9: tlmgr in GUI mode: Configuration tab

```

install: xetex-def
install: xetex-pstricks
install: xetexfontinfo
install: xltextra
install: xunicode
install: xkeyval
running mktexlsr
mktexlsr: Updating /usr/local/texlive/2008/texmf/ls-R...
mktexlsr: Updating /usr/local/texlive/2008/texmf-config/ls-R...
mktexlsr: Updating /usr/local/texlive/2008/texmf-dist/ls-R...
mktexlsr: Updating /usr/local/texlive/2008/texmf-doc/ls-R...
mktexlsr: Updating /usr/local/texlive/2008/./texmf-local/ls-R...
mktexlsr: Updating /usr/local/texlive/2008/texmf-var/ls-R...
mktexlsr: Done.
regenerating fntutil.cnf
running fntutil-sys --missing
...
Transcript written on xelatex.log.

```

```
fmtutil: /usr/local/texlive/2008/texmf-var/web2c/xetex/xelatex.fmt installed.
```

As you see, `tlmgr` installs dependencies, and takes care of any necessary post-install actions, including updating the filename database and (re)generating formats. (Here we generated new formats for XeTeX.)

This example installs a new language. This also triggers format regeneration:

```
> tlmgr install collection-langdutch
```

To describe a package:

```
> tlmgr show collection-wintools
```

which produces

```
Package:    collection-wintools
Category:   Collection
ShortDesc:  Windows support programs
LongDesc:   Utilities for Windows, since they are not readily available there:
            gzip, chktex, jpeg2ps, unzip, wget, xpdf.
Installed:  Yes
```

Updating your entire installation to the latest versions available:

```
> tlmgr update -all
```

If this makes you nervous, first try

```
> tlmgr update -all -dry-run
```

or

```
> tlmgr update -list
```

For full documentation:

```
> tlmgr -help
```

## 7 Notes on Windows

TeX Live has a single installer which runs on both Windows and Unix. This was only possible by dropping support for older Windows versions, so TeX Live can now be installed only on Windows 2000 and later.

### 7.1 Windows-specific features

Under Windows, the installer does some extra things:

**Menus and shortcuts.** A new ‘TeX Live’ submenu of the Start menu is installed, which contains entries for some GUI programs (`tlmgr`, `texdoctk`, the `PS_View` PostScript previewer) and some documentation. `PS_View` also gets a shortcut on the desktop as a drag-and-drop target for PostScript files.

**Automatic setting of environment variables.** No manual configuration steps are required.

**Uninstaller.** The installer creates an entry under ‘Add/Remove Programs’ for TeX Live. The uninstall tab of `tlmgr` refers to this.

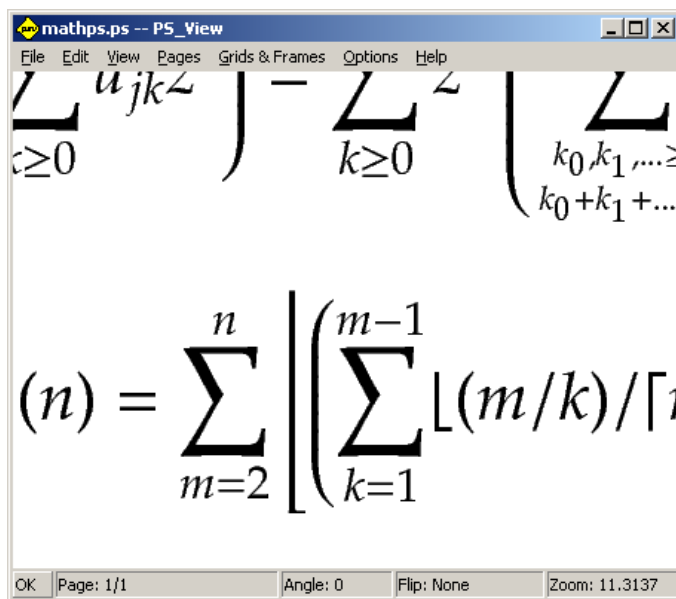


Figure 10: PS\_View: very high magnifications available!

## 7.2 Additional included software

To be complete, a T<sub>E</sub>X Live installation needs support packages that are not commonly found on a Windows machine. T<sub>E</sub>X Live provides the missing pieces:

**Perl and Ghostscript.** Because of the importance of Perl and Ghostscript, T<sub>E</sub>X Live includes ‘hidden’ copies of these programs. T<sub>E</sub>X Live programs that need them know where to find them, but they don’t betray their presence through environment variables or registry settings. They aren’t full-scale distributions, and shouldn’t interfere with any system installations of Perl or Ghostscript.

**Command-line tools.** A number of Windows ports of common Unix command-line programs are installed along with the usual T<sub>E</sub>X Live binaries. These include `gzip`, `chktex`, `jpeg2ps`, `unzip`, `wget` and the command-line utilities from the `xpdf` suite. (The `xpdf` viewer itself is not available for Windows, but the Sumatra PDF viewer is based on it: <http://blog.kowalczyk.info/software/sumatrapdf>.)

**fc-cache** helps XeT<sub>E</sub>X to handle fonts more efficiently.

**PS\_View.** Also installed is PS\_View, a new PostScript viewer with a free license; see figure 10.

**The support directory,** already mentioned in section 2.2. This directory contains various programs which can be useful in connection with T<sub>E</sub>X, in particular a number of T<sub>E</sub>X-aware editors and T<sub>E</sub>X frontends. The software in this directory isn’t covered by the installer. This directory also contains the `dviout` DVI previewer; see the next section.

## 7.3 The Dviout previewer

This T<sub>E</sub>X Live release unfortunately doesn’t offer a preinstalled DVI previewer for Windows. However, you can install `dviout` by Toshio Oshima from the `support/dviout/` directory, which contains also a short README describing how to install and configure `dviout` for T<sub>E</sub>X Live. Please read this carefully before installation.

At first, when you preview files with `dviout`, it will create fonts, because screen fonts were not installed. After a while, you will have created most of the fonts you use, and you will rarely see the font-creation window. More information can be found in the (very good) on-line help.

## 7.4 Links for additional downloadable software

Urls for some other tools you may consider installing:

**Ghostscript** <http://www.cs.wisc.edu/~ghost/>

**Perl** <http://www.activestate.com/> with supplementary packages from CPAN, <http://www.cpan.org/>

**ImageMagick** <http://www.imagemagick.com>, for graphics processing and conversion

**NetPBM** <http://netpbm.sourceforge.net/>, also for graphics.

**T<sub>E</sub>X-oriented editors** There is a wide choice, and it is a matter of the user's taste. Here is a selection, in no particular order:

- GNU Emacs is available natively under Windows, see <http://www.gnu.org/software/emacs/windows/ntemacs.html>.
- Emacs with AucT<sub>E</sub>X for Windows is available in T<sub>E</sub>X Live's support directory; its home page is <http://www.gnu.org/software/auctex>.
- WinShell is available in T<sub>E</sub>X Live's support directory; its home page is <http://www.winshell.de>.
- WinEdt is shareware available from <http://www.winedt.com>.
- Vim is free software, available from <http://www.vim.org>.
- TeXnicCenter is free software, available from <http://www.toolscenter.org> and in the proT<sub>E</sub>Xt distribution.
- LEd is available from <http://www.ctan.org/support/LEd>.
- SciTE is available from <http://www.scintilla.org/SciTE.html>.

## 7.5 User Profile is Home

The Windows counterpart of a Unix home directory is the `%USERPROFILE%` directory. Under Windows XP and Windows 2000, this is usually `C:/Documents and Settings/<username>`, and under Windows Vista `C:/Users/<username>`. In the `texmf.cnf` file, and Kpathsea in general, `~` will expand appropriately on both Windows and Unix.

## 7.6 The Windows registry

Windows stores nearly all configuration data in its registry. The registry contains a set of hierarchically organized keys, with several root keys. The most important ones for installation programs are `HKEY_CURRENT_USER` and `HKEY_LOCAL_MACHINE`, `HKCU` and `HKLM` in short. The `HKCU` part of the registry is in the user's home directory (see section 7.5). `HKLM` is normally in a subdirectory of the Windows directory.

In some cases, system information could be obtained from environment variables but for other information, for example the location of shortcuts, it is necessary to consult the registry. Setting environment variables permanently also requires registry access.



## 7.7 Windows permissions

In later versions of Windows, a distinction is made between regular users and administrators, where only the latter have free access to the entire operating system. In practice, though, you could better describe these classes of users as unprivileged users and normal users: being an administrator is the rule, not the exception. Nevertheless, we have made an effort to make T<sub>E</sub>X Live installable without administrative privileges.

If the user is an administrator, shortcuts are created for all users, and the system environment is modified. Otherwise, shortcuts and menu entries are created for the current user, and the user environment is modified. Use the `non-admin` option to `install-tl` if you prefer per-user settings even as an administrator.

Regardless of administrator status, the default root of T<sub>E</sub>X Live proposed by the installer is always under `%PROGRAMFILES%`. The installer always tests whether the root is writable for the current user.

A problem may arise if the user is not an administrator and T<sub>E</sub>X already exists in the search path. Since the effective path consists of the system path followed by the user path, the new T<sub>E</sub>X Live will never get precedence. As a workaround, in this case the installer creates shortcuts to the command-prompt in which the new T<sub>E</sub>X Live binary directory is prepended to the local searchpath. The new T<sub>E</sub>X Live will be usable when run in a command-line session started from such a shortcut.

## 8 A user's guide to Web2C

Web2C is an integrated collection of T<sub>E</sub>X-related programs: T<sub>E</sub>X itself, METAFONT, MetaPost, BibT<sub>E</sub>X, etc. It is the heart of T<sub>E</sub>X Live. The home page for Web2C, with the current manual and more, is <http://tug.org/web2c>.

A bit of history: The original implementation was by Tomas Rokicki who, in 1987, developed a first T<sub>E</sub>X-to-C system based on change files under Unix, which were primarily the original work of Howard Trickey and Pavel Curtis. Tim Morgan became the maintainer of the system, and during this period the name changed to Web-to-C. In 1990, Karl Berry took over the work, assisted by dozens of additional contributors, and in 1997 he handed the baton to Olaf Weber, who returned it to Karl in 2006.

The Web2C system runs on Unix, 32-bit Windows systems, Mac OS X, and other operating systems. It uses Knuth's original sources for T<sub>E</sub>X and other basic programs written in the WEB literate programming system and translates them into C source code. The core T<sub>E</sub>X programs handled in this way are:

`bibtex` Maintaining bibliographies.

`dmp troff` to MPX (MetaPost pictures).

`dvicopy` Expands virtual font references in DVI files.

`dvitomp` DVI to MPX (MetaPost pictures).

`dvitype` DVI to human-readable text.

`gftodvi` Generic font proofsheets.

`gftopk` Generic to packed fonts.

`gftype` GF to human-readable text.

`makempx` MetaPost label typesetting.

`mf` Creating typeface families.

`mft` Prettyprinting METAFONT source.

`mpost` Creating technical diagrams.  
`mpto` MetaPost label extraction.  
`newer` Compare modification times.  
`patgen` Creating hyphenation patterns.  
`pktogf` Packed to generic fonts.  
`pktype` PK to human-readable text.  
`pltotf` Plain text property list to TFM.  
`pooltype` Display WEB pool files.  
`tangle` WEB to Pascal.  
`tex` Typesetting.  
`tftopl` TFM to plain text property list.  
`vftovp` Virtual font to virtual property list.  
`vptovf` Virtual property list to virtual font.  
`weave` WEB to T<sub>E</sub>X.

The precise functions and syntax of these programs are described in the documentation of the individual packages and of Web2C itself. However, knowing a few principles governing the whole family of programs will help you take advantage of your Web2C installation.

All programs honor these standard GNU options:

`--help` print basic usage summary.  
`--verbose` print detailed progress report.  
`--version` print version information, then exit.

For locating files the Web2C programs use the path searching library Kpathsea (<http://tug.org/kpathsea>). This library uses a combination of environment variables and a configuration files to optimize searching the (huge) collection of T<sub>E</sub>X files. Web2C can look at many directory trees simultaneously, which is useful in maintaining T<sub>E</sub>X's standard distribution and local and personal extensions in distinct trees. To speed up file searches, the root of each tree has a file `ls-R`, containing an entry showing the name and relative pathname for all files under that root.

## 8.1 Kpathsea path searching

Let us first describe the generic path searching mechanism of the Kpathsea library.

We call a *search path* a colon- or semicolon-separated list of *path elements*, which are basically directory names. A search path can come from (a combination of) many sources. To look up a file `'my-file'` along a path `'./dir'`, Kpathsea checks each element of the path in turn: first `./my-file`, then `/dir/my-file`, returning the first match (or possibly all matches).

In order to adapt optimally to all operating systems' conventions, on non-Unix systems Kpathsea can use filename separators different from colon (':') and slash ('/').

To check a particular path element *p*, Kpathsea first checks if a prebuilt database (see "Filename database" on page 29) applies to *p*, i.e., if the database is in a directory that is a prefix of *p*. If so, the path specification is matched against the contents of the database.

If the database does not exist, or does not apply to this path element, or contains no matches, the filesystem is searched (if this was not forbidden by a specification starting with '!!' and if the file being searched for must exist). Kpathsea constructs the list of directories that correspond to this path element, and then checks in each for the file being sought.

The “file must exist” condition comes into play with ‘.vf’ files and input files read by T<sub>E</sub>X’s `\openin` command. Such files may not exist (e.g., `cmr10.vf`), and so it would be wrong to search the disk for them. Therefore, if you fail to update `ls-R` when you install a new ‘.vf’ file, it will never be found. Each path element is checked in turn: first the database, then the disk. If a match is found, the search stops and the result is returned.

Although the simplest and most common path element is a directory name, Kpathsea supports additional features in search paths: layered default values, environment variable names, config file values, users’ home directories, and recursive subdirectory searching. Thus, we say that Kpathsea *expands* a path element, meaning it transforms all the specifications into basic directory name or names. This is described in the following sections in the same order as it takes place.

Note that if the filename being searched for is absolute or explicitly relative, i.e., starts with ‘/’ or ‘./’ or ‘../’, Kpathsea simply checks if that file exists.

### 8.1.1 Path sources

A search path can come from many sources. In the order in which Kpathsea uses them:

1. A user-set environment variable, for instance, `TEXINPUTS`. Environment variables with a period and a program name appended override; e.g., if ‘`latex`’ is the name of the program being run, then `TEXINPUTS.latex` will override `TEXINPUTS`.
2. A program-specific configuration file, for example, a line ‘`S /a:/b`’ in dvips’s `config.ps`.
3. A Kpathsea configuration file `texmf.cnf`, containing a line like ‘`TEXINPUTS=/c:/d`’ (see below).
4. The compile-time default.

You can see each of these values for a given search path by using the debugging options (see “Debugging actions” on page 32).

### 8.1.2 Config files

Kpathsea reads *runtime configuration files* named `texmf.cnf` for search path and other definitions. The search path used to look for these files is named `TEXMFCNF` (by default such a file lives in the `texmf/web2c` subdirectory). All `texmf.cnf` files in the search path will be read and definitions in earlier files override those in later files. Thus, with a search path of `.: $\$$ TEXMF`, values from `./texmf.cnf` override those from  `$\$$ TEXMF/texmf.cnf`.

- Comments start with `%` and continue to the end of the line.
- Blank lines are ignored.
- A `\` at the end of a line acts as a continuation character, i.e., the next line is appended. Whitespace at the beginning of continuation lines is not ignored.
- Each remaining line has the form:

```
variable[.programe] [=] value
```

where the ‘=’ and surrounding whitespace are optional.

- The *variable* name may contain any character other than whitespace, ‘=’, or ‘.’, but sticking to ‘`A-Za-z_`’ is safest.

- If `‘.progname’` is present, the definition only applies if the program that is running is named `progname` or `progname.exe`. This allows different flavors of T<sub>E</sub>X to have different search paths, for example.
- `value` may contain any characters except `%` and `@`. The `$var.prog` feature is not available on the right-hand side; instead, you must use an additional variable. A `‘;’` in `value` is translated to `‘:’` if running under Unix; this is useful to be able to have a single `texmf.cnf` for Unix, MS-DOS and Windows systems.
- All definitions are read before anything is expanded, so variables can be referenced before they are defined.

A configuration file fragment illustrating most of these points is shown below:

```
TEXMF          = {$TEXMFLOCAL,!!$TEXMFMAIN}
TEXINPUTS.latex = .;$TEXMF/tex/{latex,generic;}//
TEXINPUTS.fontinst = .;$TEXMF/tex//;$TEXMF/fonts/afm//
% e-TeX related files
TEXINPUTS.elatex = .;$TEXMF/{etex,tex}/{latex,generic;}//
TEXINPUTS.etex   = .;$TEXMF/{etex,tex}/{eplain,plain,generic;}//
```

### 8.1.3 Path expansion

Kpathsea recognizes certain special characters and constructions in search paths, similar to those available in Unix shells. As a general example, the complex path, `~$USER/{foo,bar}//baz`, expands to all subdirectories under directories `foo` and `bar` in `$USER`'s home directory that contain a directory or file `baz`. These expansions are explained in the sections below.

### 8.1.4 Default expansion

If the highest-priority search path (see “Path sources” on page 27) contains an *extra colon* (i.e., leading, trailing, or doubled), Kpathsea inserts at that point the next-highest-priority search path that is defined. If that inserted path has an extra colon, the same happens with the next highest. For example, given an environment variable setting

```
> setenv TEXINPUTS /home/karl:
```

and a `TEXINPUTS` value from `texmf.cnf` of

```
.: $TEXMF//tex
```

then the final value used for searching will be:

```
/home/karl:.: $TEXMF//tex
```

Since it would be useless to insert the default value in more than one place, Kpathsea changes only one extra `‘:’` and leaves any others in place. It checks first for a leading `‘:’`, then a trailing `‘:’`, then a doubled `‘:’`.

### 8.1.5 Brace expansion

A useful feature is brace expansion, which means that, for instance, `v{a,b}w` expands to `vaw:vbw`. Nesting is allowed. This is used to implement multiple T<sub>E</sub>X hierarchies, by assigning a brace list to `$TEXMF`. For example, in `texmf.cnf`, the following definition (approximately; there are actually even more trees) is made:

```
TEXMF = {$TEXMFHOME,$TEXMFLOCAL,!!$TEXMFVAR,!!$TEXMFMAIN}
```

Using this you can then write something like

```
TEXINPUTS = .;$TEXMF/tex//
```

which means that, after looking in the current directory, the `$TEXMFHOME/tex`, `$TEXMFLOCAL/tex`, `$TEXMFVAR/tex` and `$TEXMFMAIN/tex` trees *only* will be searched (the last two use using `ls-R` data base files). It is a convenient way for running two parallel `TEX` structures, one “frozen” (on a CD, for instance) and the other being continuously updated with new versions as they become available. By using the `$TEXMF` variable in all definitions, one is sure to always search the up-to-date tree first.

### 8.1.6 Subdirectory expansion

Two or more consecutive slashes in a path element following a directory *d* is replaced by all subdirectories of *d*: first those subdirectories directly under *d*, then the subsubdirectories under those, and so on. At each level, the order in which the directories are searched is *unspecified*.

If you specify any filename components after the ‘//’, only subdirectories with matching components are included. For example, ‘/a//b’ expands into directories /a/1/b, /a/2/b, /a/1/1/b, and so on, but not /a/b/c or /a/1.

Multiple ‘//’ constructs in a path are possible, but ‘//’ at the beginning of a path is ignored.

### 8.1.7 List of special characters and their meaning: a summary

The following list summarizes the special characters in Kpathsea configuration files.

- :      Separator in path specification; at the beginning or the end of a path it substitutes the default path expansion.
- ;      Separator on non-Unix systems (acts like :).
- \$      Variable expansion.
- ~      Represents the user’s home directory.
- {...}   Brace expansion.
- //     Subdirectory expansion (can occur anywhere in a path, except at its beginning).
- %      Start of comment.
- \      Continuation character (allows multi-line entries).
- !!     Search *only* database to locate file, *do not* search the disk.

## 8.2 Filename databases

Kpathsea goes to some lengths to minimize disk accesses for searches. Nevertheless, at installations with enough directories, searching each possible directory for a given file can take an excessively long time (this is especially true if many hundreds of font directories have to be traversed.) Therefore, Kpathsea can use an externally-built plain text “database” file named `ls-R` that maps files to directories, thus avoiding the need to exhaustively search the disk.

A second database file `aliases` allows you to give additional names to the files listed in `ls-R`. This can be helpful to conform to DOS 8.3 filename conventions in source files.

### 8.2.1 The filename database

As explained above, the name of the main filename database must be `ls-R`. You can put one at the root of each  $\text{\TeX}$  hierarchy in your installation that you wish to be searched (`$TEXMF` by default); most sites have only one hierarchy. Kpathsea looks for `ls-R` files along the `TEXMFDBS` path.

The recommended way to create and maintain '`ls-R`' is to run the `mktexlsr` script included with the distribution. It is invoked by the various '`mktex`'... scripts. In principle, this script just runs the command

```
cd /your/texmf/root && \ls -lLAR ./ >ls-R
```

presuming your system's `ls` produces the right output format (GNU `ls` is all right). To ensure that the database is always up-to-date, it is easiest to rebuild it regularly via `cron`, so that it is automatically updated when the installed files change, such as after installing or updating a  $\text{\LaTeX}$  package.

If a file is not found in the database, by default Kpathsea goes ahead and searches the disk. If a particular path element begins with '`!!`', however, *only* the database will be searched for that element, never the disk.

### 8.2.2 kpsewhich: Standalone path searching

The `kpsewhich` program exercises path searching independent of any particular application. This can be useful as a sort of `find` program to locate files in  $\text{\TeX}$  hierarchies (this is used heavily in the distributed '`mktex`'... scripts).

```
> kpsewhich option... filename...
```

The options specified in *option* start with either '`-`' or '`--`', and any unambiguous abbreviation is accepted.

Kpathsea looks up each non-option argument on the command line as a filename, and returns the first file found. There is no option to return all the files with a particular name (you can run the Unix '`find`' utility for that).

The most common options are described next.

`--dpi=num` Set the resolution to *num*; this only affects '`gf`' and '`pk`' lookups. '`-D`' is a synonym, for compatibility with `dvips`. Default is 600.

`--format=name`

Set the format for lookup to *name*. By default, the format is guessed from the filename. For formats which do not have an associated unambiguous suffix, such as MetaPost support files and `dvips` configuration files, you have to specify the name as known to Kpathsea, such as `tex` or `enc` files. Run `kpsewhich --help` for a list.

`--mode=string`

Set the mode name to *string*; this only affects '`gf`' and '`pk`' lookups. No default: any mode will be found.

`--must-exist`

Do everything possible to find the files, notably including searching the disk. By default, only the `ls-R` database is checked, in the interest of efficiency.

`--path=string`

Search along the path *string* (colon-separated as usual), instead of guessing the search path from the filename. '`/'` and all the usual expansions are supported. The options '`--path`' and '`--format`' are mutually exclusive.

**--progrname=*name***  
 Set the program name to *name*. This can affect the search paths via the *.progrname* feature. The default is **kpsewhich**.

**--show-path=*name***  
 shows the path used for file lookups of file type *name*. Either a filename extension (**.pk**, **.vf**, etc.) or a name can be used, just as with '**--format**' option.

**--debug=*num***  
 sets the debugging options to *num*.

### 8.2.3 Examples of use

Let us now have a look at Kpathsea in action. Here's a straightforward search:

```
> kpsewhich article.cls
/usr/local/texmf-dist/tex/latex/base/article.cls
```

We are looking for the file **article.cls**. Since the '**.cls**' suffix is unambiguous we do not need to specify that we want to look for a file of type **tex** (T<sub>E</sub>X source file directories). We find it in the subdirectory **tex/latex/base** below the '**texmf-dist**' T<sub>E</sub>X Live directory. Similarly, all of the following are found without problems thanks to their unambiguous suffix.

```
> kpsewhich array.sty
/usr/local/texmf-dist/tex/latex/tools/array.sty
> kpsewhich latin1.def
/usr/local/texmf-dist/tex/latex/base/latin1.def
> kpsewhich size10.clo
/usr/local/texmf-dist/tex/latex/base/size10.clo
> kpsewhich small2e.tex
/usr/local/texmf-dist/tex/latex/base/small2e.tex
> kpsewhich tugboat.bib
/usr/local/texmf-dist/bibtex/bib/beebe/tugboat.bib
```

By the way, that last is a BIB<sub>T</sub>E<sub>X</sub> bibliography database for *TUGBoat* articles.

```
> kpsewhich cmr10.pk
```

Font bitmap glyph files of type **.pk** are used by display programs like **dvips** and **xdvi**. Nothing is returned in this case since there are no pre-generated Computer Modern '**.pk**' files in T<sub>E</sub>X Live—the Type 1 variants are used by default.

```
> kpsewhich wsuipa10.pk
/usr/local/texmf-var/fonts/pk/ljfour/public/wsuipa/wsuipa10.600pk
```

For these fonts (a phonetic alphabet from the University of Washington) we had to generate '**.pk**' files, and since the default METAFONT mode on our installation is **ljfour** with a base resolution of 600 dpi (dots per inch), this instantiation is returned.

```
> kpsewhich -dpi=300 wsuipa10.pk
```

In this case, when specifying that we are interested in a resolution of 300 dpi (**-dpi=300**) we see that no such font is available on the system. A program like **dvips** or **xdvi** would go off and actually build the required **.pk** files using the script **mktexpk**.

Next we turn our attention to **dvips**'s header and configuration files. We first look at one of the commonly used files, the general prologue **tex.pro** for T<sub>E</sub>X support, before turning our attention

to the generic configuration file (`config.ps`) and the PostScript font map `psfonts.map`—as of 2004, map and encoding files have their own search paths and new location in `texmf` trees. As the `.ps` suffix is ambiguous we have to specify explicitly which type we are considering (`dvips config`) for the file `config.ps`.

```
> kpsewhich tex.pro
  /usr/local/texmf/dvips/base/tex.pro
> kpsewhich --format="dvips config" config.ps
  /usr/local/texmf/dvips/config/config.ps
> kpsewhich psfonts.map
  /usr/local/texmf/fonts/map/dvips/updmap/psfonts.map
```

We now take a closer look at the URW Times PostScript support files. The prefix for these in the standard font naming scheme is `'utm'`. The first file we look at is the configuration file, which contains the name of the map file:

```
> kpsewhich --format="dvips config" config.utm
  /usr/local/texmf-dist/dvips/psnfss/config.utm
```

The contents of that file is

```
p +utm.map
```

which points to the file `utm.map`, which we want to locate next.

```
> kpsewhich utm.map
  /usr/local/texmf-dist/fonts/map/dvips/times/utm.map
```

This map file defines the file names of the Type 1 PostScript fonts in the URW collection. Its contents look like (we only show part of the lines):

```
utmb8r  NimbusRomNo9L-Medi    ... <utmb8a.pfb
utmbi8r NimbusRomNo9L-MediItal... <utmbi8a.pfb
utmr8r  NimbusRomNo9L-Regu    ... <utmr8a.pfb
utmri8r NimbusRomNo9L-ReguItal... <utmri8a.pfb
utmb8r  NimbusRomNo9L-Medi    ... <utmb8a.pfb
utmro8r NimbusRomNo9L-Regu    ... <utmr8a.pfb
```

Let us, for instance, take the Times Roman instance `utmr8a.pfb` and find its position in the `texmf` directory tree with a search for Type 1 font files:

```
> kpsewhich utmr8a.pfb
  /usr/local/texmf-dist/fonts/type1/urw/times/utmr8a.pfb
```

It should be evident from these examples how you can easily locate the whereabouts of a given file. This is especially important if you suspect that the wrong version of a file is picked up somehow, since `kpsewhich` will show you the first file encountered.

### 8.2.4 Debugging actions

Sometimes it is necessary to investigate how a program resolves file references. To make this practical, Kpathsea offers various levels of debugging output:

- 1 `stat` calls (disk lookups). When running with an up-to-date `ls-R` database this should almost give no output.
- 2 References to hash tables (such as `ls-R` databases, map files, configuration files).



- 4 File open and close operations.
- 8 General path information for file types searched by Kpathsea. This is useful to find out where a particular path for the file was defined.
- 16 Directory list for each path element (only relevant for searches on disk).
- 32 File searches.
- 64 Variable values.

A value of `-1` will set all the above options; in practice, this is usually the most convenient.

Similarly, with the `dvips` program, by setting a combination of debug switches, one can follow in detail where files are being picked up from. Alternatively, when a file is not found, the debug trace shows in which directories the program looks for the given file, so that one can get an indication what the problem is.

Generally speaking, as most programs call the Kpathsea library internally, one can select a debug option by using the `KPATHSEA_DEBUG` environment variable, and setting it to (a combination of) values as described in the above list.

(Note for Windows users: it is not easy to redirect all messages to a file in this system. For diagnostic purposes you can temporarily `SET KPATHSEA_DEBUG_OUTPUT=err.log`).

Let us consider, as an example, a small  $\text{\LaTeX}$  source file, `hello-world.tex`, which contains the following input.

```
\documentclass{article}
\begin{document}
Hello World!
\end{document}
```

This little file only uses the font `cmr10`, so let us look at how `dvips` prepares the PostScript file (we want to use the Type 1 version of the Computer Modern fonts, hence the option `-Pcms`).

```
> dvips -d4100 hello-world -Pcms -o
```

In this case we have combined `dvips`'s debug class 4 (font paths) with Kpathsea's path element expansion (see `dvips` Reference Manual, [texmf/doc/html/dvips/dvips\\_toc.html](http://texmf/doc/html/dvips/dvips_toc.html)). The output (slightly rearranged) appears in Figure 11.

`dvips` starts by locating its working files. First, `texmf.cnf` is found, which gives the definitions of the search paths for the other files, then the file database `ls-R` (to optimize file searching) and the file `aliases`, which makes it possible to declare several names (e.g., a short DOS-like 8.3 and a more natural longer version) for the same file. Then `dvips` goes on to find the generic configuration file `config.ps` before looking for the customization file `.dvipsrc` (which, in this case is *not found*). Finally, `dvips` locates the config file for the Computer Modern PostScript fonts `config.cms` (this was initiated with the `-Pcms` option on the `dvips` command). This file contains the list of the map files which define the relation between the  $\text{\TeX}$ , PostScript and file system names of the fonts.

```
> more /usr/local/texmf/dvips/cms/config.cms
p +ams.map
p +cms.map
p +cmbkm.map
p +amsbkm.map
```

`dvips` thus goes on to find all these files, plus the generic map file `psfonts.map`, which is always loaded (it contains declarations for commonly used PostScript fonts; see the last part of Section 8.2.3 for more details about PostScript map file handling).

At this point `dvips` identifies itself to the user:

```
This is dvips(k) 5.92b Copyright 2002 Radical Eye Software (www.radicleye.com)
```

```

debug:start search(file=texmf.cnf, must_exist=1, find_all=1,
  path=./usr/local/bin/texlive:/usr/local/bin:
    /usr/local/bin/texmf/web2c:/usr/local:
    /usr/local/texmf/web2c:./././teTeX/TeX/texmf/web2c:).
kdebug:start search(file=ls-R, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(ls-R) => /usr/local/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(aliases) => /usr/local/texmf/aliases
kdebug:start search(file=config.ps, must_exist=0, find_all=0,
  path=./tex:/usr/local/texmf/dvips/).
kdebug:search(config.ps) => /usr/local/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc, must_exist=0, find_all=0,
  path=./tex:/usr/local/texmf/dvips/).
search(file=/home/goossens/.dvipsrc, must_exist=1, find_all=0,
  path=./tex/dvips/./usr/local/texmf/dvips/).
kdebug:search($HOME/.dvipsrc) =>
kdebug:start search(file=config.cms, must_exist=0, find_all=0,
  path=./tex/dvips/./usr/local/texmf/dvips/).
kdebug:search(config.cms)
=>/usr/local/texmf/dvips/cms/config.cms

```

Figure 11: Finding configuration files

```

kdebug:start search(file=texc.pro, must\_exist=0, find\_all=0,
  path=./tex/dvips/./usr/local/texmf/dvips/./:
    ~/.tex/fonts/type1/./usr/local/texmf/fonts/type1/).
kdebug:search(texc.pro) => /usr/local/texmf/dvips/base/texc.pro

```

Figure 12: Finding the prolog file

```

kdebug:start search(file=cmr10.tfm, must\_exist=1, find\_all=0,
  path=./tex/fonts/tfm/./usr/local/texmf/fonts/tfm/./:
    /var/tex/fonts/tfm/).
kdebug:search(cmr10.tfm) => /usr/local/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must\_exist=0, find\_all=0,
  ...
<texps.pro>
kdebug:start search(file=cmr10.pfb, must\_exist=0, find\_all=0,
  path=./tex/dvips/./usr/local/texmf/dvips/./:
    ~/.tex/fonts/type1/./usr/local/texmf/fonts/type1/).
kdebug:search(cmr10.pfb) => /usr/local/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]

```

Figure 13: Finding the font file

Then it goes on to look for the prolog file `texc.pro`:

```

kdebug:start search(file=texc.pro, must_exist=0, find_all=0,
  path=./tex/dvips/./usr/local/texmf/dvips/./:
    ~/.tex/fonts/type1/./usr/local/texmf/fonts/type1/).
kdebug:search(texc.pro) => /usr/local/texmf/dvips/base/texc.pro

```

After having found the file in question, `dvips` outputs the date and time, and informs us that it will generate the file `hello-world.ps`, then that it needs the font file `cmr10`, and that the latter is declared as “resident” (no bitmaps needed):

```
TeX output 1998.02.26:1204' -> hello-world.ps
Defining font () cmr10 at 10.0pt
Font cmr10 <CMR10> is resident.
```

Now the search is on for the file `cmr10.tfm`, which is found, then a few more prolog files (not shown) are referenced, and finally the Type 1 instance `cmr10.pfb` of the font is located and included in the output file (see last line).

```
kdebug:start search(file=cmr10.tfm, must_exist=1, find_all=0,
  path=../tex/fonts/tfm/../../usr/local/texmf/fonts/tfm/../../
    /var/tex/fonts/tfm/).
kdebug:search(cmr10.tfm) => /usr/local/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must_exist=0, find_all=0,
  ...
<texps.pro>
kdebug:start search(file=cmr10.pfb, must_exist=0, find_all=0,
  path=../tex/dvips/../../usr/local/texmf/dvips/../../
    ~/tex/fonts/type1/../../usr/local/texmf/fonts/type1/).
kdebug:search(cmr10.pfb) => /usr/local/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]
```

### 8.3 Runtime options

Another useful feature of Web2C is its possibility to control a number of memory parameters (in particular, array sizes) via the runtime file `texmf.cnf` read by Kpathsea. The memory settings can be found in Part 3 of that file in the T<sub>E</sub>X Live distribution. The more important are:

- main\_memory** Total words of memory available, for T<sub>E</sub>X, METAFONT and MetaPost. You must make a new format file for each different setting. For instance, you could generate a “huge” version of T<sub>E</sub>X, and call the format file `hugetex.fmt`. Using the standard way of specifying the program name used by Kpathsea, the particular value of the `main_memory` variable will then be read from `texmf.cnf`.
- extra\_mem\_bot** Extra space for “large” T<sub>E</sub>X data structures: boxes, glue, breakpoints, etc. Especially useful if you use P<sub>I</sub>CT<sub>E</sub>X.
- font\_mem\_size** Number of words for font information available for T<sub>E</sub>X. This is more or less the total size of all TFM files read.
- hash\_extra** Additional space for the hash table of control sequence names. Only  $\approx 10,000$  control sequences can be stored in the main hash table; if you have a large book with numerous cross-references, this might not be enough. The default value of `hash_extra` is 50000.

Of course, this facility is no substitute for truly dynamic arrays and memory allocation, but since these are extremely difficult to implement in the present T<sub>E</sub>X source, these runtime parameters provide a practical compromise allowing some flexibility.

## 9 Acknowledgements

T<sub>E</sub>X Live is a joint effort by virtually all of the T<sub>E</sub>X user groups. This edition of T<sub>E</sub>X Live was overseen by Karl Berry. The other principal contributors, past and present, are listed below.

- The English, German, Dutch, and Polish T<sub>E</sub>X user groups (TUG, DANTE e.V., NTG, and GUST, respectively), which provide the necessary technical and administrative infrastructure. Please join the T<sub>E</sub>X user group near you! (See <http://tug.org/usergroups.html>.)

- The CTAN team, most notably Robin Fairbairns, Jim Hefferon, and Rainer Schöpf, which distributes the T<sub>E</sub>X Live images and provides the common infrastructure for package updates, upon which T<sub>E</sub>X Live depends.
- Peter Breitenlohner (and the  $\varepsilon$ -T<sub>E</sub>X team) for the stable foundation of future T<sub>E</sub>X's, and Peter specifically for stellar help with the GNU autotools usage throughout T<sub>E</sub>X Live.
- Thomas Esser, without whose marvelous t<sub>E</sub>T<sub>E</sub>X package T<sub>E</sub>X Live would have never existed.
- Michel Goossens, who co-authored the original documentation.
- Eitan Gurari, whose T<sub>E</sub>X4ht was used to create the HTML version of this documentation, and who worked tirelessly to improve it at short notice.
- Hàn Thế Thành, Martin Schröder, and the pdfT<sub>E</sub>X team (<http://pdfTEX.org>) for continuing enhancements of T<sub>E</sub>X's abilities.
- Hans Hagen, for much testing and making his ConT<sub>E</sub>Xt format (<http://pragma-ade.com>) work within T<sub>E</sub>X Live's framework.
- Hartmut Henkel, for significant development contributions to pdfT<sub>E</sub>X, LuaT<sub>E</sub>X, and more.
- Taco Hoekwater, for renewed development efforts on MetaPost and (Lua)T<sub>E</sub>X (<http://luatex.org>) itself, and for incorporating ConT<sub>E</sub>Xt into T<sub>E</sub>X Live.
- Paweł Jackowski, for the Windows installer tlp<sub>m</sub>, and Tomasz Łuczak, for tlp<sub>m</sub>gui, used in past releases.
- Akira Kakuto, for providing the Windows binaries from his W32T<sub>E</sub>X distribution for Japanese T<sub>E</sub>X (<http://www.fsci.fuk.kindai.ac.jp/kakuto/win32-ptex/>), and many other contributions.
- Jonathan Kew and SIL, for the major new development of XeT<sub>E</sub>X and taking the time and trouble to integrate it in T<sub>E</sub>X Live, as well as the initial version of the MacT<sub>E</sub>X installer.
- Dick Koch, for maintaining MacT<sub>E</sub>X (<http://tug.org/mactex>) in very close tandem with T<sub>E</sub>X Live, and for his great good cheer in doing so.
- Reinhard Kotucha, for major contributions to the T<sub>E</sub>X Live 2008 infrastructure and installer, as well as Windows research efforts, the `getnonfreefonts` script, and more.
- Siep Kroonenberg, also for major contributions to the T<sub>E</sub>X Live 2008 infrastructure and installer, especially on Windows, and for the bulk of work updating this manual describing the new features.
- Petr Olšák, who coordinated and checked all the Czech and Slovak material very carefully.
- Toshio Oshima, for his dviout previewer for Windows.
- Manuel Pégourié-Gonnard, for helping with package updates, documentation improvements, and tldoc development.
- Fabrice Popineau, for the original Windows support in T<sub>E</sub>X Live and work on the French documentation.
- Norbert Preining, the principal architect of the T<sub>E</sub>X Live 2008 infrastructure and installer, and also for coordinating the Debian version of T<sub>E</sub>X Live (together with Frank Küster), making many suggestions along the way.
- Sebastian Rahtz, for originally creating T<sub>E</sub>X Live and maintaining it for many years.
- Vladimir Volovich, for substantial help with porting and other maintenance issues, and especially for making it feasible to include xindy.
- Staszek Wawrykiewicz, the principal tester for all of T<sub>E</sub>X Live, and coordinator of the many major Polish contributions: fonts, Windows installation, and more.
- Olaf Weber, for his patient maintenance of Web2C.
- Gerben Wierda, for creating and maintaining the Mac OS X support, and much integration and testing.
- Graham Williams, on whose work the T<sub>E</sub>X Catalogue of packages depends.

Builders of the binaries: Tigran Aivazian (x86\_64-linux), Tim Arnold (hppa-hpux), Randy Au (amd64-freebsd), Edd Barrett (i386-openbsd), Karl Berry (i386-linux), Akira Kakuto (win32), Dick Koch (universal-darwin), Manfred Lotz (i386-freebsd), Norbert Preining (alpha-linux), Arthur

Reutenauer (`sparc-linux`), Thomas Schmitz (`powerpc-linux`), Apostolos Syropoulos (`i386-solaris`), Vladimir Volovich (`powerpc-aix`, `sparc-solaris`), Olaf Weber (`mips-irix`). For information on the T<sub>E</sub>X Live build process, see <http://tug.org/texlive/build.html>.

Documentation and translation updates: Karl Berry (English), Daniel Flipo (French), Klaus Höppner (German), Petr Sojka & Jan Busa (Czech/Slovak), Boris Veytsman (Russian), Staszek Wawrykiewicz (Polish). The T<sub>E</sub>X Live documentation web page is <http://tug.org/texlive/doc.html>.

Of course the most important acknowledgement must go to Donald Knuth, first for inventing T<sub>E</sub>X, and then for giving it to the world.

## 10 Release history

### 10.1 Past

Discussion began in late 1993 when the Dutch T<sub>E</sub>X Users Group was starting work on its 4AllT<sub>E</sub>X CD for MS-DOS users, and it was hoped at that time to issue a single, rational, CD for all systems. This was too ambitious a target for the time, but it did spawn not only the very successful 4AllT<sub>E</sub>X CD, but also the TUG Technical Council working group on a *T<sub>E</sub>X Directory Structure* (<http://tug.org/tds>), which specified how to create consistent and manageable collections of T<sub>E</sub>X support files. A complete draft of the TDS was published in the December 1995 issue of *TUGboat*, and it was clear from an early stage that one desirable product would be a model structure on CD. The distribution you now have is a very direct result of the working group's deliberations. It was also clear that the success of the 4AllT<sub>E</sub>X CD showed that Unix users would benefit from a similarly easy system, and this is the other main strand of T<sub>E</sub>X Live.

We first undertook to make a new Unix-based TDS CD in the autumn of 1995, and quickly identified Thomas Esser's teT<sub>E</sub>X as the ideal setup, as it already had multi-platform support and was built with portability across file systems in mind. Thomas agreed to help, and work began seriously at the start of 1996. The first edition was released in May 1996. At the start of 1997, Karl Berry completed a major new release of Web2c, which included nearly all the features which Thomas Esser had added in teT<sub>E</sub>X, and we decided to base the 2nd edition of the CD on the standard Web2C, with the addition of teT<sub>E</sub>X's `texconfig` script. The 3rd edition of the CD was based on a major revision of Web2C, 7.2, by Olaf Weber; at the same time, a new revision of teT<sub>E</sub>X was being made, and T<sub>E</sub>X Live included almost all of its features. The 4th edition followed the same pattern, using a new version of teT<sub>E</sub>X, and a new release of Web2C (7.3). The system now included a complete Windows setup.

For the 5th edition (March 2000) many parts of the CD were revised and checked, updating hundreds of packages. Package details were stored in XML files. But the major change for T<sub>E</sub>X Live 5 was that all non-free software was removed. Everything in T<sub>E</sub>X Live is now intended to be compatible with the Debian Free Software Guidelines (<http://www.debian.org/intro/free>); we have done our best to check the license conditions of all packages, but we would very much appreciate hearing of any mistakes.

The 6th edition (July 2001) had much more material updated. The major change was a new install concept: the user could select a more exact set of needed collections. Language-related collections were completely reorganized, so selecting any of them installs not only macros, fonts, etc., but also prepares an appropriate `language.dat`.

The 7th edition of 2002 had the notable addition of Mac OS X support, and the usual myriad of updates to all sorts of packages and programs. An important goal was integration of the source back with teT<sub>E</sub>X, to correct the drift apart in versions 5 and 6.

### 10.1.1 2003

In 2003, with the continuing flood of updates and additions, we found that T<sub>E</sub>X Live had grown so large it could no longer be contained on a single CD, so we split it into three different distributions (see section 2.1, p. 4). In addition:

- At the request of the L<sup>A</sup>T<sub>E</sub>X team, we changed the standard `latex` and `pdflatex` commands to now use  $\epsilon$ -T<sub>E</sub>X (see p. 6).
- The new Latin Modern fonts were included (and are recommended).
- Support for Alpha OSF was removed (HPUX support was removed previously), since no one had (or volunteered) hardware available on which to compile new binaries.
- Windows setup was substantially changed; for the first time an integrated environment based on XEmacs was introduced.
- Important supplementary programs for Windows (Perl, Ghostscript, ImageMagick, Ispell) are now installed in the T<sub>E</sub>X Live installation directory.
- Font map files used by `dvips`, `dvipdfm` and `pdftex` are now generated by the new program `updmap` and installed into `texmf/fonts/map`.
- T<sub>E</sub>X, METAFONT, and MetaPost now, by default, output most input characters (32 and above) as themselves in output (e.g., `\write`) files, log files, and the terminal, i.e., *not* translated using the `^^` notation. In T<sub>E</sub>X Live 7, this translation was dependent on the system locale settings; now, locale settings do not influence the T<sub>E</sub>X programs' behavior. If for some reason you need the `^^` output, rename the file `texmf/web2c/cp8bit.tcx`. (Future releases will have cleaner ways to control this.)
- This documentation was substantially revised.
- Finally, since the edition numbers had grown unwieldy, the version is now simply identified by the year: T<sub>E</sub>X Live 2003.

### 10.1.2 2004

2004 saw many changes:

- If you have locally-installed fonts which use their own `.map` or (much less likely) `.enc` support files, you may need to move those support files.  
`.map` files are now searched for in subdirectories of `fonts/map` only (in each `texmf` tree), along the `TEXFONTMAPS` path. Similarly, `.enc` files are now searched for in subdirectories of `fonts/enc` only, along the `ENCFONTS` path. `updmap` will attempt to warn about problematic files.  
 For methods of handling this and other information, please see <http://tug.org/texlive/mapenc.html>.
- The T<sub>E</sub>X Collection has been expanded with the addition of a MiK<sub>T</sub>E<sub>X</sub>-based installable CD, for those who prefer that implementation to Web2C. See section 2 (p. 4).
- Within T<sub>E</sub>X Live, the single large `texmf` tree of previous releases has been replaced by three: `texmf`, `texmf-dist`, and `texmf-doc`. See section 2.2 (p. 5), and the `README` files for each.
- All T<sub>E</sub>X-related input files are now collected in the `tex` subdirectory of `texmf*` trees, rather than having separate sibling directories `tex`, `etex`, `pdftex`, `pdfetex`, etc. See `texmf-doc/doc/english/tds/tds.html#Extensions`.
- Helper scripts (not meant to be invoked by users) are now located in a new `scripts` directory of `texmf*` trees, and searched for via `kpsewhich -format=texmfscripts`. So if you have programs which call such scripts, they'll need to be adjusted. See `texmf-doc/doc/english/tds/tds.html#Scripts`.

- Almost all formats leave most characters printable as themselves via the “translation file” `cp227.tcx`, instead of translating them with the `^^` notation. Specifically, characters at positions 32–256, plus tab, vertical tab, and form feed are considered printable and not translated. The exceptions are plain  $\text{\TeX}$  (only 32–126 printable),  $\text{\ConTeXt}$  (0–255 printable), and the  $\Omega$ -related formats. This default behavior is almost the same as in  $\text{\TeX}$  Live 2003, but it’s implemented more cleanly, with more possibilities for customization. See [texmf/doc/web2c/web2c.html#TCX-files](http://texmf/doc/web2c/web2c.html#TCX-files). (By the way, with Unicode input,  $\text{\TeX}$  may output partial character sequences when showing error contexts, since it is byte-oriented.)
- `pdfetex` is now the default engine for all formats except (plain) `tex` itself. (Of course it generates DVI when run as `latex`, etc.) This means, among other things, that the microtypographic features of `pdfetex` are available in  $\text{\LaTeX}$ ,  $\text{\ConTeXt}$ , etc., as well as the  $\varepsilon$ - $\text{\TeX}$  features ([texmf-dist/doc/etex/base/](http://texmf-dist/doc/etex/base/)).

It also means it’s *more important than ever* to use the `ifpdf` package (works with both plain and  $\text{\LaTeX}$ ) or equivalent code, because simply testing whether `\pdfoutput` or some other primitive is defined is not a reliable way to determine if PDF output is being generated. We made this backward compatible as best we could this year, but next year, `\pdfoutput` may be defined even when DVI is being written.

- `pdf $\text{\TeX}$`  (<http://pdfetex.org>) has many new features:
  - `\pdfmapfile` and `\pdfmapline` provide font map support from within a document.
  - Microtypographic font expansion can be used more easily.  
<http://www.ntg.nl/pipermail/ntg-pdfetex/2004-May/000504.html>
  - All parameters previously set through the special configuration file `pdfetex.cfg` must now be set through primitives, typically in `pdfetexconfig.tex`; `pdfetex.cfg` is no longer supported. Any extant `.fmt` files must be redumped when `pdfetexconfig.tex` is changed.
  - See the `pdf $\text{\TeX}$`  manual for more: [texmf/doc/pdfetex/manual](http://texmf/doc/pdfetex/manual).
- The `\input` primitive in `tex` (and `mf` and `mpost`) now accepts double quotes containing spaces and other special characters. Typical examples:

```
\input "filename with spaces"    % plain
\input{"filename with spaces"}    % latex
```

See the Web2C manual for more: [texmf/doc/web2c](http://texmf/doc/web2c).

- `enc $\text{\TeX}$`  support is now included within Web2C and consequently all  $\text{\TeX}$  programs, via the `-enc` option — *only when formats are built*. `enc $\text{\TeX}$`  supports general re-encoding of input and output, enabling full support of Unicode (in UTF-8). See [texmf-dist/doc/generic/encetex/](http://texmf-dist/doc/generic/encetex/) and <http://www.olsak.net/encetex.html>.
- Aleph, a new engine combining  $\varepsilon$ - $\text{\TeX}$  and  $\Omega$ , is available. A little information is available in [texmf-dist/doc/aleph/base](http://texmf-dist/doc/aleph/base) and <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=aleph>. The  $\text{\LaTeX}$ -based format for Aleph is named `lamed`.
- The latest  $\text{\LaTeX}$  release has a new version of the LPPL — now officially a Debian-approved license. Assorted other updates, see the `ltnews` files in [texmf-dist/doc/latex/base](http://texmf-dist/doc/latex/base).
- `dvipng`, a new program for converting DVI to PNG image files, is included. See [texmf/doc/man/man1/dvipng.1](http://texmf/doc/man/man1/dvipng.1).



- We reduced the `cbgreek` package to a “medium” sized set of fonts, with the assent and advice of the author (Claudio Beccari). The excised fonts are the invisible, outline, and transparency ones, which are relatively rarely used, and we needed the space. The full set is of course available from CTAN (<http://www.ctan.org/tex-archive/fonts/greek/cb>).
- `oxdvi` has been removed; just use `xdvi`.
- The `ini` and `vir` commands (links) for `tex`, `mf`, and `mpost` are no longer created, such as `initex`. The `ini` functionality has been available through the command-line option `-ini` for years now.
- `i386-openbsd` platform support was removed. Since the `tetex` package in the BSD Ports system is available, and GNU/Linux and FreeBSD binaries were available, it seemed volunteer time could be better spent elsewhere.
- On `sparc-solaris` (at least), you may have to set the `LD_LIBRARY_PATH` environment variable to run the `tlutils` programs. This is because they are compiled with C++, and there is no standard location for the runtime libraries. (This is not new in 2004, but wasn’t previously documented.) Similarly, on `mips-irix`, the MIPSpro 7.4 runtimes are required.

### 10.1.3 2005

2005 saw the usual huge number of updates to packages and programs. The infrastructure stayed relatively stable from 2004, but inevitably there were some changes there as well:

- New scripts `texconfig-sys`, `updmap-sys`, and `fmtutil-sys` were introduced, which modify the configuration in the system trees. The `texconfig`, `updmap`, and `fmtutil` scripts now modify user-specific files, under `$HOME/.texlive2005`.
- Corresponding new variables `TEXMFCONFIG` and `TEXMFSYSCONFIG` to specify the trees where configuration files (user or system, respectively) are found. Thus, you may need to move personal versions of `fmtutil.cnf` and `updmap.cfg` to these places; another option is to redefine `TEXMFCONFIG` or `TEXMFSYSCONFIG` in `texmf.cnf`. In any case the real location of these files and the values of `TEXMFCONFIG` and `TEXMFSYSCONFIG` must agree. See section 2.3, p. 5.
- Last year, we kept `\pdfoutput` and other primitives undefined for DVI output, even though the `pdfetex` program was being used. This year, as promised, we undid that compatibility measure. So if your document uses `\ifx\pdfoutput\undefined` to test if PDF is being output, it will need to be changed. You can use the package `ifpdf.sty` (which works under both plain  $\TeX$  and  $\LaTeX$ ) to do this, or steal its logic.
- Last year, we changed most formats to output (8-bit) characters as themselves (see previous section). The new TCX file `empty.tcx` now provides an easier way to get the original `^^` notation if you so desire, as in:

```
latex --translate-file=empty.tcx yourfile.tex
```

- The new program `dvipdfmx` is included for translation of DVI to PDF; this is an actively maintained update of `dvipdfm` (which is also still available for now, though no longer recommended).
- The new programs `pdfopen` and `pdfclose` are included to allow reloading of pdf files in the Adobe Acrobat Reader without restarting the program. (Other pdf readers, notably `xpdf`, `gv`, and `gsview`, have never suffered from this problem.)



- For consistency, the variables `HOMETEXMF` and `VARTEXMF` have been renamed to `TEXMFHOME` and `TEXMFSYSVAR`, respectively. There is also `TEXMFVAR`, which is by default user-specific. See the first point above.

#### 10.1.4 2006–2007

In 2006–2007, the major new addition to T<sub>E</sub>X Live was the XeT<sub>E</sub>X program, available as the `xetex` and `xelatex` programs; see <http://scripts.sil.org/xetex>.

MetaPost also received a notable update, with more planned for the future (<http://tug.org/metapost/articles>), likewise pdfT<sub>E</sub>X (<http://tug.org/applications/pdftex>).

The T<sub>E</sub>X `.fmt` (high-speed format) and the similar files for MetaPost and METAFONT are now stored in subdirectories of `texmf/web2c`, instead of in the directory itself (although the directory is still searched, for the sake of existing `.fmt`'s). The subdirectories are named for the ‘engine’ in use, such as `tex` or `pdftex` or `xetex`. This change should be invisible in normal use.

The (plain) `tex` program no longer reads `%&` first lines to determine what format to run; it is the pure Knuthian T<sub>E</sub>X. (L<sup>A</sup>T<sub>E</sub>X and everything else do still read `%&` lines).

Of course the year also saw (the usual) hundreds of other updates to packages and programs. As usual, please check CTAN (<http://www.ctan.org>) for updates.

Internally, the source tree is now stored in Subversion, with a standard web interface for viewing the tree, as linked from our home page. Although not visible in the final distribution, we expect this will provide a stable development foundation for future years.

Finally, in May 2006 Thomas Esser announced that he would no longer be updating teT<sub>E</sub>X (<http://tug.org/tetex>). As a result, there was been a surge of interest in T<sub>E</sub>X Live, especially among GNU/Linux distributors. (There is a new `tetex` installation scheme in T<sub>E</sub>X Live, which provides an approximate equivalent.) We hope this will eventually translate to improvements in the T<sub>E</sub>X environment for everyone.

Final note on the past: all releases of T<sub>E</sub>X Live, along with ancillary material such as CD labels, are available at <ftp://tug.org/historic/systems/texlive>.

## 10.2 Present

In 2008, the entire T<sub>E</sub>X Live infrastructure was redesigned and reimplemented. Complete information about an installation is now stored in a plain text file `tlpkg/texlive.tlpdb`.

Among other things, this finally makes possible upgrading a T<sub>E</sub>X Live installation over the Internet after the initial installation, a feature MiK<sub>T</sub>E<sub>X</sub> has provided for many years. We expect to regularly update new packages as they are released to CTAN.

The major new engine LuaT<sub>E</sub>X (<http://luatex.org>) is included; besides a new level of flexibility in typesetting, this provides an excellent scripting language for use both inside and outside of T<sub>E</sub>X documents.

Support among Windows and the Unix-based platforms is now much more uniform. In particular, most Perl and Lua scripts are now available on Windows, using the Perl internally distributed with T<sub>E</sub>X Live.

The new `tlmgr` script (section 6) is the general interface for managing T<sub>E</sub>X Live after the initial installation. It handles package updates and consequent regeneration of formats, map files, and language files, optionally including local additions.

With the advent of `tlmgr`, the `texconfig` actions to edit the format and hyphenation configuration files are now disabled.

The xindy indexing program (<http://xindy.sourceforge.net/>) is now included on most platforms.

The `kpsewhich` tool can now report all matches for a given file (option `-all`) and limit matches to a given subdirectory (option `-subdir`).

The `dvipdfmx` program now includes functionality to extract bounding box information, via the command name `extractbb`; this was one of the last features provided by `dvipdfm` not in `dvipdfmx`.

The font aliases `Times-Roman`, `Helvetica`, and so on have been removed. Different packages expected them to behave differently (in particular, to have different encodings), and there was no good way to resolve this.

The `platex` format has been removed, to resolve a name conflict with a completely different Japanese `platex`; the `polski` package is now the main Polish support.

Internally, the `WEB` string pool files are now compiled into the binaries, to ease upgrades.

### 10.3 Future

*T<sub>E</sub>X Live is not perfect!* (And never will be.) We intend to continue to release new versions, and would like to provide more help material, more utilities, more installation programs, and (of course) an ever-improved and better-checked tree of macros and fonts. This work is all done by hard-pressed volunteers in their limited spare time, and a great deal remains to be done. Please see <http://tug.org/texlive/contribute.html>.

Please send corrections, suggestions, and offers of help to:

[tex-live@tug.org](mailto:tex-live@tug.org)  
<http://tug.org/texlive>

*Happy T<sub>E</sub>Xing!*