

# *The T<sub>E</sub>X Live Guide—2024*

Karl Berry, editor

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

2 May 2024

## 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 . . . . .	4
1.3	Basic installation of T <sub>E</sub> X Live . . . . .	4
1.4	Security considerations . . . . .	4
1.5	Getting help . . . . .	5
<b>2</b>	<b>Overview of T<sub>E</sub>X Live</b>	<b>5</b>
2.1	The T <sub>E</sub> X Collection: T <sub>E</sub> X Live, MacT <sub>E</sub> X, MiK <sub>T</sub> <sub>E</sub> X, CTAN . . . . .	6
2.2	Top level T <sub>E</sub> X Live directories . . . . .	6
2.3	Overview of the predefined texmf trees . . . . .	7
2.4	Extensions to T <sub>E</sub> X . . . . .	8
2.5	Other notable programs in T <sub>E</sub> X Live . . . . .	8
<b>3</b>	<b>Installation</b>	<b>9</b>
3.1	Starting the installer . . . . .	9
3.1.1	Unix . . . . .	9
3.1.2	macOS . . . . .	10
3.1.3	Windows . . . . .	11
3.1.4	Cygwin . . . . .	11
3.1.5	The text installer . . . . .	11
3.1.6	The graphical installer . . . . .	12
3.2	Running the installer . . . . .	12
3.2.1	Binary systems menu (Unix only) . . . . .	12
3.2.2	Selecting what is to be installed . . . . .	13
3.2.3	Directories . . . . .	14
3.2.4	Options . . . . .	16
3.3	Command-line install-tl options . . . . .	17
3.3.1	The <code>-repository</code> option . . . . .	18
3.4	Post-install actions . . . . .	18
3.4.1	Environment variables for Unix . . . . .	18
3.4.2	Environment variables: Global configuration . . . . .	19

3.4.3	Internet updates after DVD installation . . . . .	19
3.4.4	System font configuration for XeTeX and LuaTeX . . . . .	20
3.4.5	ConTeXt LMTX and MKIV . . . . .	20
3.4.6	Integrating local and personal macros . . . . .	21
3.4.7	Integrating third-party fonts . . . . .	21
3.5	Testing the installation . . . . .	22
3.6	Uninstalling TeX Live . . . . .	23
3.7	Links for additional downloadable software . . . . .	24
<b>4</b>	<b>Specialized installations</b>	<b>24</b>
4.1	Shared-user (or cross-machine) installations . . . . .	25
4.2	Portable (USB) installations . . . . .	25
<b>5</b>	<b>tlmgr: Managing your installation</b>	<b>26</b>
5.1	GUI interfaces for tlmgr . . . . .	26
5.2	Sample tlmgr command-line invocations . . . . .	27
<b>6</b>	<b>Notes on Windows</b>	<b>29</b>
6.1	Windows-specific features . . . . .	29
6.2	Additional software included on Windows . . . . .	29
6.3	Using external Perl, Tcl/Tk and Ghostscript installations . . . . .	30
6.4	User Profile is Home . . . . .	30
6.5	The Windows registry . . . . .	30
6.6	Windows permissions . . . . .	31
6.7	Increasing maximum memory on Windows and Cygwin . . . . .	31
<b>7</b>	<b>A user's guide to Web2C</b>	<b>32</b>
7.1	Kpathsea path searching . . . . .	33
7.1.1	Path sources . . . . .	33
7.1.2	Config files . . . . .	34
7.1.3	Path expansion . . . . .	35
7.1.4	Default expansion . . . . .	35
7.1.5	Brace expansion . . . . .	35
7.1.6	Subdirectory expansion . . . . .	36
7.1.7	Summary of special characters in <code>texmf.cnf</code> files . . . . .	36
7.2	Filename databases . . . . .	36
7.2.1	The filename database . . . . .	36
7.2.2	kpsewhich: Standalone path searching . . . . .	37
7.2.3	Examples of use . . . . .	38
7.2.4	Debugging actions . . . . .	39
7.3	Runtime options . . . . .	42
7.4	<code>\$TEXMFDOTDIR</code> . . . . .	42
<b>8</b>	<b>Acknowledgements</b>	<b>43</b>

<b>9</b>	<b>Release history</b>	<b>45</b>
9.1	Past . . . . .	45
9.1.1	2003 . . . . .	46
9.1.2	2004 . . . . .	46
9.1.3	2005 . . . . .	48
9.1.4	2006–2007 . . . . .	49
9.1.5	2008 . . . . .	50
9.1.6	2009 . . . . .	50
9.1.7	2010 . . . . .	51
9.1.8	2011 . . . . .	52
9.1.9	2012 . . . . .	52
9.1.10	2013 . . . . .	53
9.1.11	2014 . . . . .	53
9.1.12	2015 . . . . .	54
9.1.13	2016 . . . . .	55
9.1.14	2017 . . . . .	56
9.1.15	2018 . . . . .	57
9.1.16	2019 . . . . .	57
9.1.17	2020 . . . . .	58
9.1.18	2021 . . . . .	59
9.1.19	2022 . . . . .	61
9.1.20	2023 . . . . .	62
9.2	Present: 2024 . . . . .	64
9.3	Future . . . . .	65

## 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, macOS, 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 user groups distribute among their members, or in other ways. Section 2.1 briefly describes the contents of the 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.

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 9 (p. 45).

## 1.2 Operating system support

T<sub>E</sub>X Live contains binaries for many Unix-based platforms, including GNU/Linux, macOS, and Cygwin. The included sources can be compiled on platforms for which we do not provide binaries.

As to Windows: Windows 7 and later are supported. Windows Vista *may* still mostly work, but T<sub>E</sub>X Live will no longer even install on Windows XP or earlier. T<sub>E</sub>X Live includes 64-bit executables for Windows.

See section 2.1 for alternate solutions for Windows and macOS.

## 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 (<https://tug.org/texlive/acquire.html>). The net installer itself is small, and downloads everything requested from the Internet.

The DVD installer lets you install to a local disk, but you cannot run T<sub>E</sub>X Live directly from the T<sub>E</sub>X Collection DVD (or the T<sub>E</sub>X Collection or T<sub>E</sub>X Live `.iso` images). You *can* prepare a runnable installation on, e.g., a USB stick (see section 4.2). Installation is described in later sections (p. 9), but here is a quick start:

- For Unix, the installation script is `install-tl`; on Windows, one should instead invoke `install-tl-windows`. The installer will operate in a graphical mode given the option `-gui` (default for Windows), or a text mode given the option `-gui=text` (default for everything else).
- One of the installed items is the ‘T<sub>E</sub>X Live Manager’ program, 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 Security considerations

To the best of our knowledge, the core T<sub>E</sub>X programs themselves are (and always have been) extremely robust. However, the contributed programs in T<sub>E</sub>X Live may not reach the same level, despite everyone’s best efforts. As always, you should be careful when running programs on untrusted input; to improve safety, use a new subdirectory or `chroot`.

This need for care is especially urgent on Windows, since in general Windows finds programs in the current directory before anything else, regardless of the search path. This opens up a wide variety of possible attacks. We have closed many holes, but undoubtedly some remain, especially with third-party programs. Thus, we recommend checking for suspicious files in the current directory, especially executables (binaries or scripts). Ordinarily they should not be present, and definitely should not normally be created by merely processing a document.

Finally, T<sub>E</sub>X (and its companion programs) are able to write files when processing documents, a feature that can also be abused in a wide variety of ways. Again, processing unknown documents in a new subdirectory is the safest bet.

Another aspect of security is ensuring that downloaded material has not been changed from what was created. The `tlmgr` program (section 5) will automatically perform cryptographic verification on downloads if the `gpg` (GNU Privacy Guard) program is

available. It is not distributed as part of T<sub>E</sub>X Live, but see <https://texlive.info/tlgpg/> for information about `gpg` if need be.

## 1.5 Getting help

The T<sub>E</sub>X community is active and friendly, and most serious questions end up getting answered. However, the support is informal, done by volunteers and casual users, so it's especially important that you do your homework before asking. (If you prefer guaranteed commercial support, you can forgo T<sub>E</sub>X Live completely and purchase a vendor's system; <https://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 <https://tug.org/begin.html> gives a brief introduction to the system.

**CTAN** If you are looking for a particular package, font, program, etc., CTAN is the place to start looking. It is a huge collection of all T<sub>E</sub>X-related items. The catalog entries also tell you whether a package is available for T<sub>E</sub>X Live or MiK<sub>T</sub>E<sub>X</sub>. See <https://ctan.org>.

**T<sub>E</sub>X FAQ** The T<sub>E</sub>X FAQ is a compendium of answers to all sorts of questions, from the most basic to the most arcane. It is available on the Internet through <https://texfaq.org>.

**T<sub>E</sub>X Web Resources** The web page <https://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 sites** A general web search never hurts. Principal support sites for T<sub>E</sub>X include the L<sup>A</sup>T<sub>E</sub>X community site at <https://latex.org>, the q&a site <https://tex.stackexchange.com>, and the mailing list <https://lists.tug.org/texhax>. Their archives have years of past questions and answers for your searching pleasure.

**asking questions** If you cannot find an answer, you can post to <http://latex.org/> and <https://tex.stackexchange.com/> through their web interfaces, to [texhax@tug.org](mailto:texhax@tug.org) through email (it's not necessary to subscribe in order to post). But before you post anywhere, please read this FAQ entry, to maximize your chances of getting a useful answer: <https://texfaq.org/FAQ-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. All the above resources 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, as well as 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, MacT<sub>E</sub>X, MiK<sub>T</sub>E<sub>X</sub>, CTAN

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

**T<sub>E</sub>X Live** A comprehensive cross-platform T<sub>E</sub>X system to be installed to disk. Home page: <https://tug.org/texlive/>.

**MacT<sub>E</sub>X** for macOS. This adds a native macOS installer and other Mac applications to T<sub>E</sub>X Live. Home page: <https://tug.org/mactex/>.

**MiK<sub>T</sub>E<sub>X</sub>** Another comprehensive cross-platform T<sub>E</sub>X distribution for Windows, GNU/Linux and macOS (but only Windows binaries are included on the DVD). It has an integrated package manager which installs missing components from the Internet, as needed. Home page: <https://miktex.org/>.

**CTAN** A snapshot of the CTAN repository (<https://ctan.org/>). CTAN does not 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 a T<sub>E</sub>X Live installation.

- `bin` The T<sub>E</sub>X system programs, arranged by platform.
- `readme-*.dir` Quick overview and useful links for T<sub>E</sub>X Live, in various languages, in both HTML and plain text.
- `source` The source to all included programs, including the main Web2C-based T<sub>E</sub>X distributions.
- `texmf-dist` The principal tree; see `TEXMFDIST` below.
- `tlpkg` Scripts, programs and data for managing the installation, and special support for Windows.

For documentation, the comprehensive links in the top-level file `doc.html` may be helpful. The documentation for nearly everything (packages, formats, fonts, program manuals, man pages, Info files) is in `texmf-dist/doc`. You can use the `texdoc` program to find documentation wherever it is located.

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

- Czech/Slovak: `texmf-dist/doc/texlive/texlive-cz`
- German: `texmf-dist/doc/texlive/texlive-de`
- English: `texmf-dist/doc/texlive/texlive-en`
- French: `texmf-dist/doc/texlive/texlive-fr`
- Italian: `texmf-dist/doc/texlive/texlive-it`
- Japanese: `texmf-dist/doc/texlive/texlive-ja`
- Polish: `texmf-dist/doc/texlive/texlive-pl`
- Russian: `texmf-dist/doc/texlive/texlive-ru`
- Serbian: `texmf-dist/doc/texlive/texlive-sr`
- Spanish: `texmf-dist/doc/texlive/texlive-es`
- Simplified Chinese: `texmf-dist/doc/texlive/texlive-zh-cn`

### 2.3 Overview of the predefined texmf trees

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

All of the trees, including the personal ones, should follow the T<sub>E</sub>X Directory Structure (TDS, <https://tug.org/tds>), with all its myriad subdirectories, or files may not be found. Section 3.4.6 (p. 21) describes this in more detail. The order here is the reverse order in which the trees are searched, that is, later trees in the list override earlier ones.

**TEXMFDIST** The tree which holds nearly all of the files in the original distribution—configuration files, scripts, packages, fonts, etc. (The main exceptions are the per-platform executables, which are stored in a sibling directory `bin/.`)

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

**TEXMFSYSCONFIG** The (site-wide) tree used by `texconfig-sys`, `updmap-sys`, and `fmtutil-sys` to store modified configuration data.

**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. For each user, this variable points to their own individual directory.

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

**TEXMFCONFIG** The (personal) tree used by the utilities `texconfig`, `updmap-user`, and `fmtutil-user` to store modified configuration data.

**TEXMFCACHE** The tree(s) used by ConT<sub>E</sub>Xt MkIV and LuaL<sup>A</sup>T<sub>E</sub>X to store (cached) runtime data; defaults to **TEXMFSYSVAR**, or (if that's not writable), **TEXMFVAR**.

The default layout is:

**system-wide root** can span multiple T<sub>E</sub>X Live releases (`/usr/local/texlive` by default on Unix):

2023 A previous release.

2024 The current release.

`bin`

`i386-linux` GNU/Linux binaries (32-bit)

...

`universal-darwin` macOS binaries

`x86_64-linux` GNU/Linux binaries (64-bit)

`windows` Windows binaries (64-bit)

`texmf-dist` **TEXMFDIST** and **TEXMFMAIN**

`texmf-var` **TEXMFSYSVAR**, **TEXMFCACHE**

`texmf-config` **TEXMFSYSCONFIG**

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

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

`.texlive2023` Privately generated and configuration data for a previous release.



`.texlive2024` Privately generated and configuration data for the current release.  
`texmf-var` TEXMFVAR, TEXMFCACHE  
`texmf-config` TEXMFCONFIG  
`texmf` TEXMFHOME Personal macros, etc.

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

Knuth's original T<sub>E</sub>X itself is frozen, apart from rare bug fixes. It is present in T<sub>E</sub>X Live as the program `tex`, and will remain so for the foreseeable future. T<sub>E</sub>X Live also contains several extended versions of T<sub>E</sub>X (also known as T<sub>E</sub>X engines):

**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>L 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`.

**pdfT<sub>E</sub>X** builds on the ε-T<sub>E</sub>X extensions, adding support for writing PDF output as well as DVI, and many non-output-related extensions. This is the program invoked for many common formats, e.g., `etex`, `latex`, `pdflatex`. Its web site is <http://www.pdfTEX.org/>. See `texmf-dist/doc/pdfTEX/manual/pdfTEX-a.pdf` for the manual, and `texmf-dist/doc/pdfTEX/samplepdfTEX/samplepdf.tex` for example usage of some of its features.

**LuaT<sub>E</sub>X** adds support for Unicode input and OpenType/TrueType- and system fonts. It also incorporates a Lua interpreter (<https://lua.org/>), enabling solutions for many thorny T<sub>E</sub>X problems. When called as `texlua`, it functions as a standalone Lua interpreter. Its web site is <http://www.luatex.org/>, and the reference manual is `texmf-dist/doc/luatex/base/luatex.pdf`.

**(e)(u)pT<sub>E</sub>X** have native support for Japanese typesetting requirements; pT<sub>E</sub>X is the basic engine, while the e- variants add ε-T<sub>E</sub>X functionality and u- add Unicode support.

**XeT<sub>E</sub>X** adds support for Unicode input and OpenType/TrueType- and system fonts, implemented using standard third-party libraries. See <https://tug.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. Omega is no longer included in T<sub>E</sub>X Live as a separate program; only Aleph is provided:

**Aleph** combines the Ω and ε-T<sub>E</sub>X extensions. See `texmf-dist/doc/aleph/base`.

## 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`, `biber` bibliography support.

`makeindex`, `upmendex`, `xindex`, `xindy` index support.

`dvips` convert DVI to PostScript.

`dvipdfmx` convert DVI to PDF, an alternative approach to pdfT<sub>E</sub>X (mentioned above).

`xdvi` DVI previewer for the X Window System.

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



psselect, psnup, ... PostScript utilities.  
 pdfjam, pdfjoin, ... PDF utilities.  
 context, mtxrun ConTeXt and PDF processor.  
 htlatex, ... tex4ht: (L<sup>A</sup>)TeX to HTML (and XML and DocX and more) converter.

## 3 Installation

### 3.1 Starting the installer

To begin, get the TeX Collection DVD or download the TeX Live net installer. See <https://tug.org/texlive/acquire.html> for more information and other methods of getting the software.

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

**Net installer, Windows .exe:** Download from CTAN as above, and double-click. This starts up a first-stage installer and unpacker; see figure 1. It gives two choices: ‘Install’ and ‘Unpack only’.

**TeX Collection DVD:** go to the DVD’s `texlive` subdirectory. Under Windows, the installer normally starts automatically when you insert the DVD. You can get the DVD by becoming a member of a TeX user group (highly recommended, <https://tug.org/usergroups.html>), or purchasing it separately (<https://tug.org/store>). You can burn your own DVD for TeX Live from the ISO image downloaded from CTAN. You can also mount the ISO directly on most systems. After installing from DVD or ISO, if you want to get continuing updates from the Internet, please see section 3.4.3.

The same installer program is run, whatever the source. The most notable difference between the two is that with the net installer, what you end up with is the packages that are currently available. This is in contrast to the DVD and ISO images, which are not updated between the major public releases.

If you need to download through proxies, use a `~/.wgetrc` file or environment variables with the proxy settings for Wget ([https://www.gnu.org/software/wget/manual/html\\_node/Proxies.html](https://www.gnu.org/software/wget/manual/html_node/Proxies.html)), or the equivalent for whatever download program you are using. This does not matter if you are installing from the DVD or ISO image.

The following sections explain installer start-up in more detail.

#### 3.1.1 Unix

Below, `>` denotes the shell prompt; user input is **bold**. The `install-tl` program is a Perl script. The simplest way to start it on a Unix-compatible system is as follows:

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

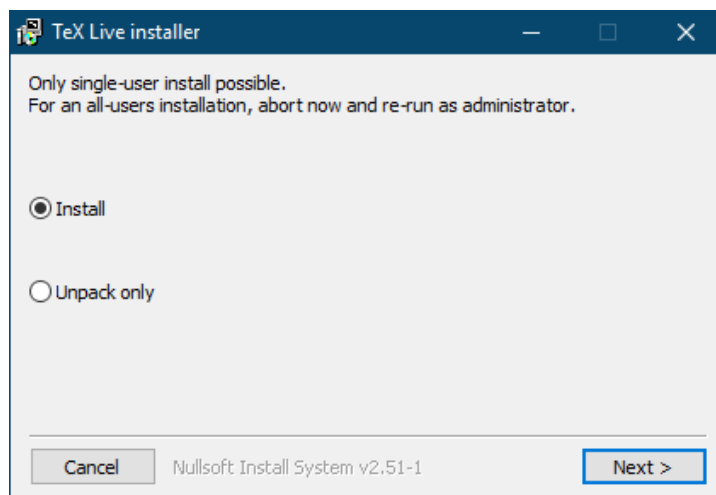


Figure 1: First stage of Windows `.exe` installer. Pressing the Install button will give you the window displayed in figure 3.

(Or you can invoke `/path/to/installer/install-tl` if it stayed executable, or `cd` to the directory first, etc.; we won't repeat all the variations.) You may have to enlarge your terminal window so that it shows the full text installer screen (figure 2).

To install in GUI mode (figure 4), you'll need to have Tcl/Tk installed. Given that, you can run:

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

The old `-wizard` and `-perltk/-expert` options now do the same thing as `-gui`. For a complete listing of the various options:

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

**About Unix permissions:** Your `umask` at the time of installation will be respected by the TeX 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.

**Special considerations for Cygwin:** Unlike other Unix-compatible systems, Cygwin does not by default include all of the prerequisite programs needed by the TeX Live installer. See section 3.1.4 for the packages which you need to install first.

### 3.1.2 macOS

As mentioned in section 2.1, a separate distribution is prepared for macOS, named MacTeX (<https://tug.org/mactex>). We recommend using the native MacTeX installer instead of the TeX Live installer on macOS, because the native installer makes a few Mac-specific adjustments, in particular to allow easily switching between different TeX Live releases on Macs, using the so-called TeXDist data structure.

MacTeX is firmly based on TeX Live, and the main TeX trees and binaries are precisely the same. It adds a few extra folders with Mac-specific documentation and applications.

### 3.1.3 Windows

If you are using the unpacked downloaded zip file, or the DVD installer failed to start automatically, double-click `install-tl-windows.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-windows
```

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

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

for the T<sub>E</sub>X Collection DVD, supposing that `D:` is the optical drive. Figure 3 displays the initial basic screen of the GUI installer, which is the default for Windows.

To install in text mode, use:

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

For a complete listing of the various options:

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

Add a `.bat` extension to the invocations if the same directory also contains `install-tl-windows.exe`. This will not normally be the case, unless you've mirrored the `tlnet` directory locally.

Non-ASCII characters in the path for the T<sub>E</sub>X Live installation must be avoided. This may be a problem particularly under Windows 11, where users are forced to use a Microsoft network account. See <https://tug.org/texlive/windows.html#nonascii> for some workarounds.

### 3.1.4 Cygwin

Before beginning the installation, use Cygwin's `setup.exe` program to install the `perl` and either `curl` or `wget` packages if you have not already done so. The following additional packages are recommended:

- `fontconfig` [needed by XeT<sub>E</sub>X and LuaT<sub>E</sub>X]
- `ghostscript` [needed by various utilities]
- `libXaw7` [needed by `xdvi`]
- `ncurses` [provides the `clear` command used by the installer]

### 3.1.5 The text installer

Figure 2 displays the main text mode screen under Unix. The text installer is the default on 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 in order to make it run on as many platforms as possible, even with a minimal Perl.

```

Installing TeX Live 2024 from: ...
Platform: x86_64-linux => 'GNU/Linux on x86_64'
Distribution: inst (compressed)
Directory for temporary files: /tmp
...
Detected platform: GNU/Linux on Intel x86_64

<B> binary platforms: 1 out of 16

<S> set installation scheme: scheme-full

<C> customizing installation collections
    40 collections out of 41, disk space required: 8296 MB (free: 138718 MB)

<D> directories:
    TEXDIR (the main TeX directory):
        /usr/local/texlive/2024
    ...

<O> options:
    [ ] use letter size instead of A4 by default
    ...

<V> set up for portable installation

Actions:
<I> start installation to hard disk
<P> save installation profile to 'texlive.profile' and exit
<H> help
<Q> quit

```

Figure 2: Main text installer screen (GNU/Linux)

### 3.1.6 The graphical installer

The default graphical installer starts out simple, with just a few options; see figure 3. It can be started with

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

The Advanced button gives access to most of the options of the text installer; see figure 4.

The `perlTk/expert` and `wizard` GUI options now invoke the regular graphical installer.

## 3.2 Running the installer

The installer is intended to be mostly self-explanatory, but following are a few notes about the various options and submenus.

### 3.2.1 Binary systems menu (Unix only)

Figure 5 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

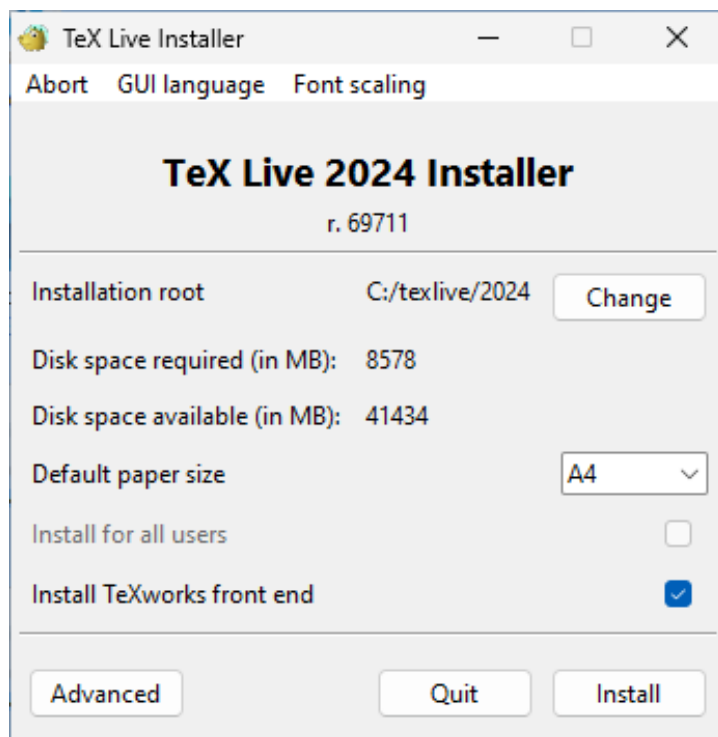


Figure 3: Basic installer screen (Windows); the Advanced button will result in something like figure 4

for other platforms as well. This can be useful if you are sharing a  $\text{\TeX}$  tree across a network of heterogeneous machines, or for a dual-boot system.

### 3.2.2 Selecting what is to be installed

Figure 6 displays the  $\text{\TeX}$  Live scheme menu; from here, you choose a “scheme”, which is an overall set of package collections. The default **full** scheme installs everything available. This is recommended, but you can also choose the **basic** scheme for just plain and  $\text{\LaTeX}$ , **small** for a few more programs (equivalent to the so-called Basic $\text{\TeX}$  installation of Mac $\text{\TeX}$ ), **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 ‘collections’ menu (figure 7, shown in GUI mode for a change).

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  $\text{\TeX}$  Live Manager (**tlmgr**) program after installation (see section 5); using that, you can control the installation at the package level.

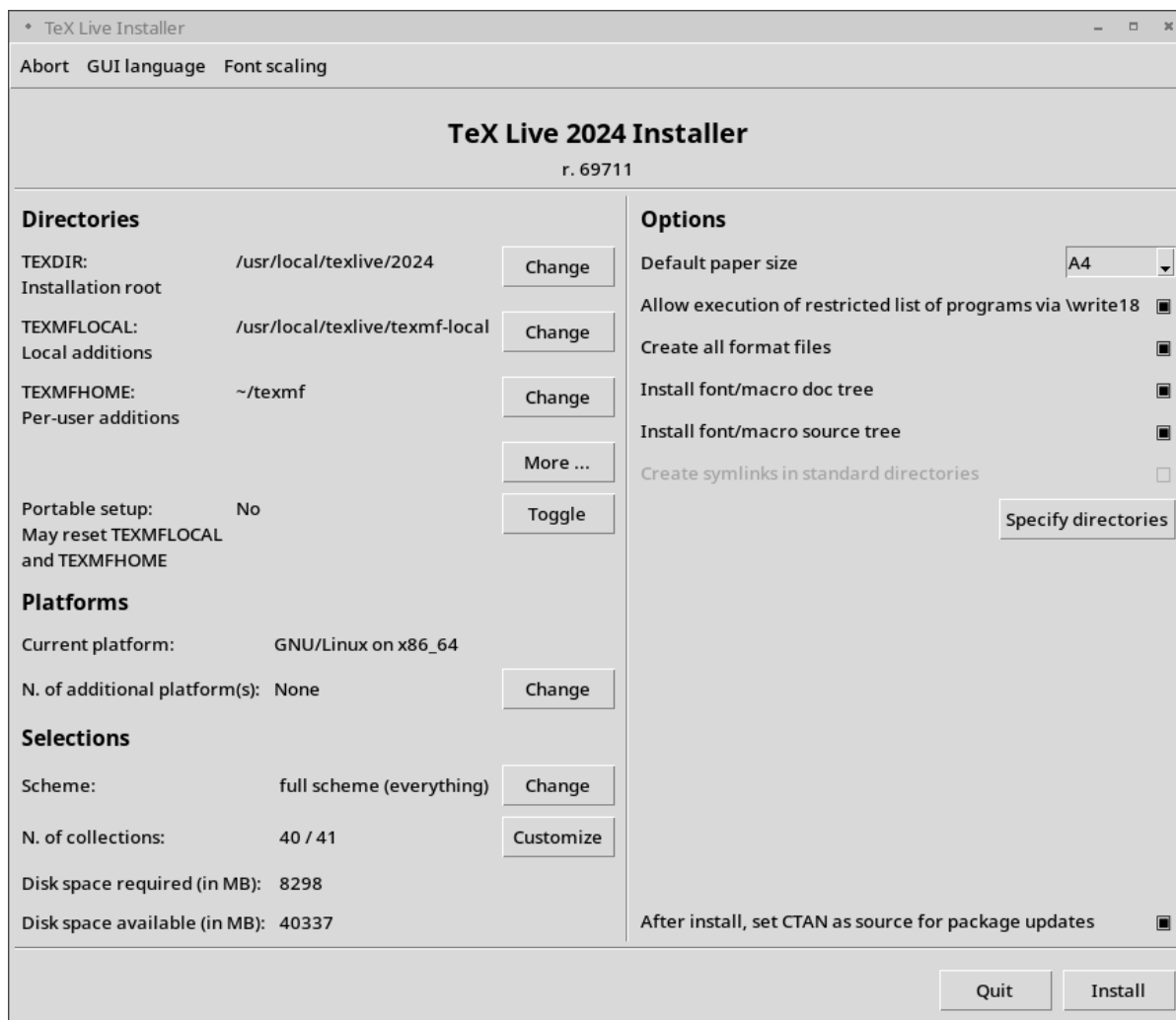


Figure 4: Advanced GUI installer screen (GNU/Linux)

### 3.2.3 Directories

The default layout is described in section 2.3, p. 7. The default installation directory is `/usr/local/texlive/2024` on Unix and `C:\texlive\2024` on Windows. This arrangement enables having many parallel TeX Live installations, such as one for each release (typically by year, as here), and you can switch between them merely by altering your search path.

That installation directory can be overridden by setting the so-called `TEXDIR` in the installer. The GUI screen for this and other options is shown in figure 4. The most common reasons to change it are either lacking enough disk space in that partition (the full TeX Live needs several gigabytes), or lacking write permission for the default location (you don't have to be root or administrator to install TeX Live, but you do need write access to the target directory).

On Windows, you normally do not need to be an administrator to create `C:\texlive\2024` (or, more generally, `%SystemDrive%\texlive\2024`).

The installation directories can also be changed by setting a variety of environment variables before running the installer (most likely, `TEXLIVE_INSTALL_PREFIX` or

```

Available platforms:
=====
a [ ] Cygwin on x86_64 (x86_64-cygwin)
b [ ] MacOSX current (10.14-) on ARM/x86_64 (universal-darwin)
c [ ] MacOSX legacy (10.6-) on x86_64 (x86_64-darwinlegacy)
d [ ] FreeBSD on x86_64 (amd64-freebsd)
e [ ] FreeBSD on Intel x86 (i386-freebsd)
f [ ] GNU/Linux on ARM64 (aarch64-linux)
g [ ] GNU/Linux on RPi(32-bit) and ARMv7 (armhf-linux)
h [ ] GNU/Linux on Intel x86 (i386-linux)
i [X] GNU/Linux on x86_64 (x86_64-linux)
j [ ] GNU/Linux on x86_64 with musl (x86_64-linuxmusl)
k [ ] NetBSD on x86_64 (amd64-netbsd)
l [ ] NetBSD on Intel x86 (i386-netbsd)
m [ ] Solaris on Intel x86 (i386-solaris)
o [ ] Solaris on x86_64 (x86_64-solaris)
p [ ] Windows (64-bit) (windows)

```

Figure 5: Binaries menu

```

Select scheme:
=====
a [X] full scheme (everything)
b [ ] medium scheme (small + more packages and languages)
c [ ] small scheme (basic + xetex, metapost, a few languages)
d [ ] basic scheme (plain and latex)
e [ ] minimal scheme (plain only)
f [ ] infrastructure-only scheme (no TeX at all)
g [ ] book publishing scheme (core LaTeX and add-ons)
h [ ] ConTeXt scheme
i [ ] GUST TeX Live scheme
j [ ] teTeX scheme (more than medium, but nowhere near full)
k [ ] custom selection of collections

```

Figure 6: Scheme menu

TEXLIVE\_INSTALL\_TEXDIR); see the documentation from `install-tl --help` (available online at <https://tug.org/texlive/doc/install-tl.html>) for the full list and more details.

A reasonable alternative destination is a directory under your home, especially if you will be the sole user. Use `~` to indicate this, as in `~/texlive/2024`.

We recommend including the year in the name, to enable keeping different releases of TeX Live side by side. (You may wish to also maintain a version-independent name, such as `/usr/local/texlive-cur`, via a symbolic link, which you can then repoint 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` (`~/Library/texmf` on Macs). In contrast to `TEXDIR`, here a `~` is preserved in the newly-written configuration files, since it usefully refers to the home directory of the user running TeX. It expands to `$HOME` on Unix and `%USERPROFILE%` on



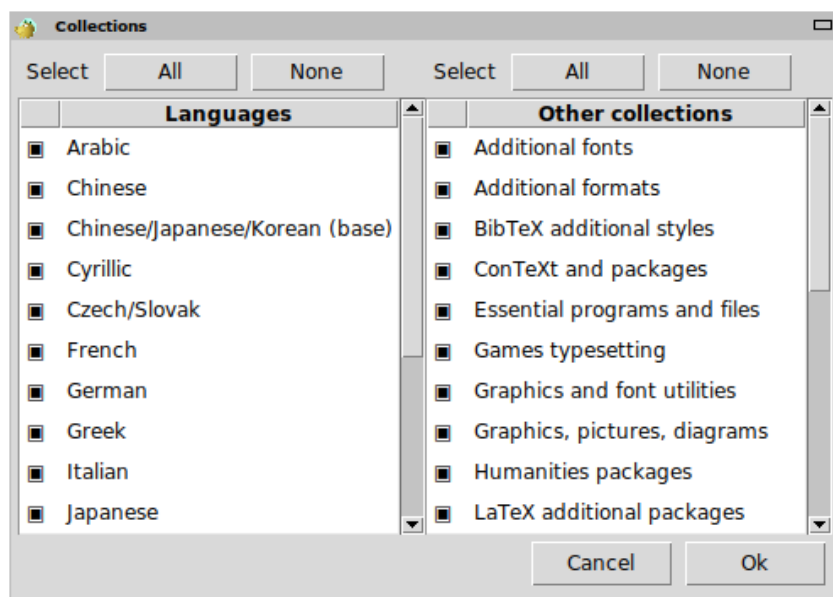


Figure 7: Collections menu

Windows. Special redundant note: `TEXMFHOME`, like all trees, must be organized according to the TDS, or files may not be found.

`TEXMFVAR` is the location for storing most cached runtime data specific to each user. `TEXMFCACHE` is the variable name used for that purpose by `Lua $\LaTeX$`  and `Con $\TeX$ t MkIV` (see section 3.4.5, p. 20); its default value is `TEXMFSYSVAR`, or (if that's not writable), `TEXMFVAR`.

### 3.2.4 Options

```
Options customization:
=====
<P> use letter size instead of A4 by default: [ ]
<E> execution of restricted list of programs: [X]
<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:
<Y> after install, set CTAN as source for package updates: [X]
```

Figure 8: Options menu (Unix)

Figure 8 shows the text mode options menu. More info on each:

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

**execution of restricted list of programs:** As of `T $\TeX$  Live 2010`, execution of a few external programs is allowed by default. The (very short) list of allowed programs is given in the `texmf.cnf`. See the 2010 news (section 9.1.7) for more details.

**create all format files:** We recommend leaving this option checked, to avoid unnecessary problems when creating formats dynamically. See the `fmtutil` documentation for more details.

**install font/macro ... tree:** Download/install the documentation and source files included in most packages. Unchecking is not recommended.

**create symlinks in standard directories:** This option (Unix only) bypasses the need to change environment variables. Without this option, T<sub>E</sub>X Live directories usually have to be added to `PATH`, `MANPATH` and `INFOPATH`. You will need write permissions to the target directories. This option is intended for accessing the T<sub>E</sub>X system through directories that are already known to users, such as `/usr/local/bin`, which don't already contain any T<sub>E</sub>X files. Do not overwrite existing files on your system with this option, e.g., by specifying system directories. The safest and recommended approach is to leave the option unchecked.

**after install, set CTAN as source for package updates:** When installing from a DVD or ISO image, this option is enabled by default, since usually one wants to take any subsequent package updates from the CTAN area that is updated throughout the year. The only likely reason to disable it is if you install only a subset from the DVD and plan to augment the installation later. In any case, the package repository for the installer, and for updates after installation, can be set independently as needed; see section 3.3.1 and section 3.4.3.

Windows-specific options, as displayed in the advanced GUI interface:

**adjust searchpath** This ensures that all programs will see the T<sub>E</sub>X Live binary directory on their search path.

**add menu shortcuts** If set, there will be a T<sub>E</sub>X Live submenu of the Start menu. There is a third option 'Launcher entry' besides 'TeX Live menu' and 'No shortcuts'. This option is described in section 4.1.

**File associations** The options are 'Only new' (create file associations, but do not overwrite existing ones), 'All' and 'None'.

**install T<sub>E</sub>Xworks front end**

When all the settings are to your liking, you can type 'I' in the text interface, or press the 'Install' button in the GUI, to start the installation process. When it is done, skip to section 3.4 to read what else needs to be done, if anything.

### 3.3 Command-line install-tl options

Type

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

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

`-gui` Use the GUI installer if possible. This requires Tcl/Tk version 8.5 or higher. This was distributed with older macOS; for Big Sur and later, you'll need to install Tcl/Tk yourself, if you don't choose to use the MacT<sub>E</sub>X installer. Tcl/Tk is distributed with T<sub>E</sub>X Live on Windows. The legacy options `-gui=perltk` and `-gui=wizard` are still available but invoke the same GUI interface; if Tcl/Tk is not available, installation continues in text mode.

`-no-gui` Force using the text mode installer.

- lang *LL* Specify the installer interface language as a standard (usually two-letter) code. The installer tries to automatically determine the right language but if it fails, or if the right language is not available, then it uses English as a fallback. Run `install-tl --help` to get the list of available languages.
- portable Install for portable use on, e.g., a USB stick. Also selectable from within the text installer with the `V` command, and from the GUI installer. See section 4.2.
- profile *file* Load the installation profile *file* and do the installation with no user interaction. The installer always writes a file `texlive.profile` to the `tlpkg` subdirectory of your installation. That file can be given as the argument to redo the exact same installation on a different system, for example. Alternatively, you can use a custom profile, most easily created by starting from a generated one and changing values, or an empty file, which will take all the defaults.
- repository *url-or-directory* Specify package repository from which to install; see following.
- in-place (Documented only for completeness: Do not use this unless you know what you are doing.) If you already have an rsync, svn, or other copy of T<sub>E</sub>X Live (see <https://tug.org/texlive/acquire-mirror.html>) then this option will use what you've got, as-is, and do only the necessary post-install actions. Be warned that the file `tlpkg/texlive.tlpdb` may be overwritten; saving it is your responsibility. Also, package removal has to be done manually. This option cannot be toggled via the installer interface.

### 3.3.1 The `-repository` option

The default network package repository is a CTAN mirror chosen automatically via <https://mirror.ctan.org>.

If you want to override that, the location value can be a url starting with `ftp:`, `http:`, `https:`, `file:/`, or a plain directory path. (When giving an `http:`, `https:` or `ftp:` location, trailing `'/'` characters and/or a trailing `'/tlpkg'` component are ignored.)

For example, you could choose a particular CTAN mirror with something like: <http://ctan.example.org/tex-archive/systems/texlive/tlnet/>, substituting a real host-name and its particular top-level CTAN path for `ctan.example.org/tex-archive`. The list of CTAN mirrors is maintained at <https://ctan.org/mirrors>.

If the given argument is local (either a path or a `file:/` url), compressed files in an `archive` subdirectory of the repository path are used (even if uncompressed files are available as well).

## 3.4 Post-install actions

Some post-installation may be required.

### 3.4.1 Environment variables for Unix

If you elected to create symlinks in standard directories (described in section 3.2.4), then there is no need to edit environment variables. Otherwise, on Unix systems, the directory of the binaries for your platform must be added to the search path. (On Windows, the installer takes care of this.)

Each supported platform has its own subdirectory under `TEXDIR/bin`. See figure 5 for the list of subdirectories and corresponding platforms.

Optionally, you can also add the documentation `man` and `Info` directories to their respective search paths, if you want the system tools to find them. The `man` pages might be found automatically after the addition to `PATH`.

For Bourne-compatible shells such as `bash`, and using Intel x86 GNU/Linux and the `TeX Live` 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/2024/bin/x86_64-linux:$PATH; export PATH
MANPATH=/usr/local/texlive/2024/texmf-dist/doc/man:$MANPATH; export MANPATH
INFOPATH=/usr/local/texlive/2024/texmf-dist/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/2024/bin/x86_64-linux:$PATH
setenv MANPATH /usr/local/texlive/2024/texmf-dist/doc/man:$MANPATH
setenv INFOPATH /usr/local/texlive/2024/texmf-dist/doc/info:$INFOPATH
```

If you're not on an `x86_64-linux` platform, use the appropriate platform name; similarly, if you didn't install in the default directory, change the directory name. The `TeX Live` installer reports the full lines to use at the end of installation.

If you already have `PATH` settings somewhere in your startup files, merge in the `TeX Live` directories as you see fit.

### 3.4.2 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/2024/bin/x86_64-linux \
            /usr/local/texlive/2024/texmf-dist/doc/man
```

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

In each (Unix) binary directory, we also create a symbolic link named `man` to the directory `texmf-dist/doc/man`. Some `man` programs, such as the standard macOS `man`, will automatically find that, obviating the need for any `man` page setup.

### 3.4.3 Internet updates after DVD installation

If you installed `TeX Live` from DVD and then wish to get updates from the Internet, you need to run this command—*after* you've updated your search path (as described in the previous section):

```
> tlmgr option repository https://mirror.ctan.org/systems/texlive/tlnet
```

This tells `tlmgr` to use a nearby CTAN mirror for future updates. This is done by default when installing from DVD, via the option described in section 3.2.4.

If there are problems with the automatic mirror selection, you can specify a particular CTAN mirror from the list at <https://ctan.org/mirrors>. Use the exact path to the `tlnet` subdir on that mirror, as shown above.

### 3.4.4 System font configuration for XeTeX and LuaTeX

XeTeX and LuaTeX can use any font installed on the system, not just those in the TeX trees. Such system fonts (not part of TeX Live) are usually accessed by giving the font name, e.g., ‘`Liberation Serif`’, although the system filename can also be used.

A related issue is making the fonts from the TeX Live distribution available as system fonts, which will in turn make them available by font name.

For LuaTeX: for access by font name, nothing special need be done. All fonts in TeX Live should be equally accessible by either font name or filename to LuaTeX, via the `luaotfload` package, which supports both L<sup>A</sup>TeX and plain TeX. The `luaotfload` font name index may need to be rebuilt for new fonts; this is triggered automatically when trying to load a font which isn’t known yet.

For XeTeX: on Windows, fonts shipped with TeX Live are automatically made available (by running the `fc-cache` program provided for Windows as part of TeX Live). For the Mac, you’ll need to consult other documentation. For Unix systems other than macOS, the procedure is the following.

When the `xetex` package is installed (either at initial installation or later), the necessary configuration file is created in `TEXMFSYSVAR/fonts/conf/texlive-fontconfig.conf`. To make the TeX Live fonts available as system fonts,

1. Copy that `texlive-fontconfig.conf` file to (typically) `/etc/fonts/conf.d/09-texlive.conf`.
2. Run `fc-cache -fsv`.

If you do not have sufficient privileges to carry out the steps above, or if you want to make the TeX Live fonts available to only one user, you can do the following:

1. Copy the `texlive-fontconfig.conf` file to (typically) `~/.fonts.conf.d/09-texlive.conf`, where `~` is your home directory.
2. Run `fc-cache -fv`.

You can run `fc-list` to see the names of the available system fonts. The incantation `fc-list : family style file spacing` (all those arguments are literal strings) shows some generally interesting information.

### 3.4.5 ConTeXt LMTX and MKIV

Both the ‘old’ ConTeXt (Mark IV or MkIV) and the ‘new’ ConTeXt (LMTX or MkXL) should run out of the box after TeX Live installation, and should need no special attention as long as you stick to using `tlmgr` for updates.

ConTeXt does not use the Kpathsea library or the system `ls-R` files; instead, ConTeXt maintains a per-user filename database for all known trees. Running `mktexlsr` won’t update these databases directly, but ConTeXt will automatically rebuild them the next time it runs. (The TeX Live-specific files `cont-sys.mkiv` and `cont-sys.mkxl` implement this.)

If you ever want to manually rebuild these databases, run either or both of:

```
mtxrun --generate          # LMTX
mtxrun --luatex --generate # MkIV
```

The resulting files are stored under `TEXMFCACHE`, whose default value in T<sub>E</sub>X Live is `TEXMFSYSVAR;TEXMFVAR`.

ConT<sub>E</sub>Xt will read from all paths mentioned in `TEXMFCACHE`, and write to the first path that is writable. While reading, the last found match will take precedence in the case of duplicated cache data.

For more information, see <https://wiki.contextgarden.net/LMTX> and [https://wiki.contextgarden.net/Running\\_Mark\\_IV](https://wiki.contextgarden.net/Running_Mark_IV).

### 3.4.6 Integrating local and personal macros

This is already mentioned implicitly in section 2.3: `TEXMFLOCAL` (by default, `/usr/local/texlive/texmf-local` or `%SystemDrive%\texlive\texmf-local` on Windows) 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 T<sub>E</sub>X Directory Structure (TDS) subdirectories; see <https://tug.org/tds> or consult `texmf-dist/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 `mktxlsr` or use the ‘Update filename database’ button on the ‘Actions’ tab of the T<sub>E</sub>X Live Manager GUI.

By default, each of these variables is defined to be a single directory, as shown. This is not a hard-and-fast requirement. If you need to easily switch back and forth between different versions of large packages, for example, you can maintain multiple trees for your own purposes. This is done by setting `TEXMFHOME` to the list of directories, within braces, separated by commas:

```
TEXMFHOME = {/my/dir1,/mydir2,/a/third/dir}
```

Section 7.1.5 describes brace expansion further.

### 3.4.7 Integrating third-party fonts

This is unfortunately a messy topic for T<sub>E</sub>X and pdfT<sub>E</sub>X. Forget about it unless you want to delve into many details of the T<sub>E</sub>X installation. Many fonts are included in T<sub>E</sub>X Live already, so take a look if you like; the web pages under <https://tug.org/FontCatalogue> display nearly all of the text fonts included in the main T<sub>E</sub>X distributions, categorized in various ways.

If you do want to install your own fonts, see <https://tug.org/fonts/fontinstall.html> for our best effort at describing the procedure.



Also consider using XeTeX or LuaTeX (see section 2.4), which let you use operating system fonts without any installation in TeX. (But beware that using system fonts usually makes your document sources unusable by anyone in a different environment.)

### 3.5 Testing the installation

After installing TeX Live, you naturally want to test it out, so you can start creating beautiful documents and/or fonts.

One thing you may immediately be looking for is a front-end with which to edit files. TeX Live installs TeXworks (<https://tug.org/texworks>) on Windows (only), and MacTeX installs TeXShop (<https://pages.uoregon.edu/koch/texshop>). On other Unix systems, it's left up to you to choose an editor. There are many choices available, some of which are listed in the next section; see also <https://tug.org/interest.html#editors>. Any plain text editor will work; something TeX-specific is not required.

The rest of this section gives some basic procedures for testing that the new system is functional. We give Unix commands here; under macOS 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 3.14159265 (TeX Live ...)
Copyright ... D.E. Knuth.
...
```

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. 18.

2. Process a basic L<sup>A</sup>TeX file, generating PDF:

```
> pdflatex sample2e.tex
This is pdfTeX 3.14...
...
Output written on sample2e.pdf (3 pages, 142120 bytes).
Transcript written on sample2e.log.
```

If this fails to find `sample2e.tex` or other files, you may have interference from old environment variables or configuration files; we recommend unsetting all TeX-related environment variables for a start. (For a deep analysis, you can ask TeX to report on exactly what it is searching for, and finding; see "Debugging actions" on page 39.)

3. Preview the PDF file, for example:

```
> xpdf sample2e.pdf
```

You should see a new window with a nice document explaining some of the basics of L<sup>A</sup>TeX. (It's well worth reading, by the way, if you're new to TeX.)

Of course there are many other PDF viewers; on Unix systems, `evince` and `okular` are commonly used. For Windows, we recommend trying Sumatra PDF (<https://www.sumatrapdfreader.org/free-pdf-reader.html>). No PDF viewers are included in TeX Live, so you must install whatever you want to use separately.

4. Of course you can still generate TeX's original DVI format:

```
> latex sample2e.tex
```



5. And preview the DVI online:

```
> xdvi sample2e.dvi    # Unix
> dviout sample2e.dvi # Windows
```

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'.

6. To create a PostScript file from the DVI:

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

7. Or to create PDF from the DVI, an alternate path to using pdfTeX (or XeTeX or LuaTeX) which can be useful sometimes:

```
> dvi2pdf sample2e.dvi -o sample2e.pdf
```

8. Other 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 TeX.

`story.tex` The most canonical (plain) TeX test file of all. You must type '\bye' to the \* prompt after 'tex story.tex'.

9. If you have installed the `xetex` package, you can test its access to system fonts as follows:

```
> xetex opentype-info.tex
This is XeTeX, Version 3.14...
...
```

Output written on `opentype-info.pdf` (1 page).

Transcript written on `opentype-info.log`.

If you get an error message saying "Invalid fontname 'Latin Modern Roman/ICU'...", then you need to configure your system so that the fonts shipped with TeX Live can be found. See section 3.4.4.

## 3.6 Uninstalling TeX Live

To uninstall TeX Live (after a successful installation; for Windows, see below):

```
> tlmgr uninstall --all
```

You will be asked for confirmation, else nothing will be done. (Without `--all`, the `uninstall` action is used to remove individual packages.)

This does not remove user-specific directories, namely (see also section 2.3):

`TEXMFCONFIG` This is intended for user configuration changes. If you want to preserve them, be sure you know how to recreate them before removing.

`TEXMFVAR` This is intended to store automatically-generated runtime data, such as local format files. Unless you've used it for other purposes, it should be safe to remove.

`TEXMFHOME` Contains only files you yourself installed, typically those not available in the distributions. Unless you are completely stopping using TeX, or want to start over from scratch, you probably don't want to remove this.

You can find the directory paths for these variables by running `kpsewhich -var-value=var`.

This `tlmgr` uninstallation also does not undo post-install actions, such as `PATH` changes in your shell initialization files and system access to the fonts in `TEX Live` (see section 3.4). You must manually reverse such actions, if desired.

On Windows, uninstallation can be done through the GUI; see section 6.1.

### 3.7 Links for additional downloadable software

If you are new to `TEX`, or otherwise need help with actually writing `TEX` or `LATEX` documents, please visit <https://tug.org/begin.html> for some introductory resources.

Links for some other tools you may consider installing:

**Ghostscript** <https://ghostscript.com/>, a free PostScript and PDF interpreter.

**Perl** <https://perl.org/> with supplementary packages from CPAN, <https://cpan.org/>.

**ImageMagick** <https://imagemagick.org>, 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 alphabetical order (a few here are for Windows only).

- GNU Emacs is available for all major platforms; see <https://www.gnu.org/software/emacs>.
- AUCT<sub>E</sub>X runs under Emacs; it's available through Emacs' package manager ELPA. Sources are also available from CTAN. The AUCT<sub>E</sub>X home page is <https://www.gnu.org/software/auctex>.
- SciTE is available from <https://www.scintilla.org/SciTE.html>.
- Texmaker is free software, available from <https://www.xmlmath.net/texmaker>.
- TeXstudio started as a fork of Texmaker and given additional features; available from <https://texstudio.org/>.
- TeXnicCenter is free software, available from <https://www.texniccenter.org>.
- TeXworks is free software, available from <https://tug.org/texworks> and installed as part of `TEX Live` for Windows (only).
- Vim is free software, available from <https://www.vim.org>.
- WinEdt is shareware available through <https://tug.org/winedt> or <https://www.winedt.com>.
- WinShell is available from <https://www.winshell.de>.

For a much longer list of packages and programs, see <https://tug.org/interest.html>.

## 4 Specialized installations

The previous sections described the basic installation process. Here we turn to some specialized cases.

## 4.1 Shared-user (or cross-machine) installations

T<sub>E</sub>X Live has been designed to be shared between 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 found relative to the programs. You can see this in the principal configuration file `$TEXMFDIST/web2c/texmf.cnf`, which contains lines such as

```
TEXMFROOT = $SELFAUTOPARENT
...
TEXMFDIST = $TEXMFROOT/texmf-dist
...
TEXMFLocal = $SELFAUTOGRANDPARENT/texmf-local
```

This means that adding the directory for T<sub>E</sub>X Live executables for their platform to their search path is sufficient to get a working setup.

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.

For Windows, T<sub>E</sub>X Live includes a launcher `tlaunch`. Its main window contains menu entries and buttons for various T<sub>E</sub>X-related programs and documentation, customizable via an `ini` file. On first use, it replicates the usual Windows-specific post-install, *i.e.*, search path modification and file associations, but only for the current user. Therefore, workstations with access to T<sub>E</sub>X Live on the local network only need a menu shortcut for the launcher. See the `tlaunch` manual (`texdoc tlaunch`, or <https://ctan.org/pkg/tlaunch>).

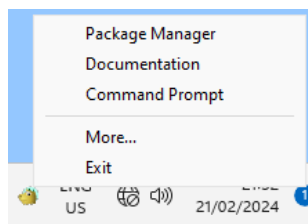
## 4.2 Portable (USB) installations

The `-portable` installer option (or `V` command in the text installer or corresponding GUI option) creates a completely self-contained T<sub>E</sub>X Live installation under a common root and forgoes system integration. You can create such an installation directly on a USB stick, or copy it to a USB stick afterwards.

Technically, the portable installation is made self-contained by setting the default values of `TEXMFHOME`, `TEXMFVAR`, and `TEXMFCONFIG` to be the same as `TEXMFLocal`, `TEXMFSYSVAR`, and `TEXMFSYSCONFIG` respectively; thus, per-user configuration and caches will not be created.

To run T<sub>E</sub>X using this portable installation, you need to add the appropriate binary directory to the search path during your terminal session, as usual.

On Windows, you can double-click `t1-tray-menu` at the root of the installation and create a temporary ‘tray menu’ offering a choice of a few common tasks, as shown in this screenshot:



The ‘More...’ entry explains how you can customize this menu.

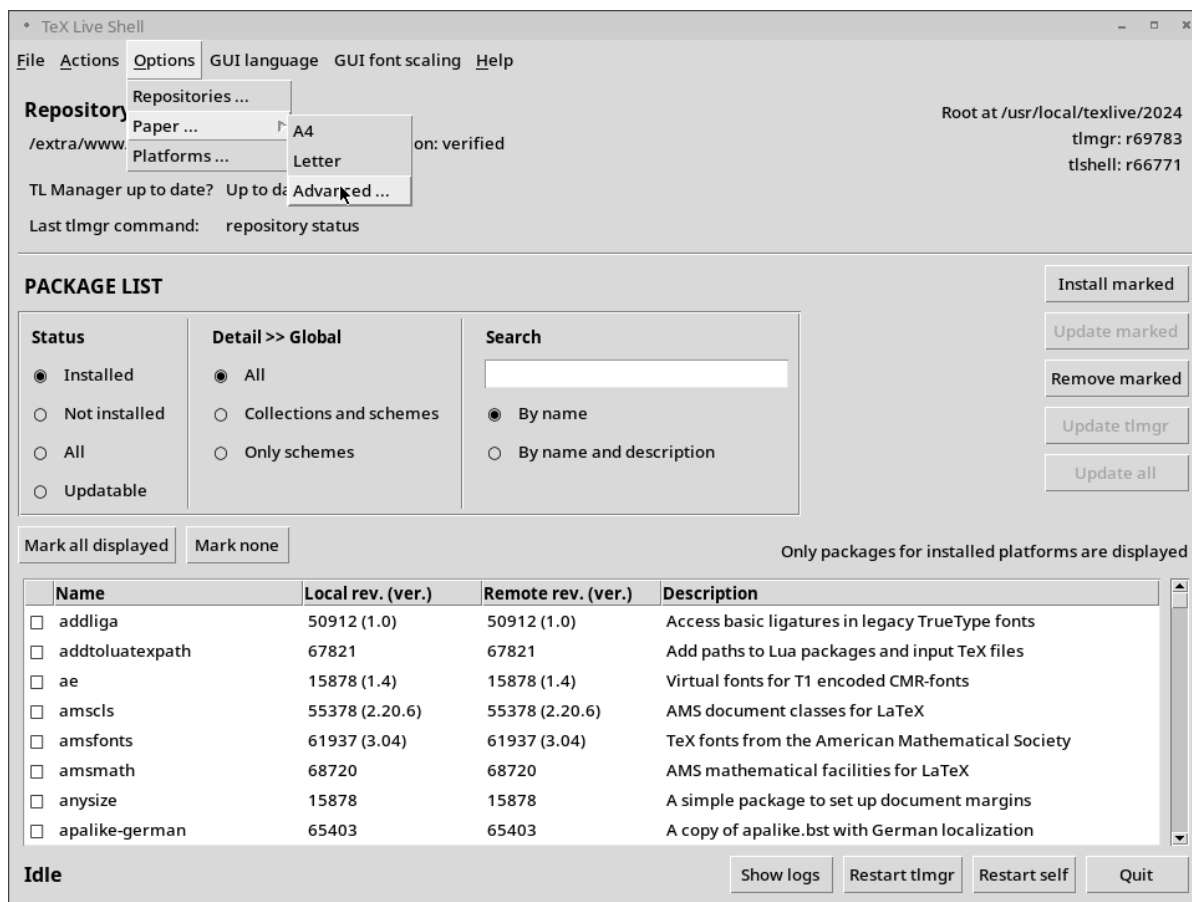


Figure 9: tlshell GUI, showing the Actions menu (GNU/Linux)

## 5 tlmgr: Managing your installation

TeX Live includes a program named `tlmgr` for managing TeX Live after the initial installation. Its capabilities include:

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

`tlmgr`'s functionality completely subsumes the `texconfig` program. We still distribute and maintain `texconfig` for the sake of anyone used to its interface, but we recommend using `tlmgr` nowadays.

### 5.1 GUI interfaces for tlmgr

TeX Live contains several GUI front-ends for `tlmgr`. Two notable ones: (1) Figure 9 shows `tlshell`, which is written in Tcl/Tk and runs out of the box under Windows; (2) Figure 10 shows `tlcockpit`, which requires Java version 8 or higher and JavaFX. Both are separate packages.

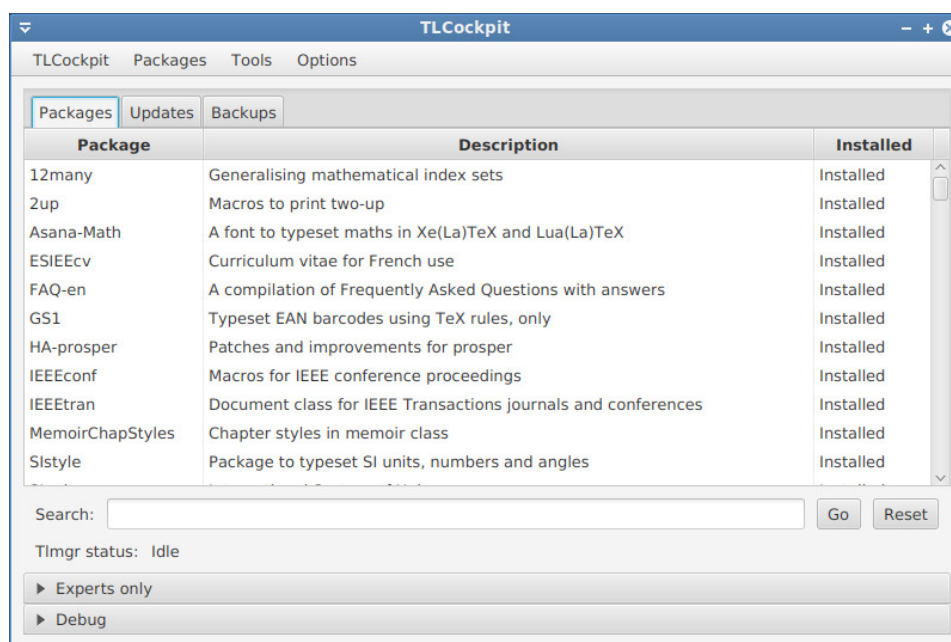


Figure 10: tlmgr GUI for tlmgr

tlmgr also has a native GUI mode (shown in figure 11), which is started with:

```
> tlmgr -gui
```

However, this GUI extension requires Perl/Tk, which module is no longer included in TeX Live's Perl distribution for Windows.

## 5.2 Sample tlmgr command-line invocations

After the initial installation, you can update your system to the latest versions available with:

```
> tlmgr update -all
```

If this makes you nervous, first try

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

or (less verbose):

```
> tlmgr update -list
```

This more complex example adds a collection, for the engine XeTeX, from a local directory:

```
> tlmgr -repository /local/mirror/tlnet install collection-xetex
```

It generates the following output (abridged):

```
install: collection-xetex
install: arabxetex
...
install: xetex
install: xetexconfig
install: xetex.i386-linux
running post install action for xetex
install: xetex-def
...
```

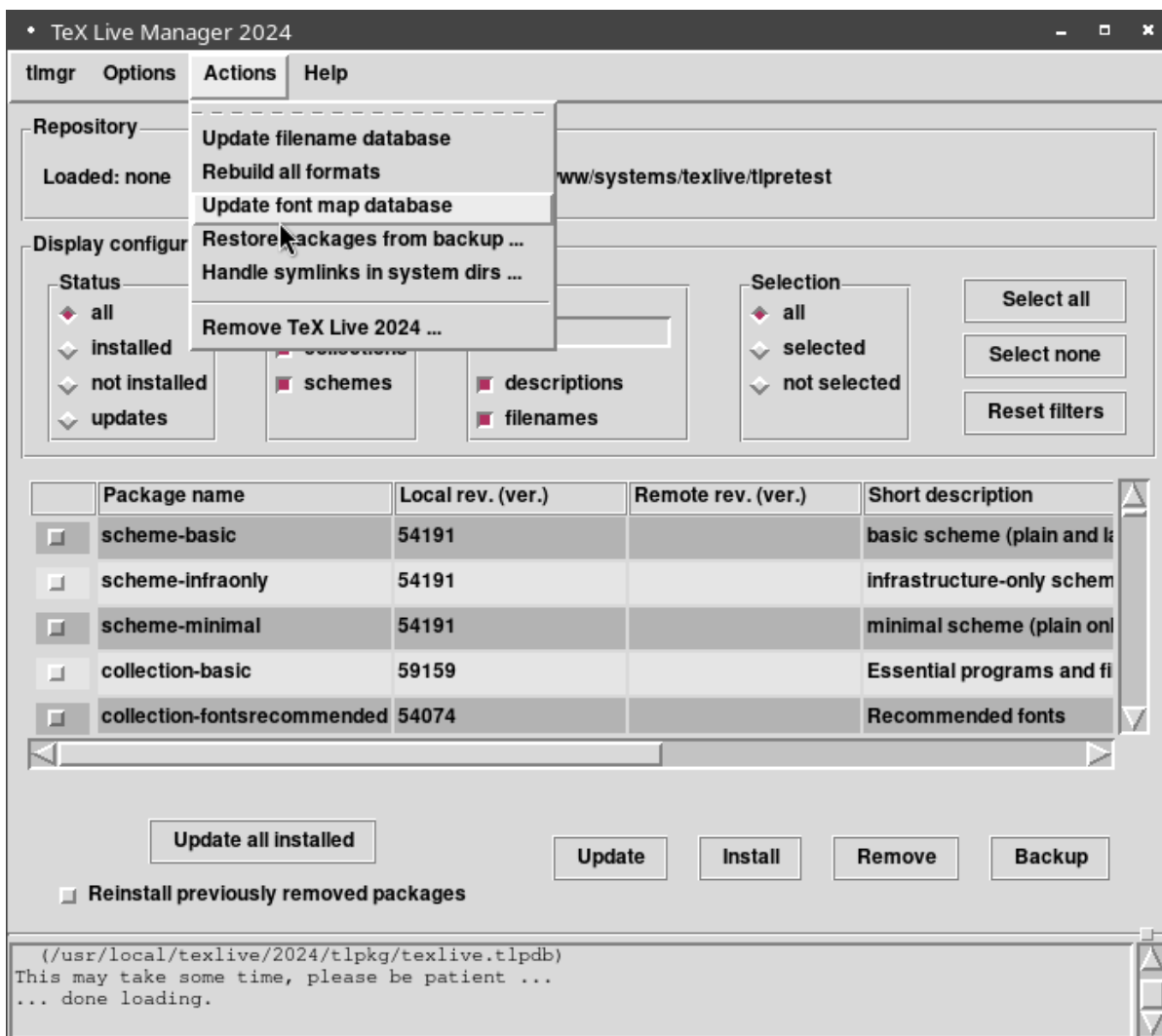


Figure 11: Legacy tlmgr GUI mode: main window, after ‘Load’

```

running mktexlsr
mktexlsr: Updating /usr/local/texlive/2024/texmf-dist/ls-R...
...
running fmtutil-sys --missing
...
Transcript written on xelatex.log.
fmtutil: /usr/local/texlive/2024/texmf-var/web2c/xetex/xelatex.fmt installed.

```

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

To describe a package (or collection or scheme):

```
> tlmgr show collection-latexextra
```

which produces output like this:

```

package:    collection-latexextra
category:   Collection
shortdesc:  LaTeX supplementary packages
longdesc:   A very large collection of add-on packages for LaTeX.
installed:  Yes

```

```
revision: 46963
sizes:    657941k
```

Last and most important, for full documentation see <https://tug.org/texlive/tlmgr.html>, or:

```
> tlmgr -help
```

## 6 Notes on Windows

### 6.1 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. This contains entries for some GUI programs, such as `tlshell` (a GUI for `tlmgr`) and `dviout`, and some entries for documentation.

**File associations.** If enabled, TeXworks and Dviout become either the default program for their respective filetypes, or get an entry in the ‘Open with’ right-click menus of those filetypes. However, higher-priority ‘User Choice’ file associations, which can only be specified interactively, can get in the way.

**PostScript support.** For PostScript files, a PSviewer filetype now converts PostScript to a temporary PDF, which is then displayed by the default PDF viewer. Various bitmapped formats get an entry `bitmap2eps` in their ‘Open with’ right-click menu for converting to EPS, letting `sam2p` or `bmeps` do the real work.

**Automatic path adjustment.** No manual configuration steps are required.

**Uninstaller.** The installer creates an entry for TeX Live, either under ‘Add/Remove Programs’ (admin installation) or under the TeX Live menu (single-user installation).

**Write-protect.** For an admin installation, the TeX Live directories are write-protected, at least if TeX Live is installed on a normal NTFS-formatted non-removable disk. Also, have a look at `tlaunch`, described in section 4.1, for a different approach.

### 6.2 Additional software included on Windows

To be complete, a TeX Live installation needs support programs that are not commonly found on a Windows machine. TeX Live provides the missing pieces. These programs are all installed as part of TeX Live only on Windows.

**Perl, Tcl/Tk and Ghostscript.** Because of the importance of Perl and Ghostscript, and because the installer and `tlshell` GUIs are written in Tcl/Tk, TeX Live includes ‘hidden’ copies of these programs. TeX 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 installations (except for Ghostscript), and shouldn’t interfere with any system installations of Perl, Tcl/Tk or Ghostscript. See subsection 6.3 how to tell TeX Live that you want to use your own external installations for contributed scripts within TeX Live.

**dviout.** Also installed is `dviout`, a DVI viewer. 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 (highly recommended) on-line help.

**TeXworks.** TeXworks is a TeX-oriented editor with an integrated PDF viewer.

**Command-line tools.** A number of Windows ports of common Unix command-line programs are installed along with the usual TeX Live binaries. These include `gzip`, `zip`, `unzip`, and the utilities from the `poppler` suite (`pdffonts`, `pdfinfo`, ...).

**fc-list, fc-cache, ...** The tools from the `fontconfig` library allow XeTeX to find system fonts on Windows. You can use `fc-list` to determine the font names to pass to XeTeX's extended `\font` command. If necessary, run `fc-cache` first to update font information.

### 6.3 Using external Perl, Tcl/Tk and Ghostscript installations

Normally, TeX Live will use its built-in Perl, Tcl/Tk and Ghostscript also for contributed scripts within TeX Live. If you wish to use your own external versions, then you can configure that in the `texmf.cnf` file *in the root of the installation*.

For Perl, you need to add a line

```
TEXLIVE_WINDOWS_TRY_EXTERNAL_PERL = 1
```

TeX Live (specifically, `bin/windows/runscript.tlu`) will then search for `perl.exe` on the search path, except for scripts which belong to the TeX Live infrastructure. This is the most likely setting of the three to be useful; although the TeX Live Perl includes many additional modules, it cannot support all third-party scripts.

Similarly, for Tcl/Tk you need a line

```
TEXLIVE_WINDOWS_TRY_EXTERNAL_TCL = 1
```

TeX Live will then search for `tclkit.exe`, `wish.exe`, `wish85.exe`, `wish86.exe` and `wish87.exe` on the searchpath.

Ghostscript is handled differently, in that you need to specify the filename or the full path of your command-line Ghostscript:

```
TEXLIVE_WINDOWS_EXTERNAL_GS = path of command-line ghostscript
```

Another difference is that the Ghostscript shipped with TeX Live is complete, with only documentation and printer drivers omitted. So it's unlikely you'll have any need to override it.

See also section 7.1.2 about `texmf.cnf`.

### 6.4 User Profile is Home

The Windows counterpart of a Unix home directory is the `%USERPROFILE%` directory. Under Windows Vista and later it is `C:\Users\<username>`. In the `texmf.cnf` file, and Kpathsea in general, `~` will expand appropriately on both Windows and Unix.

### 6.5 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 6.4). 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.

## 6.6 Windows permissions

In later versions of Windows, a distinction is made between regular users and administrators, where only the latter have free access to almost the entire operating system. We have made an effort to make T<sub>E</sub>X Live installable without administrative privileges.

If the installer is started with administrative permissions, there is an option to install for all users. If this option is chosen, shortcuts are created for all users, and the system search path is modified. Otherwise, shortcuts and menu entries are created for the current user, and the user search path is modified.

Regardless of administrator status, the default root of T<sub>E</sub>X Live proposed by the installer is always under %SystemDrive%. 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 search path consists of the system search path followed by the user search path, the new T<sub>E</sub>X Live would never get precedence. As a fallback, the installer creates a shortcut to the command-prompt in which the new T<sub>E</sub>X Live binary directory is prepended to the local search path. The new T<sub>E</sub>X Live will be always usable from within such a command-prompt. The shortcut for T<sub>E</sub>Xworks, if installed, also prepends T<sub>E</sub>X Live to the search path, so it should also be immune to this path problem.

You should be aware that even if you are logged in as administrator, you need to explicitly ask for administrator privileges. In fact, there is not much point in logging in as administrator. Instead, right-clicking on the program or shortcut that you want to run usually gives you a choice 'Run as administrator'.

## 6.7 Increasing maximum memory on Windows and Cygwin

Windows and Cygwin (see section 3.1.4 for Cygwin installation specifics) users may find that they run out of memory when running some of the programs shipped with T<sub>E</sub>X Live. For example, `asy` might run out of memory if you try to allocate an array of 25,000,000 reals, and LuaT<sub>E</sub>X might run out of memory if you try to process a document with a lot of big fonts.

For Cygwin, you can increase the amount of available memory by following the instructions in the Cygwin User's Guide (<https://cygwin.com/cygwin-ug-net/setup-maxmem.html>).

For Windows, you have to create a file, say `moremem.reg`, with these four lines:

```
Windows Registry Editor Version 5.00
```

```
[HKEY_LOCAL_MACHINE\Software\Cygwin]
"heap_chunk_in_mb"=dword:ffffff00
```

and then execute the command `regedit /s moremem.reg` as administrator. (If you want to change memory only for the current user instead of system-wide, use `HKEY_CURRENT_USER`.)

## 7 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 <https://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, Windows, macOS, 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.
- `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.
- `mf` Creating typeface families.
- `mft` Prettyprinting METAFONT source.
- `mpost` Creating technical diagrams.
- `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.

`--version` print version information, then exit.

And most also honor:

`--verbose` print detailed progress report.

For locating files the Web2C programs use the path searching library Kpathsea (<https://tug.org/kpathsea>). This library uses a combination of environment variables and 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.

## 7.1 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 36) 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.

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.

### 7.1.1 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 39).

### 7.1.2 Config files

Kpathsea reads *runtime configuration files* named `texmf.cnf` for search path and other definitions. The search path `TEXMFCNF` is used to look for these files, but we do not recommend setting this (or any) environment variable to override the system directories.

Instead, normal installation results in a file `.../2024/texmf.cnf`. If you must make changes to the defaults (not normally necessary), this is the place to put them. The main configuration file is in `.../2024/texmf-dist/web2c/texmf.cnf`. You should not edit this latter file, as your changes will be lost when the distributed version is updated.

As an aside, if you merely wish to add a personal directory to a particular search path, setting an environment variable is a reasonable method:

```
TEXINPUTS=./my/macro/dir:
```

To keep the setting maintainable and portable over the years, use a trailing `‘:’` (`‘;’` on Windows) to insert the system paths, instead of trying to write them all out explicitly (see section 7.1.4). Another option is to use the `TEXMFHOME` tree (see section 3.2.3).

All `texmf.cnf` files in the search path will be read and definitions in earlier files override those in later files. For example, with a search path of `.:$TEXMF`, values from `./texmf.cnf` override those from `$TEXMF/texmf.cnf`.

- Comments start with `%`, either at the beginning of a line or preceded by whitespace, 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 [.prognam] [=] value
```

where the `‘=’` and surrounding whitespace are optional. (But if `value` begins with `‘.’`, it is simplest to use the `‘=’` to avoid the period being interpreted as the program name qualifier.)

- The `variable` name may contain any character other than whitespace, `‘=’`, or `‘.’`, but sticking to `‘A-Za-z_’` is safest.
- If `‘.prognam’` is present, the definition only applies if the program that is running is named `prognam` or `prognam.exe`. This allows different flavors of `TeX` to have different search paths, for example.
- Considered as strings, `value` may contain any character. However, in practice most `texmf.cnf` values are related to path expansion, and since various special characters are used in expansion (see section 7.1.7), such as braces and commas, they cannot be used in directory names.

A `‘;’` in `value` is translated to `‘:’` if running under Unix, in order to have a single `texmf.cnf` that can support both Unix and Windows systems. This translation happens with any value, not just search paths, but fortunately in practice `‘;’` is not needed in other values.

The `$var.prog` feature is not available on the right-hand side; instead, you must use an additional variable.

- 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;}//

```

### 7.1.3 Path expansion

Kpathsea recognizes certain special characters and constructions in search paths, similar to those available in Unix shells. As a general example, the 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.

### 7.1.4 Default expansion

If the highest-priority search path (see “Path sources” on page 33) 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 ‘:’.

### 7.1.5 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`. In the distributed `texmf.cnf`, a definition like this (simplified for this example) is made:

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

We then use this to define, for example, the T<sub>E</sub>X input path:

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

which means that, after looking in the current directory, the `$TEXMFVAR/tex`, `$TEXMFHOME/tex`, `$TEXMFLOCAL/tex` and `$TEXMFDIST/tex` trees will be searched (the last two using `ls-R` data base files).

### 7.1.6 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.

### 7.1.7 Summary of special characters in `texmf.cnf` files

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

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

Exactly when a character will be considered special or act as itself depends on the context in which it is used. The rules are inherent in the multiple levels of interpretation of the configuration (parsing, expansion, search, ...) and so cannot be concisely stated, unfortunately. There is no general escape mechanism; in particular, ‘\’ is not an “escape character” in `texmf.cnf` files.

When it comes choosing directory names for installation, it is safest to avoid them all.

## 7.2 Filename databases

Kpathsea goes to some lengths to minimize disk accesses for searches. Nevertheless, in the standard T<sub>E</sub>X Live, or at any installation with enough directories, searching every possible directory for a given file will take an excessively long time. 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`.

### 7.2.1 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 T<sub>E</sub>X hierarchy in your installation that you wish to be searched



( $\$TEXMF$  by default). 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  $\LaTeX$  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.

### 7.2.2 `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  $\TeX$  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-formats` 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.

`--progname=name`

Set the program name to *name*. This can affect the search paths via the `.progname` 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*.

### 7.2.3 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` (TeX source file directories). We find it in the subdirectory `tex/latex/base` below the ‘`texmf-dist`’ TeX 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 BibTeX 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 TeX 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 TeX 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
utmb08r 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.

#### 7.2.4 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 the `dvips` reference manual). The output (slightly rearranged) appears in Figure 12.

`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 7.2.3 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
```

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//).
```

```

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 12: Finding configuration files

```
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]

```

### 7.3 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>C<sub>T</sub>E<sub>X</sub>.
- `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; its default value is 600000.

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.

### 7.4 \$TEXMFDOTDIR

In various places above, we gave various search paths starting with `.` (to search the current directory first), as in

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

This is a simplification. The `texmf.cnf` file we distribute in T<sub>E</sub>X Live uses `$TEXMFDOTDIR` instead of just `'.'`, as in:

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

(In the distributed file, the second path element is also slightly more complicated than `$TEXMF/tex//`. But that's minor; here we want to discuss the `$TEXMFDOTDIR` feature.)

The reason to use the variable `$TEXMFDOTDIR` in the path definitions instead of simply `'.'` is purely so that it can be overridden. For example, a complex document may have many source files arranged in many subdirectories. To handle that, you can set `TEXMFDOTDIR` to `./` (for example, in the environment when you build the document) and they will all get searched. (Warning: don't use `./` by default; it's usually highly undesirable, and potentially insecure, to search through all subdirectories for an arbitrary document.)

As another example, you may wish not to search the current directory at all, e.g., if you have arranged for all the files to be found via explicit paths. You can set `$TEXMFDOTDIR` to, say, `/nonesuch` or any nonexistent directory for this.

The default value of `$TEXMFDOTDIR` is just `'.'`, as set in our `texmf.cnf`.

## 8 Acknowledgements

TeX Live is a joint effort by virtually all of the TeX user groups. This edition of TeX Live was overseen by Karl Berry. The other principal contributors, past and present, are listed below.

- The English, German, Dutch, and Polish TeX user groups (TUG, DANTE e.V., NTG, and GUST, respectively), which provide the necessary technical and administrative infrastructure. Please join the TeX user group near you! (See <https://tug.org/usergroups.html>.)
- The CTAN team (<https://ctan.org>), which distributes the TeX Live images and provides the common infrastructure for package updates, upon which TeX Live depends.
- Nelson Beebe, for making many platforms available to TeX Live developers, and his own comprehensive testing and unparalleled bibliographic efforts.
- John Bowman, for making many changes to his advanced graphics program Asymptote to make it work in TeX Live.
- Peter Breitenlohner and the  $\varepsilon$ -TeX team for the stable foundation of future TeX's, and Peter specifically for years of stellar help with GNU autotools and keeping sources up to date. Peter passed away in October 2015, and we dedicate the continuing work to his memory.
- Jin-Hwan Cho and all of the DVIPDFMx team, for their excellent driver and responsiveness to configuration issues.
- Thomas Esser, without whose marvelous `teTeX` package TeX Live would have never existed.
- Michel Goossens, who co-authored the original documentation.
- Eitan Gurari, whose TeX4ht is used to create the HTML version of this documentation, and who worked tirelessly to improve it at short notice, every year. Eitan prematurely passed away in June 2009, and we dedicate this documentation to his memory.
- Hans Hagen, for much testing and making his ConTeXt package (<https://pragma-ade.com>) work within TeX Live's framework, and continually driving TeX development.
- Hàn Thế Thành, Martin Schröder, and the pdfTeX team (<http://pdftex.org>), for continuing enhancements of TeX's abilities.
- Hartmut Henkel, for significant development contributions to pdfTeX, LuaTeX, and more.
- Shunshaku Hirata, for much original and continuing work on DVIPDFMx.
- Taco Hoekwater, for major renewed development efforts on MetaPost and (Lua)TeX (<http://luatex.org>) itself, incorporating ConTeXt into TeX Live, giving Kpathsea multi-threaded functionality, and much more.
- Khaled Hosny, for substantial work on XeTeX, DVIPDFMx, and efforts with Arabic and other fonts.
- Paweł Jackowski, for the Windows installer `tlpm`, and Tomasz Łuczak, for `tlpmgui`, used in past releases.
- Akira Kakuto, for providing the Windows binaries from his W32TEX and W64TEX distributions for Japanese TeX (<http://w32tex.org>), and many other development contributions.



- Jonathan Kew, for developing the remarkable XeTeX engine and taking the time and trouble to integrate it in TeX Live, as well as the initial version of the MacTeX installer, and also for our recommended front-end TeXworks.
- Hironori Kitagawa, for maintenance of (e)pTeX and related support.
- Dick Koch, for maintaining MacTeX (<https://tug.org/mactex>) in very close tandem with TeX Live, and for his great good cheer in doing so.
- Reinhard Kotucha, for major contributions to the TeX Live 2008 infrastructure and installer, as well as Windows research efforts, the `getnonfreefonts` script, and more.
- Siep Kroonenberg, also for major contributions to the TeX Live 2008 infrastructure and installer, especially on Windows, and for the bulk of work updating this manual describing those features.
- Clerk Ma, for engine bug fixes and extensions.
- Mojca Miklavec, for much help with ConTeXt, building many binary sets, and plenty more.
- Heiko Oberdiek, for the `epstopdf` package and many others, compressing the huge `pst-geo` data files so we could include them, and most of all, for his remarkable work on `hyperref`.
- Phelype Oleinik, for the group-delimited `\input` across engines in 2020, and more.
- 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 `texdoc` development.
- Fabrice Popineau, for the original Windows support in TeX Live and work on the French documentation.
- Norbert Preining, the principal architect of the current TeX Live infrastructure and installer, and also for coordinating the Debian version of TeX Live (together with Frank Küster), and doing so much work along the way.
- Sebastian Rahtz, for originally creating TeX Live and maintaining it for many years. Sebastian passed away in March 2016, and we dedicate the continuing work to his memory.
- Luigi Scarso, for continuing development of MetaPost, LuaTeX, and much more.
- Andreas Scherer, for `cwebbin`, the CWEB implementation used in TeX Live, and continuing maintenance of the original CWEB.
- Takuji Tanaka, for maintenance of (e)(u)pTeX and related support.
- Tomasz Trzeciak, for wide-ranging help with Windows.
- Vladimir Volovich, for substantial help with porting and other maintenance issues, and especially for making it feasible to include `xindy`.
- Staszek Wawrykiewicz, a principal tester for all of TeX Live, and coordinator of the many major Polish contributions: fonts, Windows installation, and more. Staszek passed away in February 2018, and we dedicate the continuing work to his memory.
- Olaf Weber, for his patient maintenance of Web2C in past years.
- Gerben Wierda, for creating and maintaining the original macOS support.
- Graham Williams, the originator of the TeX Catalogue.
- Joseph Wright, for much work on making the same primitive functionality available across engines.

- Hironobu Yamashita, for much work on p $\TeX$  and related support.

Builders of the binaries: Marc Baudoin (`amd64-netbsd`, `i386-netbsd`), Ken Brown (`x86_64-cygwin`), Akira Kakuto (`windows`), Dick Koch (`universal-darwin`), Mojca Miklavc (`x86_64-darwinlegacy`), Norbert Preining (`aarch64-linux`, `armhf-linux`, `i386-freebsd`, `amd64-freebsd`, `i386-linux`, `x86_64-linux`, `i386-solaris`, `x86_64-solaris`, `x86_64-linuxmusl`). For information on the  $\TeX$  Live build process, see <https://tug.org/texlive/build.html>.

Translators of this manual: Takuto Asakura (Japanese), Denis Bitouzé & Patrick Bideault (French), Carlos Enriquez Figueras (Spanish), Jjgod Jiang, Jinsong Zhao, Yue Wang, & Helin Gai (Chinese), Nikola Lečić (Serbian), Marco Pallante & Carla Maggi (Italian), Petr Sojka & Jan Busa (Czech/Slovak), Boris Veytsman & Vladimir Lomov (Russian), Uwe Ziegenhagen (German). The  $\TeX$  Live documentation web page is <https://tug.org/texlive/doc.html>.

Of course the most important acknowledgement must go to Donald Knuth, first for inventing  $\TeX$ , and then for giving it to the world.

## 9 Release history

### 9.1 Past

Discussion began in late 1993 when the Dutch  $\TeX$  Users Group was starting work on its 4All $\TeX$  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 4All $\TeX$  CD, but also the TUG Technical Council working group on a  *$\TeX$  Directory Structure* (<https://tug.org/tds>), which specified how to create consistent and manageable collections of  $\TeX$  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 4All $\TeX$  CD showed that Unix users would benefit from a similarly easy system, and this is the other main strand of  $\TeX$  Live.

We first undertook to make a new Unix-based TDS CD in the autumn of 1995, and quickly identified Thomas Esser's te $\TeX$  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 te $\TeX$ , and we decided to base the 2nd edition of the CD on the standard Web2C, with the addition of te $\TeX$ '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 te $\TeX$  was being made, and  $\TeX$  Live included almost all of its features. The 4th edition followed the same pattern, using a new version of te $\TeX$ , and a new release of Web2C (7.3). The system now included a complete Windows setup, thanks to Fabrice Popineau.

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  $\TeX$  Live 5 was that all non-free software was removed. Everything in  $\TeX$  Live is now intended to be compatible with the Debian Free Software Guidelines (<https://www.debian.org/doc/faq/free-software.html>):

[//debian.org/intro/free](http://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 macOS support, and the usual myriad of updates to all sorts of packages and programs. An important goal was integration of the source back with  $\text{\TeX}$ , to correct the drift apart in versions 5 and 6.

### 9.1.1 2003

In 2003, with the continuing flood of updates and additions, we found that  $\text{\TeX}$  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. 6). In addition:

- At the request of the  $\text{\LaTeX}$  team, we changed the standard `latex` and `pdflatex` commands to now use  $\epsilon$ - $\text{\TeX}$  (see p. 8).
- 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  $\text{\TeX}$  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`.
- $\text{\TeX}$ , `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  $\text{\TeX}$  Live 7, this translation was dependent on the system locale settings; now, locale settings do not influence the  $\text{\TeX}$  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:  $\text{\TeX}$  Live 2003.

### 9.1.2 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 `TEXFONTPATHS` 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 <https://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. 5).
- 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. 6), 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-dist/doc/generic/tds/tds.html#Extensions`.
- Helper scripts (not meant to be invoked by users) are now located in a new `scripts` subdirectory of `texmf*` trees, and can be searched for via `kpsewhich -format=texmfscripts`. So if you have programs which call such scripts, they'll need to be adjusted. See <https://tug.org/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 T<sub>E</sub>X (only 32–126 printable), ConT<sub>E</sub>Xt (0–255 printable), and the  $\Omega$ -related formats. This default behavior is almost the same as in T<sub>E</sub>X Live 2003, but it's implemented more cleanly, with more possibilities for customization. See `texmf-dist/doc/web2c/web2c.html#TCX-files`. (By the way, with Unicode input, T<sub>E</sub>X 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 `pdftex` are available in L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt, etc., as well as the  $\varepsilon$ -T<sub>E</sub>X features (`texmf-dist/doc/etex/base/`). It also means it's *more important than ever* to use the `ifpdf` package (works with both plain and L<sup>A</sup>T<sub>E</sub>X) 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.
- pdfT<sub>E</sub>X (<http://pdftex.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-pdftex/2004-May/000504.html>
  - All parameters previously set through the special configuration file `pdftex.cfg` must now be set through primitives, typically in `pdftexconfig.tex`; `pdftex.cfg` is no longer supported. Any extant `.fmt` files must be redumped when `pdftexconfig.tex` is changed.
  - See the pdfT<sub>E</sub>X manual for more: `texmf-dist/doc/pdftex/manual/pdftex-a.pdf`.
- 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-dist/doc/web2c`.

- `encTeX` support is now included within Web2C and consequently all `TeX` programs, via the `-enc` option — *only when formats are built*. `encTeX` supports general re-encoding of input and output, enabling full support of Unicode (in UTF-8). See `texmf-dist/doc/generic/encTeX/` and <http://olsak.net/encTeX.html>.
- Aleph, a new engine combining  $\epsilon$ -`TeX` and  $\Omega$ , is available. Some information is available in `texmf-dist/doc/aleph/base` and <https://texfaq.org/FAQ-enginedev>. The  $\LaTeX$ -based format for Aleph is named `lamed`.
- The latest  $\LaTeX$  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`.
- `dvipng`, a new program for converting DVI to PNG image files, is included. See <https://ctan.org/pkg/dvipng>.
- 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 (<https://ctan.org/pkg/cbgreek-complete>).
- `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.

### 9.1.3 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. 7.



- 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.

#### 9.1.4 2006–2007

In 2006–2007, the major new addition to  $\TeX$  Live was the  $\XeTeX$  program, available as the `xetex` and `xelatex` programs; see <https://scripts.sil.org/xetex>.

MetaPost also received a notable update, with more planned for the future (<https://tug.org/metapost/articles>), likewise `pdf $\TeX$`  (<https://tug.org/applications/pdftex>).

The  $\TeX$  `.fmt` (high-speed format) and the similar files for MetaPost and `META-FONT` 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  $\TeX$ . ( $\LaTeX$  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 (<https://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 `te $\TeX$`  (<https://tug.org/tetex>). As a result, there was a surge of interest in  $\TeX$  Live, especially among GNU/Linux distributors. (There is a new `tetex` installation scheme in  $\TeX$  Live, which provides an approximate equivalent.) We hope this will eventually translate to improvements in the  $\TeX$  environment for everyone.

### 9.1.5 2008

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 5) 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.

Finally, the changes made by Donald Knuth in his ‘T<sub>E</sub>X tuneup of 2008’ are included in this release. See <https://tug.org/TUGboat/Articles/tb29-2/tb92knut.pdf>.

### 9.1.6 2009

In 2009, the default output format for Lua(L<sup>A</sup>)T<sub>E</sub>X is now PDF, to take advantage of LuaT<sub>E</sub>X’s OpenType support, et al. New executables named `dviluatex` and `dvilualatex` run LuaT<sub>E</sub>X with DVI output. The LuaT<sub>E</sub>X home page is <http://luatex.org>.

The original Omega engine and Lambda format have been excised, after discussions with the Omega authors. The updated Aleph and Lamed remain, as do the Omega utilities.

A new release of the AMS Type 1 fonts is included, including Computer Modern: a few shape changes made over the years by Knuth in the Metafont sources have been integrated, and the hinting has been updated. The Euler fonts have been thoroughly reshaped by Hermann Zapf (see <https://tug.org/TUGboat/Articles/tb29-2/tb92hagen-euler>).



pdf). In all cases, the metrics remain unchanged. The AMS fonts home page is <https://ams.org/tex/amsfonts.html>.

The new GUI front end T<sub>E</sub>Xworks is included for Windows, and also in MacT<sub>E</sub>X. For other platforms, and more information, see the T<sub>E</sub>Xworks home page, <https://tug.org/texworks>. It is a cross-platform front-end inspired by the macOS TeXShop editor, aiming at ease-of-use.

The graphics program Asymptote is included for several platforms. This implements a text-based graphics description language vaguely akin to MetaPost, but with advanced 3D support and other features. Its home page is <https://asymptote.sourceforge.io>.

The separate `dvipdfm` program has been replaced by `dvipdfmx`, which operates in a special compatibility mode under that name. `dvipdfmx` includes CJK support and has accumulated many other fixes over the years since the last `dvipdfm` release.

Executables for the `cygwin` and `i386-netbsd` platforms are now included, while we were advised that OpenBSD users get T<sub>E</sub>X through their package systems, plus there were difficulties in making binaries that have a chance of working on more than one version.

A miscellany of smaller changes: we now use `xz` compression, the stable replacement for `lzma` (<https://tukaani.org/xz/>); a literal `$` is allowed in filenames when it does not introduce a known variable name; the Kpathsea library is now multi-threaded (made use of in MetaPost); the entire T<sub>E</sub>X Live build is now based on Automake.

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>.

### 9.1.7 2010

In 2010, the default version for PDF output is now 1.5, enabling more compression. This applies to all the T<sub>E</sub>X engines when used to produce PDF and to `dvipdfmx`. Loading the `pdf14` L<sup>A</sup>T<sub>E</sub>X package changes back to PDF 1.4, or set `\pdfminorversion=4`.

`pdf(LA)TEX` now *automatically* converts a requested Encapsulated PostScript (EPS) file to PDF, via the `epstopdf` package, when and if the L<sup>A</sup>T<sub>E</sub>X `graphics.cfg` configuration file is loaded, and PDF is being output. The default options are intended to eliminate any chance of hand-created PDF files being overwritten, but you can also prevent `epstopdf` from being loaded at all by putting `\newcommand{\DoNotLoadEpstopdf}{}` (or `\def...`) before the `\documentclass` declaration. It is also not loaded if the `pst-pdf` package is used. For more details, see the `epstopdf` package documentation (<https://ctan.org/pkg/epstopdf-pkg>).

A related change is that execution of a very few external commands from T<sub>E</sub>X, via the `\write18` feature, is now enabled by default. These commands are `repstopdf`, `makeindex`, `kpsewhich`, `bibtex`, and `bibtex8`; the list is defined in `texmf.cnf`. Environments which must disallow all such external commands can deselect this option in the installer (see section 3.2.4), or override the value after installation by running `tlmgr conf texmf shell_escape 0`.

Yet another related change is that BIBT<sub>E</sub>X and Makeindex now refuse to write their output files to an arbitrary directory (like T<sub>E</sub>X itself), by default. This is so they can now be enabled for use by the restricted `\write18`. To change this, the `TEXMFOUTPUT` environment variable can be set, or the `openout_any` setting changed.

XeT<sub>E</sub>X now supports margin kerning along the same lines as pdfT<sub>E</sub>X. (Font expansion is not presently supported.)

By default, `tlmgr` now saves one backup of each package updated (`tlmgr option autobackup 1`), so broken package updates can be easily reverted with `tlmgr restore`. If you do post-install updates, and don't have the disk space for the backups, run `tlmgr option autobackup 0`.

New programs included: the `pTeX` engine and related utilities for typesetting Japanese; the `BIBTeXU` program for Unicode-enabled `BIBTeX`; the `chktex` utility (originally from <http://baruch.ev-en.org/proj/chktex>) for checking (L<sup>A</sup>)TeX documents; the `dvisvgm` (<https://dvisvgm.de>) DVI-to-SVG translator.

Executables for these new platforms are included: `amd64-freebsd`, `amd64-kfreebsd`, `i386-freebsd`, `i386-kfreebsd`, `x86_64-darwin`, `x86_64-solaris`.

A change in TeX Live 2009 that we failed to note: numerous TeX4ht-related executables (<https://tug.org/tex4ht>) were removed from the binary directories. The generic `mk4ht` program can be used to run any of the various `tex4ht` combinations.

Finally, the TeX Live release on the TeX Collection DVD can no longer be run live (oddly enough). A single DVD no longer has enough room. One beneficial side effect is that installation from the physical DVD is much faster.

### 9.1.8 2011

The macOS binaries (`universal-darwin` and `x86_64-darwin`) now work only on Leopard or later; Panther and Tiger are no longer supported.

The `biber` program for bibliography processing is included on common platforms. Its development is closely coupled with the `biblatex` package, which completely reimplements the bibliographical facilities provided by LaTeX.

The MetaPost (`mpost`) program no longer creates or uses `.mem` files. The needed files, such as `plain.mp`, are simply read on every run. This is related to supporting MetaPost as a library, which is another significant though not user-visible change.

The `updmap` implementation in Perl, previously used only on Windows, has been revamped and is now used on all platforms. There shouldn't be any user-visible changes as a result, except that it runs much faster.

The `initex` and `inimf` programs were restored (but no other `ini*` variants).

### 9.1.9 2012

`tlmgr` supports updates from multiple network repositories. The section on multiple repositories in the `tlmgr help` output has more.

The parameter `\XeTeXdashbreakstate` is set to 1 by default, for both `xetex` and `xelatex`. This allows line breaks after em-dashes and en-dashes, which has always been the behavior of plain TeX, L<sup>A</sup>TeX, LuaTeX, etc. Existing XeTeX documents which must retain perfect line-break compatibility will need to set `\XeTeXdashbreakstate` to 0 explicitly.

The output files generated by `pdftex` and `dvips`, among others, can now exceed 2 gigabytes.

The 35 standard PostScript fonts are included in the output of `dvips` by default, since so many different versions of them are extant.

In the restricted `\write18` execution mode, set by default, `mpost` is now an allowed program.

A `texmf.cnf` file is also found in `../texmf-local`, e.g., `/usr/local/texlive/texmf-local/web2c/texmf.cnf`, if it exists.

The `updmap` script reads a per-tree `updmap.cfg` instead of one global config. This change should be invisible, unless you edited your `updmap.cfg`'s directly. The `updmap --help` output has more.

Platforms: `armel-linux` and `mipsel-linux` added; `sparc-linux` and `i386-netbsd` are no longer in the main distribution.

### 9.1.10 2013

Distribution layout: the top-level `texmf/` directory has been merged into `texmf-dist/`, for simplicity. Both the `TEXMFMAIN` and `TEXMFDIST` Kpathsea variables now point to `texmf-dist`.

Many small language collections have been merged together, to simplify installation.

MetaPost: native support for PNG output and floating-point (IEEE double) has been added.

LuaTeX: updated to Lua 5.2, and includes a new library (`pdfscanner`) to process external PDF page content, among much else (see its web pages).

XeTeX (also see its web pages for more):

- The HarfBuzz library is now used for font layout instead of ICU. (ICU is still used to support input encodings, bidirectionality, and the optional Unicode line breaking.)
- Graphite2 and HarfBuzz are used instead of SilGraphite for Graphite layout.
- On Macs, Core Text is used instead of the (deprecated) ATSUI.
- Prefer TrueType/OpenType fonts to Type1 when the names are the same.
- Fix occasional mismatch in font finding between XeTeX and `xdvipdfmx`.
- Support OpenType math cut-ins.

`xdvi`: now uses FreeType instead of `t1lib` for rendering.

`microtype.sty`: some support for XeTeX (protrusion) and LuaTeX (protrusion, font expansion, tracking), among other enhancements.

`tlmgr`: new pinning action to ease configuring multiple repositories; that section in `tlmgr --help` has more, online at <https://tug.org/texlive/doc/tlmgr.html#MULTIPLE-REPOSITORIES>.

Platforms: `armhf-linux`, `mips-irix`, `i386-netbsd`, and `amd64-netbsd` added or revived; `powerpc-aix` removed.

### 9.1.11 2014

2014 saw another TeX tune-up from Knuth; this affected all engines, but the only visible change likely is the restoration of the `preloaded format` string on the banner line. Per Knuth, this now reflects the format that *would* be loaded by default, rather than an undumped format that is actually preloaded in the binary; it may be overridden in various ways.

pdfTeX: new warning-suppression parameter `\pdfsuppresswarningpagegroup`; new primitives for fake interword spaces to help with PDF text reflowing: `\pdfinterwordspaceon`, `\pdfinterwordspaceoff`, `\pdffakespace`.

LuaTeX: Notable changes and fixes were made to font loading and hyphenation. The biggest addition is a new engine variant, `luajitex` and its siblings `texluajit` and

`texluajitc`. This uses a just-in-time Lua compiler (detailed *TUGboat* article at <https://tug.org/TUGboat/tb34-1/tb106scarso.pdf>). `luajittex` is still in development, is not available on all platforms, and is considerably less stable than `luatex`. Neither we nor its developers recommend using it except for the specific purpose of experimenting with jit on Lua code.

**XeTeX**: The same image formats are now supported on all platforms (including Mac); avoid Unicode compatibility decomposition fallback (but not other variants); prefer OpenType to Graphite fonts, for compatibility with previous XeTeX versions.

**MetaPost**: A new numbersystem `decimal` is supported, along with a companion internal `numberprecision`; a new definition of `drawdot` in `plain.mp`, per Knuth; bug fixes in SVG and PNG output, among others.

The `pstopdf` ConTeXt utility will be removed as a standalone command at some point after the release, due to conflicts with OS utilities of the same name. It can still (and now) be invoked as `mtxrun --script pstopdf`.

`psutils` has been substantially revised by a new maintainer. As a result, several seldom-used utilities (`fix*`, `getafm`, `psmerge`, `showchar`) are now only in the `scripts/` directory rather than being user-level executables (this can be reversed if it turns out to be problematic). A new script, `psjoin`, has been added.

The MacTeX redistribution of TeX Live (section 3.1.2) no longer includes the optional Mac-only packages for the Latin Modern and TeX Gyre fonts, since it is easy enough for individual users to make them available to the system. The `convert` program from ImageMagick has also been excised, since TeX4ht (specifically `tex4ht.env`) now uses Ghostscript directly.

The `langcjk` collection for Chinese, Japanese, and Korean support has been split into individual language collections for the sake of more moderate sizes.

**Platforms**: `x86_64-cygwin` added, `mips-irix` removed; Microsoft no longer supports Windows XP, so our programs may start failing there at any time.

### 9.1.12 2015

L<sup>A</sup>T<sub>ε</sub>X 2<sub>ε</sub> now incorporates, by default, changes previously included only by explicitly loading the `fixltx2e` package, which is now a no-op. A new `latexrelease` package and other mechanisms allow for controlling what is done. The included L<sup>A</sup>T<sub>ε</sub>X News #22 and “L<sup>A</sup>T<sub>ε</sub>X changes” documents have details. Incidentally, the `babel` and `psnfss` packages, while core parts of L<sup>A</sup>T<sub>ε</sub>X, are maintained separately and are not affected by these changes (and should still work).

Internally, L<sup>A</sup>T<sub>ε</sub>X 2<sub>ε</sub> now includes Unicode-related engine configuration (what characters are letters, naming of primitives, etc.) which was previously part of TeX Live. This change is intended to be invisible to users; a few low-level internal control sequences have been renamed or removed, but the behavior should be just the same.

**pdfTeX**: Support JPEG Exif as well as JFIF; do not emit a warning if `\pdfinclusionerrorlevel` is negative; sync with `xpdf` 3.04.

**LuaTeX**: New library `newtokenlib` for scanning tokens; bug fixes in the `normal` random number generator and other places.

**XeTeX**: Image handling fixes; `xdvipdfmx` binary looked for first as a sibling to `xetex`; internal XDV opcodes changed.

MetaPost: New numbersystem binary; new Japanese-enabled `upmpost` and `updvtomp` programs, analogous to `up*tex`.

MacTeX: Updates to the included Ghostscript package for CJK support. The TeX Distribution Preference Pane now works in Yosemite (macOS 10.10). Resource-fork font suitcases (generally without an extension) are no longer supported by XeTeX; data-fork suitcases (`.dfont`) remain supported.

Infrastructure: The `fmtutil` script has been reimplemented to read `fmtutil.cnf` on a per-tree basis, analogous to `updmap`. Web2C `mktex*` scripts (including `mktexlsr`, `mktexfm`, `mktexpk`) now prefer programs in their own directory, instead of always using the existing `PATH`.

Platforms: `*-kfreebsd` removed, since TeX Live is now easily available through the system platform mechanisms. Support for some additional platforms is available as custom binaries (<https://tug.org/texlive/custom-bin.html>). In addition, some platforms are now omitted from the DVD (simply to save space), but can be installed normally over the net.

### 9.1.13 2016

LuaTeX: Sweeping changes to primitives, both renames and removals, along with some node structure rearrangements. The changes are summarized in an article by Hans Hagen, “LuaTeX 0.90 backend changes for PDF and more” (<https://tug.org/TUGboat/tb37-1/tb115hagen-pdf.pdf>); for all the details, see the LuaTeX manual, `texmf-dist/doc/luatex/base/luatex.pdf`.

Metafont: New highly experimental sibling programs MFlua and MFluajit, integrating Lua with METAFONT, for trial testing purposes.

MetaPost: Bug fixes and internal preparations for MetaPost 2.0.

`SOURCE_DATE_EPOCH` support in all engines except LuaTeX (which will come in the next release) and original `tex` (intentionally omitted): if the environment variable `SOURCE_DATE_EPOCH` is set, its value is used for timestamps in the PDF output. If `SOURCE_DATE_EPOCH_TEX_PRIMITIVES` is also set, the `SOURCE_DATE_EPOCH` value is used to initialize the TeX primitives `\year`, `\month`, `\day`, `\time`. The pdfTeX manual has examples and details.

pdfTeX: Three new primitives `\pdfinfoomitdate`, `\pdfsuppressptexinfo`, and `\pdftrailerid`, to control values appearing in the output which normally change with each run. These features are for PDF output only, not DVI.

XeTeX: New primitives `\XeTeXhyphenatablelength`, `\XeTeXgenerateactualtext`, `\XeTeXinterwordspaceshaping`, `\mdfivesum`; character class limit increased to 4096; DVI id byte incremented.

Other utilities:

- `gregorio` is a new program, part of the `gregoriotex` package for typesetting Gregorian chant scores; it is included in `shell_escape_commands` by default.
- `upmendex` is an index creation program, mostly compatible with `makeindex`, with support for Unicode sorting, among other changes.
- `afm2tfm` now makes only accent-based height adjustments upward; a new option `-a` omits all adjustments.
- `ps2pk` can handle extended PK/GF fonts.



MacTeX: The TeX Distribution Preference Pane is gone; its functionality is now in TeX Live Utility; bundled GUI applications upgraded; new script `cjk-gs-integrate` to be run by users who wish to incorporate various CJK fonts into Ghostscript.

Infrastructure: System-level `tlmgr` configuration file supported; verify package checksums; if GPG is available, verify signature of network updates. These checks happen with both the installer and `tlmgr`. If GPG is not available, updates proceed as usual.

Platforms: `alpha-linux` and `mipsel-linux` removed.

### 9.1.14 2017

LuaTeX: More callbacks, more typesetting control, more access to internals; `ffi` library for dynamic code loading added on some platforms.

pdfTeX: Environment variable `SOURCE_DATE_EPOCH_TEX_PRIMITIVES` from last year renamed to `FORCE_SOURCE_DATE`, with no changes in functionality; if the `\pdfpageattr` token list contains the string `/MediaBox`, omit output of the default `/MediaBox`.

XeTeX: Unicode/OpenType math now based on HarfBuzz's MATH table support; some bug fixes.

Dvips: Make the last papersize special win, for consistency with `dvipdfmx` and package expectations; the `-L0` option (L0 config setting) restores the previous behavior of the first special winning.

epTeX, eupTeX: New primitives `\pdfuniformdeviate`, `\pdfnormaldeviate`, `\pdfrandomseed`, `\pdfsetrandomseed`, `\pdfelapsedtime`, `\pdfresettimer`, from pdfTeX.

MacTeX: As of this year, only macOS releases for which Apple still releases security patches will be supported in MacTeX, under the platform name `x86_64-darwin`; currently this means Yosemite, El Capitan, and Sierra (10.10 and newer). Binaries for older macOS versions are not included in MacTeX, but are still available in TeX Live (`x86_64-darwinlegacy`, `i386-darwin`, `powerpc-darwin`).

Infrastructure: The `TEXMFLOCAL` tree is now searched before `TEXMFSYSCONFIG` and `TEXMFSYSVAR` (by default); the hope is that this will better match expectations of local files overriding system files. Also, `tlmgr` has a new mode `shell` for interactive and scripted use, and a new action `conf auxtrees` to easily add and remove extra trees.

`updmap` and `fmtutil`: These scripts now give a warning when invoked without explicitly specifying either so-called system mode (`updmap-sys`, `fmtutil-sys`, or option `-sys`), or user mode (`updmap-user`, `fmtutil-user`, or option `-user`). The hope is that this will reduce the perennial problem of invoking user mode by accident and thus losing future system updates. See <https://tug.org/texlive/scripts-sys-user.html> for details.

`install-tl`: Personal paths such as `TEXMFHOME` are now set to MacTeX values (`~/Library/...`) by default on Macs. New option `-init-from-profile` to start an installation with the values from a given profile; new command `P` to explicitly save a profile; new profile variable names (but previous ones are still accepted).

SyncTeX: the name of the temporary file now looks like `foo.synctex(busy)`, instead of `foo.synctex.gz(busy)` (no `.gz`). Front-ends and build systems that want to remove temp files may need adjusting.

Other utilities: `texosquery-jre8` is a new cross-platform program for retrieving locale and other OS information from a TeX document; it is included in

`shell_escape_commands` by default for restricted shell execution. (Older JRE versions are supported by `texosquery`, but cannot be enabled in restricted mode, as they are no longer supported by Oracle, even for security issues.)

Platforms: See `MacTeX` entry above; no other changes.

### 9.1.15 2018

`Kpathsea`: Case-insensitive filename matching now done by default in non-system directories; set `texmf.cnf` or environment variable `texmf_casefold_search` to 0 to disable. Full details in the `Kpathsea` manual (<https://tug.org/kpathsea>).

`epTeX`, `eupTeX`: New primitive `\epTeXversion`.

`LuaTeX`: Preparation for moving to Lua 5.3 in 2019: a binary `luatex53` is available on most platforms, but must be renamed to `luatex` to be effective. Or use the `ConTeXt Garden` (<https://wiki.contextgarden.net>) files; more information there.

`MetaPost`: Fixes for wrong path directions, TFM and PNG output.

`pdfTeX`: Allow encoding vectors for bitmap fonts; current directory not hashed into PDF ID; bug fixes for `\pdfprimitive` and related.

`XeTeX`: Support `/Rotate` in PDF image inclusion; exit nonzero if the output driver fails; various obscure UTF-8 and other primitive fixes.

`MacTeX`: See version support changes below. In addition, the files installed in `/Applications/TeX/` by `MacTeX` have been reorganized for greater clarity; now this location contains four GUI programs (`BibDesk`, `LaTeXiT`, `TeX Live Utility`, and `TeXShop`) at the top level and folders with additional utilities and documentation.

`tlmgr`: new front-ends `tlshell` (Tcl/Tk) and `tlcockpit` (Java); JSON output; `uninstall` now a synonym for `remove`; new action/option `print-platform-info`.

Platforms:

- Removed: `armel-linux`, `powerpc-linux`.
- `x86_64-darwin` supports 10.10–10.13 (Yosemite, El Capitan, Sierra, and High Sierra).
- `x86_64-darwinlegacy` supports 10.6–10.10 (though `x86_64-darwin` is preferred for 10.10). All support for 10.5 (Leopard) is gone, that is, both the `powerpc-darwin` and `i386-darwin` platforms have been removed.
- Windows: XP is no longer supported.

### 9.1.16 2019

`Kpathsea`: More consistent brace expansion and path splitting; new variable `TEXMFDOTDIR` instead of hard-coded `.` in paths allows for easily searching additional or sub-directories (see comments in `texmf.cnf`).

`epTeX`, `eupTeX`: New primitives `\readpapersizespecial` and `\expanded`.

`LuaTeX`: Lua 5.3 now used, with concomitant arithmetic and interface changes. The homegrown library `pplib` is used to read pdf files, thus eliminating the dependency on `poppler` (and the need for C++); Lua interface changed accordingly.

`MetaPost`: `r-mpost` command name recognized as an alias for invocation with the `--restricted` option, and added to the list of restricted commands available by default. Minimum precision now 2 for decimal and binary mode. Binary mode no longer available in `MPLib` but still available in standalone `MetaPost`.



pdf $\TeX$ : New primitive `\expanded`; if new primitive parameter `\pdfomitcharset` is set to 1, the `/CharSet` string omitted from the PDF output, since it cannot feasibly be guaranteed correct, as required by PDF/A-2 and PDF/A-3.

Xe $\TeX$ : New primitives for cross-engine support: `\expanded`, `\creationdate`, `\elapsedtime`, `\filedump`, `\filemdate`, `\filesize`, `\resettimer`, `\normaldeviate`, `\uniformdeviate`, `\randomseed`; extend `\Ucharcat` to produce active characters.

tlmgr: Support `curl` as a download program; use `lz4` and `gzip` before `xz` for local backups, if available; prefer system-provided binaries over binaries provided with  $\TeX$  Live for compressor and download programs, unless the environment variable `TEXLIVE_PREFER_OWN` is set.

`install-tl`: New option `-gui` (with no argument) is the default on Windows and Macs, and invokes a new Tcl/TK GUI (see sections 1.3 and 3.1.6).

Utilities:

- `cwebbin` (<https://ctan.org/pkg/cwebbin>) is now the CWEB implementation in  $\TeX$  Live, with support for more language dialects, and including the `ctwill` program to make mini-indexes.
- `chkdvifont`: report font information from DVI files, also from `tfm/ofm`, `vf`, `gf`, `pk`.
- `dvispc`: make a DVI file page-independent with respect to specials.

Mac $\TeX$ : `x86_64-darwin` now supports 10.12 and higher (Sierra, High Sierra, Mojave); `x86_64-darwinlegacy` still supports 10.6 and newer. The spell checker `Excalibur` is no longer included, since it requires 32-bit support.

Platforms: removed `sparc-solaris`.

### 9.1.17 2020

General:

- The `\input` primitive in all  $\TeX$  engines, including `tex`, now also accepts a group-delimited filename argument, as a system-dependent extension. The usage with a standard space/token-delimited filename is completely unchanged. The group-delimited argument was previously implemented in Lua $\TeX$ ; now it is available in all engines. ASCII double quote characters (") are removed from the filename, but it is otherwise left unchanged after tokenization. This does not currently affect  $\LaTeX$ 's `\input` command, as that is a macro redefinition of the standard `\input` primitive.
- New option `--cnf-line` for `kpsewhich`, `tex`, `mf`, and all other engines, to support arbitrary configuration settings on the command line.
- The addition of various primitives to various engines in this and previous years is intended to result in a common set of functionality available across all engines (*LaTeX News #31*, <https://latex-project.org/news>).

ep $\TeX$ , eup $\TeX$ : Several new primitives, mostly for cross-engine support: `\Uchar`, `\Ucharcat`, `\current(x)spacingmode`, `\ifincsname`; also, revise `\fontchar??` and `\iffontchar`. For eup $\TeX$  only: `\currentcjktoken`.

Lua $\TeX$ : Integration with `HarfBuzz` library, available as new engines `luahtex` (used for `lualatex`) and `luajithbtex`. New primitives: `\eTeXgluestretchorder`, `\eTeXglueshrinkorder`.

pdf $\TeX$ : New primitive `\pdfmajorversion`; this merely changes the version number in the PDF output; it has no effect on any PDF content. `\pdfximage` and similar now search for image files in the same way as `\openin`.

p $\TeX$ : New primitives `\ifjfont`, `\iftfont`. Also in ep $\TeX$ , up $\TeX$ , eup $\TeX$ .

Xe $\TeX$ : Fixes for `\Umathchardef`, `\XeTeXinterchartoks`, `\pdfsavepos`.

Dvips: Output encodings for bitmap fonts, for better copy/paste capabilities (<https://tug.org/TUGboat/tb40-2/tb125rokicki-type3search.pdf>).

Mac $\TeX$ : Mac $\TeX$  and `x86_64-darwin` now require 10.13 or higher (High Sierra, Mojave, and Catalina); `x86_64-darwinlegacy` supports 10.6 and newer. Mac $\TeX$  is notarized and command line programs have hardened runtimes, as now required by Apple for install packages. BibDesk and  $\TeX$  Live Utility are not in Mac $\TeX$  because they are not notarized, but a `README` file lists urls where they can be obtained.

`x86_64-darwinlegacy`: drop X11 support, including `xdvi`.

`tlmgr` and infrastructure:

- Automatically retry (once) packages that fail to download.
- New option `tlmgr check texmfdb`s, to check consistency of `ls-R` files and `!!` specifications for each tree.
- Use versioned filenames for the package containers, as in: `tlnet/archive/pkgname.rNMM.tar.xz`. This should be invisible to users, but is a notable change in distribution.
- `catalogue-date` information no longer propagated from the  $\TeX$  Catalogue, since it was often unrelated to package updates.

### 9.1.18 2021

General:

- Donald Knuth's changes for his 2021 tuneup of  $\TeX$  and Metafont are incorporated (<https://tug.org/TUGboat/tb42-1/tb130knuth-tuneup21.pdf>). They are also available on CTAN as the `knuth-dist` and `knuth-local` packages. As expected, the fixes are for obscure cases and do not affect any behavior in practice.
- Except in original  $\TeX$ : if `\tracinglostchars` is set to 3 or more, missing characters will result in an error, not just a message in the log file, and the missing character code will be shown in hex.
- Except in original  $\TeX$ : a new integer parameter `\tracingstacklevels`, if positive, and `\tracingmacros` is also positive, causes a prefix indicating the macro expansion depth to be output on each relevant log line (e.g., `~..` at depth 2). Also, macro logging is truncated at a depth  $\geq$  the parameter value.

Aleph: The Aleph-based  $\LaTeX$  format, named `lamed`, has been removed. The `aleph` binary itself is still included and supported.

Lua $\TeX$ :

- Lua 5.3.6.
- Callback for nesting level used in `\tracingmacros`, as generalized variant of the new `\tracingstacklevels`.
- Mark math glyphs as protected to prevent processing as text.
- Removed width/ic compensation for traditional math code path.

MetaPost:

- `SOURCE_DATE_EPOCH` environment variable support for reproducible output.
- Avoid wrong final % in `mpto`.
- Document `-T` option, other fixes to manual.
- Value of `epsilon` changed in binary and decimal modes, so that the function `mp_solve_rising_cubic` works as expected.

pdf $\TeX$ :

- New primitives `\pdfrunninglinkoff` and `\pdfrunninglinkon`; e.g., for disabling generation of links in headers and footers.
- Warn instead of aborting when “`\pdfendlink` ended up in different nesting level than `\pdfstartlink`”.
- Dump `\pdfglyphtounicode` assignments in `fmt` file.
- Source: `poppler` support removed, as it was too hard to keep in sync with upstream. In native TL, pdf $\TeX$  has always used `libs/xpdf`, which is cut-down and adapted code from `xpdf`.

Xe $\TeX$ : Fixes for math kerning.

Dvipdfmx:

- Ghostscript is now invoked safely by default; to override (thereby assuming all input files are trusted), use `-i dvipdfmx-unsafe.cfg`. To use PSTricks with Xe $\TeX$ , this is required, as in:  

```
xetex -output-driver="xdvipdfmx -i dvipdfmx-unsafe.cfg -q -E" ...
```
- If an image file is not found, exit with bad status.
- Extended special syntax for color support.
- Specials for manipulating `ExtGState`.
- Compatibility specials `pdfcolorstack` and `pdffontattr`.
- Experimental support for `dviluatex`'s extended `fnt_def`.
- Support new feature of virtual font to fallback Japanese font definition.

Dvips:

- Default PostScript document title is now the basename of the input file, and can be overridden with the new option `-title`.
- If an `.eps` or other image file is not found, exit with bad status.
- Support new feature of virtual font to fallback Japanese font definition.

Mac $\TeX$ : Mac $\TeX$  and its new binary folder `universal-darwin` now require macOS 10.14 or higher (Mojave, Catalina, and Big Sur); the `x86_64-darwin` binary folder is no longer present. The `x86_64-darwinlegacy` binary folder, available only with the Unix `install-tl`, supports 10.6 and newer.

This is an important year for the Macintosh because Apple introduced ARM machines in November and will sell and support both ARM and Intel machines for many years. All programs in `universal-darwin` have executable code for both ARM and Intel. Both binaries are compiled from the same source code.

The additional programs Ghostscript, LaTeXiT,  $\TeX$  Live Utility, and TeXShop are all universal and are signed with a hardened runtime, so all are included in Mac $\TeX$  this year.

`tlmgr` and infrastructure:

- keep only one backup of the main repository's `texlive.tlpdb`.
- even more portability across systems and Perl versions.

- `tlmgr info` reports new `lcat-*` and `rcat-*` fields for local vs. remote Catalogue data.
- full logging of subcommands moved to a new log file:  
`texmf-var/web2c/tlmgr-commands.log`.

### 9.1.19 2022

General:

- New engine `hitex`, which outputs its own HINT format, designed especially for reading technical documents on mobile devices. HINT viewers for GNU/Linux, Windows, and Android are available separately from `TEX Live`.
- `tangle`, `weave`: support optional third argument to specify output file.
- Knuth's program `twill` for making mini-indexes for original `WEB` programs now included.

Cross-engine extensions (except in original `TEX`, `Aleph`, and `hiTEX`):

- New primitive `\showstream` to redirect `\show` output to a file.
- New primitives `\partokenname` and `\partokencontext` allow overriding the name of the `\par` token emitted at blank lines, the end of `vboxes`, etc.

`epTEX`, `eupTEX`:

- New primitives: `\lastnodefont`, `\suppresslongerror`, `\suppressoutererror`, `\suppressmathparerror`.
- `pdfTEX` extension `\vadjust pre` now available.

`LuaTEX`:

- Support structured destinations from PDF 2.0.
- PNG /Smask for PDF 2.0.
- Variable font interface for `luaHbtex`.
- Different radical style defaults in `mathdefaultsmode`.
- Optionally block selected discretionary creation.
- Improvements for TrueType fonts implementation.
- More efficient `\fontdimen` allocation.
- Ignore paragraphs with only a local `par` node followed by direction synchronization nodes.

MetaPost: Bug fix for infinite macro expansion.

`pdfTEX`:

- Support structured destinations from PDF 2.0.
- For letterspaced fonts, use explicit `\fontdimen6` if specified.
- Always start a warning at the beginning of a line.
- For characters with autokern (`\pdfappendkern` and `\pdfprependkern`), still do protrusion; likewise, autokern both implicit and explicit hyphens.

`pTEX` et al.:

- Major update of `pTEX` to 4.0.0 to better support current `LATEX`.
- New primitives `\ptexlineendmode` and `\toucs`.
- `\ucs` (formerly available in `uptex`, `euptex`) now available also in `pTEX` and `epTEX`.
- Distinguish eight-bit characters and Japanese characters, as discussed in a TUGboat article by Hironori Kitagawa (<https://tug.org/TUGboat/tb41-3/tb129kitagawa-char.pdf>).

XeTeX: New wrapper scripts `xetex-unsafe` and `xelatex-unsafe` for simpler invocation of documents requiring both XeTeX and PSTricks transparency operators, which is inherently unsafe (until and unless reimplementations in Ghostscript happens). For safety, use Lua(L)TeX.

Dvipdfmx:

- Support for PSTricks without requiring `-dNOSAFER`, except for transparency.
- The `-r` option to set bitmap font resolution works again.

Dvips: By default, do not attempt automatic media adjustment for rotated paper sizes; the new option `--landscaperotate` re-enables.

upmendex: Experimental support for Arabic and Hebrew scripts; improved character classification and language support.

Kpathsea: First path returned from `kpsewhich -all` is now the same as a regular (non-all) search.

tlmgr and infrastructure:

- use `https` for `mirror.ctan.org` by default.
- use `TEXMFROOT` instead of `SELFAUTOPARENT` for easier relocating.
- `install-tl`: if download or installation fails for a given package, automatically continue and later retry (once).

MacTeX: MacTeX and its binary folder `universal-darwin` require macOS 10.14 or higher (Mojave, Catalina, Big Sur, Monterey). The `x86_64-darwinlegacy` binary folder, available only with the Unix `install-tl`, supports 10.6 (Snow Leopard) and newer.

Platforms: No changes to platform support for this year (2022). However, for next year's release (2023), we are planning to switch the Windows binaries from 32-bit to 64-bit. Unfortunately we cannot feasibly support both simultaneously.

### 9.1.20 2023

Windows: As announced previously, TeX Live now contains 64-bit Windows binaries instead of 32-bit. The new directory name is `bin/windows` (it did not seem right to put 64-bit binaries into a directory named with "32"). We know this will cause extra work for Windows users, but there seemed no better alternative. If you need 32-bit Windows support, you'll need to install TeX Live 2022. From Windows 11 onward, Windows on ARM supports emulation of 64-bit AMD/Intel and thus can use the current TeX Live. However, Windows 10 on ARM cannot emulate Intel/AMD 64-bit, and will therefore also need a 32-bit TeX Live. See the separate TeX Live Windows web page (<https://tug.org/texlive/windows.html>).

Cross-engine extensions (except in original TeX and e-TeX): `\special` followed by a new keyword `"shipout"` delays expansion of the argument tokens until `\shipout` time, as with a non-`\immediate\write`.

epTeX, eupTeX:

- "Raw" (u)ptex no longer built; (u)ptex now runs in e(u)ptex's compatibility mode. Same for pTeX tools, listed below.
- New primitives: `\tojis`, `\ptextracingfonts`, `\ptexfontname`.
- For `\font`, new syntax for JIS/UCS is supported.

LuaTeX:

- new primitive `\variablefam` to allow math characters to keep their class while still letting the family adapt.
- improved `r2l` annotation areas
- cross-engine “late `\special`” described above.
- in LuaTeX 1.17.0, released in May 2023: the socket library is now disabled by default; a new command line option `--socket` enables it, as well as `--shell-escape` (not `--shell-restricted`). In addition, the mime library is now always available, and new functions `os.socketsleep` and `os.socketgettime` are also always available. Finally, some obscure ways to work around various security features were fixed.

MetaPost: Bug fixes. `svg->dx` and `svg->dy` are now `double`, for better precision; `mp_begin_iteration` updated; memory leak in `mplib` fixed.

pdfTeX:

- new primitive `\pdfomitinfodict` to omit `/Info` dictionary completely.
- new primitive `\pdfomitprocset` to control omitting `/ProcSet` array: `/ProcSet` is included if this parameter is negative, or if this parameter is zero and `pdftex` is generating PDF 1.x output.
- with `\pdfinterwordspaceon`, if the current font’s encoding has a `/space` character at slot 32, it is used; otherwise, the `/space` from the (new) default font `pdftexspace` is used. That default font can be overridden with the new primitive `\pdfspacefont`. This same new procedure is used for `\pdffakepace`.

pTeX et al.:

- As mentioned above, `ptex` now runs `eptex` in compatibility mode instead of being built separately.
- pTeX tools (`pbibtex`, `pdvitype`, `ppltotf`, `ptftopl`) merged into corresponding upTeX versions, running in compatibility mode.

XeTeX: fixed `\topskip` and `\splittopskip` computation when `\XeTeXupwardsmode` is active; the cross-engine “late `\special`” described above.

Dvipdfmx: new option `--pdfm-str-utf8` to make `pdfmark` and/or `bookmark`.

BibTeXu:

- This BibTeX variant is mostly upward-compatible with BibTeX, with much better (Unicode-based) multilingual support. It’s been in TeX Live for some years.
- This year, more features to support CJK languages have been added, some extended from the Japanese `(u)pbibtex` and other programs.

Kpathsea: Support guessing input file encodings for Unix-ish platforms, as on Windows; enabled for `(e)p(la)tex`, `pbibtex`, `mendex`.

`tlmgr` and infrastructure:

- default to text interface on macOS.
- install core packages first, retry other packages once.
- simplistic checks are done for enough disk space.

MacTeX:

- MacTeX and its binary folder `universal-darwin` require macOS 10.14 or higher (Mojave, Catalina, Big Sur, Monterey, Ventura). The `x86_64-darwinlegacy` binary folder, available only with the Unix `install-tl`, supports 10.6 (Snow Leopard) and later.
- The GUI package in MacTeX now contains `hintview`, a macOS viewer for HINT documents (created by the `hitex` and `hilatex` engines for mobile devices; see

the HiTeX web page, <https://hint.userweb.mwn.de/hint/hitex.html>). The GUI package no longer installs a folder of documents, replacing them with a short READ ME for new users and a page about `hintview`.

- The `Extras` folder of additional TeX software on the DVD has been replaced with a document containing links to download sites.

Platforms:

- As mentioned above, the new `windows` binary directory contains 64-bit Windows binaries, and
- the `bin/win32` binary directory is gone, since we cannot support 32-bit and 64-bit Windows simultaneously.
- The `i386-cygwin` binary directory is gone, since Cygwin no longer supports i386.

## 9.2 Present: 2024

Cross-engine changes (except in original TeX, e-TeX, and LuaTeX):

- Avoid crashes on devious use of `\output` and `\write` braces.
- Tiny inner-loop optimization. (Both these changes are due to David Fuchs.)

LuaTeX:

- Lua-level checks for writing to files now similar to the TeX-level checks.
- If the Lua value `texconfig.trace_extra_newline` is set, blank lines are output in the log between macro parameters. In 2020–2023, this was the case by default; before that, such blank lines were never output, which is the behavior of the other engines.
- In the past, Windows `texlua` (not `luatex`, and not any other platform), internally performed wildcard expansion on the arguments. This no longer happens.

MetaPost:

- MPFI library now included for interval arithmetic; enabled with `--numbersystem=interval`.

pdfTeX:

- More reliable text extraction on BigEndian systems.
- More checks against corrupt pfb, fnt, pdf files.
- Avoid writing a corrupt pdf given unusual pdf inclusions.
- Manual converted to L<sup>A</sup>TeX and thoroughly reviewed, though substantive updates are minimal.

epTeX, eupTeX:

- New simple encoding conversion utility `ptekf`.
- `eptex` no longer built separately; it runs in `euptex`'s legacy-encoding mode.

XeTeX:

- Unicode characters in filenames beyond the BMP are supported.

Dvipdfmx:

- Calculate time zone offset in the same way as pdfTeX.
- Handle variation selectors to cope with the new Adobe-Japan1-UCS2 CMap.
- Fallback multibyte characters in virtual fonts to OFM fonts for (u)pTeX.
- Support the `named` keyword in the `pdf:image` special.

Kpathsea:



- Support an extended check for safe filenames which also allows writing under `TEXMF[SYS]VAR`, for Lua(L<sup>A</sup>)T<sub>E</sub>X; new functions and corresponding `kpsewhich` options.
- Support a new variable `TEXMF_OUTPUT_DIRECTORY` (alongside the traditional `TEXMFOUTPUT`), so that subprograms can have access to an `--output-directory` setting in an engine invocation.

CWEB:

- Support LuaT<sub>E</sub>X as backend T<sub>E</sub>X engine.
- Fix buffer overflow on lines with no breakpoint.

tlmgr:

- New action `bug` to help find bug-reporting information.
- Use our own root certificate bundle on all platforms, to support LWP more widely, notably on macOS.
- Detect a remote repository older than the local installation (i.e., a stale mirror), and don't update, to avoid incorrect package removals.
- Support ConT<sub>E</sub>Xt paper size setting.
- Prune backups of `tlpdb` files for mirrored repositories (main and `tlcontrib`).

Platforms:

- The ISO image again includes binaries for all platforms. For the last several years, some binary sets had been pruned, but this year, we are not creating a physical DVD as a user group benefit, so may as well include everything again. Also, even a maximally-pruned image is too large to fit on a single-layer DVD, so pruning has no benefit for the 2024 volunteer burners. More info at <https://tug.org/texcollection/>.
- Robert Alessi provides binaries for a native TL installation on OpenBSD at <https://sr.ht/~ralessi/texlive-openbsd/>.

MacT<sub>E</sub>X: now contains the latest Ghostscript 10.03.0, updates of all four GUI programs, along with (as always) all new TL changes. The binaries still support macOS 10.14 (Mojave, 2018) and later. (As usual, earlier Mac versions are supported by the `x86_64-darwinlegacy` binaries, which are part of TL, but not MacT<sub>E</sub>X.)

### 9.3 Future

We intend to continue to release new versions of T<sub>E</sub>X Live, and would like to provide more documentation, more programs, an ever-improved and better-checked tree of macros and fonts, and anything else T<sub>E</sub>X. This work is all done by volunteers in their spare time, and there is always more to do. Please see <https://tug.org/texlive/contribute.html>.

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

[tex-live@tug.org](mailto:tex-live@tug.org) (you need not subscribe in order to post)  
<https://tug.org/texlive>

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