The TEX Live Guide
TEX Collection 2006

Karl Berry, editor

http://tug.org/texlive/

December 2006

Contents

1 Introduction **2**
1.1 Basic usage of TEX Live . . . 2
1.2 Getting help . . . 3

2 Structure of TEX Live **4**
2.1 Multiple distributions: live, inst, protext . . . 4
2.2 Top level directories . . . 4
2.3 Overview of the predefined texmf trees . . . 5
2.4 Extensions to TEX . . . 5
2.5 Other notable programs in TEX Live . . . 6

3 Unix installation **6**
3.1 Running TEX Live directly from media (Unix) . . . 7
3.2 Installing TEX Live to disk . . . 8
3.3 Installing individual packages to disk . . . 11

4 Post-installation **13**
4.1 The texconfig program . . . 13
4.2 Testing the installation . . . 14

5 Mac OS X installation **15**

6 Windows installation **15**
6.1 Installing TEX Live to disk . . . 16
6.2 Support packages for Windows . . . 17

7 Maintenance of the installation in Windows **17**
7.1 Adding/removing packages . . . 18
7.2 Configuring and other management tasks . . . 18
7.3 Uninstalling TEX Live . . . 19
7.4 Adding your own packages to the installation . . . 19
7.5 Running tlmp.exe from the command line . . . 19
7.6 Network installation . . . 19
7.7 What's different in Windows? . . . 20
7.8 Personal configurations . . . 20

7.9 Testing . . . 21
7.10 Printing . . . 22
7.11 Tips and tricks for Win32 . . . 22
7.12 In case of problems . . . 25

8 A user's guide to Web2C 26
8.1 Kpathsea path searching . . . 27
8.2 Filename databases . . . 30
8.3 Runtime options . . . 36

9 Acknowledgements 36

10 Release history 37
10.1 Past . . . 37
10.2 Present . . . 41
10.3 Future . . . 41

List of Tables

1 Supported system architectures. . . . 8
2 Main menu options for the installation. . . . 9

1 Introduction

This document describes the main features of the TeX Live software distribution — TeX and friends for GNU/Linux and other Unix flavors, Mac OS X, and (32-bit) Windows systems. (Warning: it is not especially useful for older Mac or MS-DOS systems.)

TeX Live includes executables for TeX, LaTeX 2ε, ConTeXt, METAFONT, MetaPost, BibTeX and many other programs; an extensive collection of macros, fonts and documentation; and support for typesetting in many different scripts from around the world. It is part of the even larger TeX Collection (briefly described below in section 2, p. 4). Both are cooperative efforts by the TeX user groups.

For newer versions of the packages included here, please check CTAN: `http://www.ctan.org`.

For a brief summary of the major changes in this edition of TeX Live, see the end of the document, section 10 (p. 37).

1.1 Basic usage of TeX Live

You can use TeX Live in three principal ways:

1. You can run TeX Live directly from the live DVD (see section 2.1, p. 4). This takes almost no disk space, and gives you immediate access to everything in TeX Live. Of course performance will be worse than running on local disk, but you may well find it useful.

2. You can install all or part of TeX Live to a local disk, from either the DVD or the inst CD. This is the most common use of TeX Live. You will need (approximately) 100 megabytes for the most minimal system, and upwards of 1.3 gigabytes for a full system.

3. You can integrate a particular package or collection into your existing TeX system, either a TeX Live system you installed earlier, or a different system.

All of these are described in detail in the OS-specific installation sections following, but here is a quick start:

- The main installation script for Unix and Mac OS X is `install-tl.sh`. GNU/Linux users can also try a new GUI installation program: `cd setuptl` and run `tlpmgui`. Information can be found in section 6 on p.15.
- The single package installation script is `install-pkg.sh`. (After installation on Linux with `tlpmgui`, you can also try running `tlpmgui` again to add or remove individual packages or collections.)
- The installation program for Windows is `tlpmgui.exe`. It can be used also for adding or removing the packages. See section 6 below for more information.

1.2 Getting help

The TEX community is both active and friendly, and virtually all serious questions end up getting answered. However, the support is informal, done by volunteers and casual readers, so it's especially important that you do your homework before asking. (If you prefer guaranteed commercial support, you can forego TEX Live completely and purchase a vendor's system; http://tug.org/interest.html#vendors has a list.)

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

Getting Started If you are new to TEX, the web page http://tug.org/begin.html gives a brief introduction to the system.

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

TEX Catalogue If you are looking for a specific package, font, program, etc., the TEX Catalogue is the place to look. It is a huge collection of all TEX-related items. See texmf-doc/doc/english/catalogue, or http://www.ctan.org/tex-archive/help/Catalogue.

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

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

asking questions If you cannot find an answer, you can post to `comp.text.tex` through Google or your newsreader, or to texhax@tug.org through email. But before you post, *please* read this FAQ entry on asking questions in such a way that you're most likely to get an answer: http://www.tex.ac.uk/cgi-bin/texfaq2html?label=askquestion.

TEX Live support If you want to report a bug or have suggestions or comments on the TEX Live distribution, installation, or documentation, the mailing list is tex-live@tug.org. However, if your question is about how to use a particular program included in TEX Live, you are better off writing to that program's maintainer or mailing list.

The other side of the coin is helping others who have questions. Both `comp.text.tex` and `texhax` are open to anyone, so feel free to join, start reading, and help out where you can. Welcome to TEX!

2 Structure of TEX Live

This section describes the structure and contents of TEX Live and the TEX Collection of which it is a part.

2.1 Multiple distributions: live, inst, protext

Space limitations of CD-ROM format have forced us to divide TEX Collection into several distributions, as follows.

live A complete system on DVD; it is too large for CD. It can be run live or installed to disk. It also includes a snapshot of the CTAN repository, the protext distribution for Windows and the MacTEX distribution for Mac OS X, entirely independent of TEX Live, as well as assorted other packages in a `texmf-extra` directory.

CTAN, protext, MacTEX, and `texmf-extra` do not follow the same copying conditions as TEX Live, so be careful when redistributing or modifying.

inst(allable) A complete system on CD; in order to make this fit, the packages and programs are compressed. Therefore, it is not possible to run TEX directly from the installable CD, you have to install it to disk (hence its name). Installation is described in subsequent sections.

protext An enhancement of the MiKTEX distribution for Windows, ProTEXt adds a few extra tools to MiKTEX, and simplifies installation. It is entirely independent of TEX Live, and has its own installation instructions. It can be run live, or installed to disk. The ProTEXt home page is http://tug.org/protext.

ProTEXt is provided as both the top level of the live DVD and on its own CD (for those who cannot use the DVD).

You can tell which type of distribution you're in by looking for a `00`*type*`.TL` file at the top of the TEX Live directory. This file also contains the TEX Live release date.

Naturally, each user group chooses what to distribute, at its own discretion. (TUG is sending all three discs above to all of its members.)

2.2 Top level directories

Here is a brief listing and description of the top level directories in the TEX Live distribution. On the live DVD, the entire TEX Live hierarchy is in a subdirectory `texliveYYYY`, not at the top level of the disc.

`bin`	The TEX system programs, arranged by platform.
`source`	The source of all programs, including the main Web2C TEX and METAFONT distributions. These are stored in a bzip2-compressed tar archive.
`support`	Assorted auxiliary packages and programs. These are *not* installed automatically. This includes assorted editors and TEX shells.
`texmf`	Tree for the programs, along with their support files and documentation. Does not include TEX formats and packages. (`TEXMFMAIN` in the next section.)
`texmf-dist`	The main tree of formats and packages. (`TEXMFDIST` in the next section.)
`texmf-doc`	Tree for self-contained pure documentation, arranged by language.
`texmf-var`	Tree for files automatically generated and stored. (`TEXMFSYSVAR` in the next section.)
`xemtex`	Tree for supporting programs used only in Windows. These programs generally come pre-installed on Unix systems, or are at least easy to compile.

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

The `texmf-doc` directory contains only documentation, but it does not contain all the documentation. The documentation for the programs (manuals, man pages, Info files) is in `texmf/doc`, since the programs are in `texmf`. Similarly, the documentation for TEX packages and formats is in `texmf-dist/doc`. You can use the `texdoc` or `texdoctk` programs to find any documentation wherever it's located. The comprehensive links in the top-level file `doc.html` may also be helpful.

2.3 Overview of the predefined texmf trees

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

`TEXMFMAIN` The tree which holds vital parts of the system such as helper scripts (e.g., `web2c/mktexdir`), pool files and other support files.

`TEXMFDIST` The tree which holds the main set of macro packages, fonts, etc., as originally distributed.

`TEXMFLOCAL` The tree which administrators can use for system-wide installation of additional or updated macros, fonts, etc.

`TEXMFHOME` The tree which users can use for their own individual installations of additional or updated macros, fonts, etc. The expansion of this variable depends on `$HOME` by default, which dynamically adjusts for each user to an individual directory.

`TEXMFCONFIG` The tree used by teTEX's utilities `texconfig`, `updmap`, and `fmtutil` to store modified configuration data. Under `$HOME` by default.

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

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

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

For more discussion of `texconfig` and related utilities, please see section 4.1, p. 13.

2.4 Extensions to TEX

TEX Live contains several extended versions of TEX:

ε-TEX adds a small but powerful set of new primitives (related to macro expansion, character scanning, classes of marks, additional debugging features, and more) and the TEX--XET extensions for bidirectional typesetting. In default mode, ε-TEX is 100% compatible with ordinary TEX. See `texmf-dist/doc/etex/base/etex_man.pdf`.

pdfTEX builds on the ε-TEX extensions, adding support for writing PDF output as well as DVI. See `texmf/doc/pdftex/manual/` for the manual, and `texmf/doc/pdftex/manual/samplepdf/samplepdf.tex`. This is the default program for all formats except plain TEX.

XeTEX adds support for Unicode input and OpenType fonts, using third-party libraries. See `http://scripts.sil.org/xetex`.

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

Aleph combines the Ω and ε-TEX extensions. See `texmf-dist/doc/aleph/base` for some minimal documentation.

2.5 Other notable programs in TEX Live

Here are a few other commonly-used programs included in TEX Live:

bibtex bibliography support.

makeindex index support.

dvips convert DVI to PostScript.

xdvi DVI previewer for the X Window System.

dvilj DVI drive for the HP LaserJet family.

dv2dt, dt2dv convert DVI to/from plain text.

dviconcat, dviselect cut and paste pages from DVI files.

dvipdfmx convert DVI to PDF, an alternative approach to pdfTEX (mentioned above). See the ps4pdf and pdftricks packages for still more alternatives.

psselect, psnup, ... PostScript utilities.

lacheck LATEX syntax checker.

texexec ConTEXt and PDF processor.

tex4ht TEX to HTML converter.

3 Unix installation

As introduced in section 1.1 (p. 2), TEX Live has three principal uses:

1. Run directly from media.
2. Install to disk.
3. Integrate a particular package or collection into your existing TEX installation.

The following sections describes the Unix-specific procedures for each of these.

Warning: The TEX Collection CDs and DVD are in ISO 9660 (High Sierra) format, *with* Rock Ridge (and Joliet, for Windows) extensions. Therefore, in order to take full advantage of the TEX Collection under Unix, your system needs to be able to use the Rock Ridge extensions. Please consult the documentation for your mount command to see how to do this. If you have several different machines on a local network, you may be able to mount the discs on one which does support Rock Ridge, and use this with the others.

Modern systems should be able to use the discs without problems. If troubles, let us know. The discussion below assumes you have been able to mount the CDs with full Rock Ridge compatibility.

3.1 Running TEX Live directly from media (Unix)

It is possible to use the TEX system directly from the live DVD, without installing the distribution to disk. (Thus the name TEX 'Live', in fact.) It is *not* possible to run TEX directly from the other CDs (see section 2.1, p. 4). To start, you mount the CD or DVD, with Rock Ridge extensions enabled. The exact command to do this varies from system to system; the following works under Linux, except the name of the device (`/dev/cdrom`, here) may vary. (All our examples will use `>` as the shell prompt; user input is `underlined`.)

```
> mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Change the current directory to the mount point:

```
> cd /mnt/cdrom
```

Under Mac OS X, the directory is typically under `/Volumes`, and the media will be mounted automatically. Run the installation script `install-tl.sh`:

```
> sh install-tl.sh
Welcome to TeX Live...
```

After various greeting messages and a list of the main menu options, the installation will ask you to enter a command. Do this by typing the desired character and hitting return; don't type the angle brackets shown. Either uppercase or lowercase is ok; we'll use lowercase in our examples.

For running live, our first command will be `d` and then the subcommand `1` to set directories. Even in this case, we must choose a directory on the local disk to place files that the TEX system itself generates, such as fonts and formats, and also to provide a place for updated configuration files, if need be.

We'll use `/opt/texlive2006` in this example. It's good to include the year in the name, as these generated files are not in general compatible from release to release. (If the default value of `/usr/local/texlive/2006` works for you, then you can skip this step.)

```
Enter command: d
Current directories setup:
<1>  TEXDIR:     /usr/local/texlive/2006
...
Enter command: 1
New value for TEXDIR [/usr/local/texlive/TeX]: /opt/texlive2006
...
Enter command: r
```

Back at the main menu, our second and last command is `r`, to set up for running live off the media without installing to disk:

```
Enter command: r
Preparing destination directories...
...
Welcome to TeX Live!
>
```

And we are back at the system prompt, as shown.

Next, it is necessary to alter two environment variables: `PATH`, to an architecture-dependent value (so that we can run the programs), and `TEXMFSYSVAR`, to the value specified above. See table 1 for a list of the architecture names for the different systems.

After the main installation has completed, and environment variables have been set, the last step is to run texconfig or texconfig-sys to customize your installation to your needs. This is explained in section 4.1, p. 13.

Table 1: Supported system architectures.

`alpha-linux`	HP Alpha GNU/Linux
`i386-darwin`	Intel x86 Mac OS X
`i386-freebsd`	Intel x86 FreeBSD
`i386-linux`	Intel x86 GNU/Linux
`mips-irix`	SGI IRIX
`powerpc-aix`	IBM RS/6000 AIX
`powerpc-darwin`	PowerPC Mac OS X
`sparc-linux`	Sun Sparc GNU/Linux
`sparc-solaris`	Sun Sparc Solaris
`win32`	Windows (32-bit)
`x86_64-linux`	Intel x86 64-bit GNU/Linux

The syntax for setting the environment variables, and the initialization file to put them in, depends on the shell you use. If you use a Bourne-compatible shell (sh, bash, ksh, et al.), put the following into your `$HOME/.profile` file:

```
PATH=/mnt/cdrom/bin/archname:$PATH; export PATH
TEXMFSYSVAR=/opt/texlive2006/texmf-var; export TEXMFSYSVAR
```

For C shell-compatible shells (csh, tcsh), put the following into your `$HOME/.cshrc` file:

```
setenv PATH /mnt/cdrom/bin/archname:$PATH
setenv TEXMFSYSVAR /opt/texlive2006/texmf-var
```

Then log out, log back in, and test your installation (see section 4.2, p. 14).

If in doubt, please ask any local system gurus to help you with problems; for example, the way to mount the TEX Live media, which directory or directories to use, and precise details of the changes to your personal initialization files can and do vary from site to site.

3.2 Installing TEX Live to disk

It is possible, indeed typical, to install the TEX Live distribution to hard disk. This can be done from either the live or inst distributions. (See section 2.1, p. 4, for an explanation of the different distributions.)

To start, you mount the CD or DVD, with Rock Ridge extensions enabled. The exact command to do this varies from system to system; the following works under Linux, except the name of the device (`/dev/cdrom`, here) may vary. (All our examples will use `>` as the shell prompt; user input is underlined.)

```
> mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Change the current directory to the mount point:

```
> cd /mnt/cdrom
```

Under Mac OS X, the directory is typically under `/Volumes`, and the media will be mounted automatically. Run the installation script `install-tl.sh`:

```
> sh install-tl.sh
Welcome to TeX Live...
```

After various greeting messages and a list of the main menu options, the installation will ask you to enter a command. Do this by typing the desired character and hitting return; don't type the angle brackets shown. Either uppercase or lowercase is ok; we'll use lowercase in our examples.

Table 2 briefly lists the options in the main menu. The order in which you select the options makes little difference, except that `i` must be last. It's reasonable to go through them in the order presented here.

Table 2: Main menu options for the installation.

`p`	The platform you are running on.
`b`	The architectures for which to install binaries.
`s`	The base installation scheme to use (minimal, recommended, full, etc.)
`c`	Override the base scheme for individual collections.
`l`	Override for language collections.
`d`	Directories in which to install.
`o`	Other options.
`i`	Perform the installation.

Here are further details on each option.

`p` – **Current platform.** Since the installation script automatically guesses which platform you're running on, it is usually unnecessary to use this option. It's there in case you need to override the automatic detection.

`b` – **Binary architectures.** By default, only the binaries for your current platform will be installed. From this menu, you can select installation of binaries for other architectures as well (or omit installing the current platform). This can be useful if you are sharing a TEX tree across a network of heterogenous machines. For a list of the supported architectures, see table 1, p. 8.

`s` – **Base installation scheme.** From this menu, you can choose an overall set of package collections, called a "scheme". The default `full` scheme installs everything available, but you can also choose the `basic` scheme for a minimal system, or `medium` to get something in between. There are also specific sets for Omega and XML.

`c` – **Individual collections.** From this menu, you can override the scheme's set of collections to install. Collections are one level more detailed than schemes — collections consist of one or more packages, where packages (the lowest level grouping in TEX Live) contain the actual TEX macro files, font families, and so on. In this menu, selection letters are case-sensitive.

`l` – **Language collections**. This menu has the same basic purpose as `c`, to override the collection set in the chosen scheme. In this case, the collections are specifically for different languages. Selection letters are case-sensitive here too. Here is a list of the language collections in TEX Live:

(some) African scripts	Arabic	Armenian	Chinese Japanese Korean
Croatian	Cyrillic	Czech/Slovak	Danish
Dutch	Finnish	French	German
Greek	Hebrew	Hungarian	Indic
Italian	Latin	Manju	Mongolian
Norwegian	Polish	Portuguese	Spanish
Swedish	Tibetan	UK English	Vietnamese

Language collections typically include fonts, macros, hyphenation patterns, and other support files. (For instance, frenchle.sty is installed if you select the `French` collection.) In addition, installing a language collection will alter the `language.dat` configuration file controlling which hyphenation patterns are loaded.

`d` – **Installation directories**. Three directories can be changed here:

`TEXDIR` The top-level directory under which everything else will be installed. The default value is `/usr/local/texlive/2006`, and is often changed. We recommend including the year in the name, to keep different releases of TeX Live separate. (You may wish to make a version-independent name such `/usr/local/texlive` via a symbolic link, which you can then update after testing the new release.)

Under Mac OS X, the usual frontends look for TeX in `/usr/local/teTeX`, so you may wish to install TeX Live there.

`TEXMFLOCAL` This tree is where the TeX system (not as part of the initial installation, but rather as time goes by) puts non-version-specific files, primarily fonts. The default value is `/usr/local/texlive/texmf-local`, independent of the current TeX Live release, because it's also the recommended location to put any local packages or configuration settings.

`TEXMFSYSVAR` This tree is where texconfig-sys puts files that *are* version-specific. The default value is `TEXDIR/texmf-var`, and there's generally no reason to change it. There is also `TEXMFSYSCONFIG`, which is where texconfig looks for modified configuration data. See section 4.1, p. 13 for more information.

`o` – **Other options.** From this menu, you can select the following general options:

`a` Specify an alternate directory for generated fonts. The default is to use the `TEXMFVAR` tree, as explained above. Setting this is useful if you plan to mount the main tree read-only, and therefore you need another location (perhaps host-specific) for dynamically created fonts.

`l` Create symbolic links for the binaries, man pages, and/or GNU Info files in other locations. For example, you may wish to make the man pages available under `/usr/local/man` and the Info files available under `/usr/local/info`. (Of course you need appropriate privileges to write in the specified directories.)

It is not advisable to overwrite a TeX system that came with your system with this option. It's intended primarily for creating the links in standard directories that are known to users, such as `/usr/local/bin`, which don't already contain any TeX files.

`d` Skip installation of the font/macro documentation tree. This is useful if you need to save space, or if you've previously installed the documentation elsewhere.

`s` Skip installation of the main font/macro source tree. This is useful if you are arranging to share that tree between machines and/or architectures in some other way, such as NFS.

i – Perform installation. When you're satisfied with your configuration options, enter `i` to actually do the installation from the media to your chosen locations.

After the installation completes, your next step is to include the architecture-specific subdirectory of `TEXDIR/bin` in your `PATH`, so the newly-installed programs can be found. The architecture names are listed in table 1, p. 8, or you can simply list the directory `TEXDIR/bin`.

The syntax for doing this, and the initialization file to use, depends on your shell. If you use a Bourne-compatible shell (sh, bash, ksh, et al.), put the following into your `$HOME/.profile` file:

```
PATH=/usr/local/texlive/2006/bin/archname:$PATH; export PATH
```

For C shell-compatible shells (csh, tcsh), put the following into your `$HOME/.cshrc` file:

```
setenv PATH /usr/local/texlive/2006/bin/archname:$PATH
```

After the main installation has completed, and environment variables have been set, the last step is to run texconfig or texconfig-sys to customize your installation to your needs. This is explained in section 4.1, p. 13.

Here is a minimal annotated example which accepts the default directories and installs binaries for the current system only. Thus, only one command is needed, `i` for install. The `>` is the shell prompt as usual.

```
> sh install-tl.sh
i                          # perform installation
> texconfig ...            # see section 4.1
# New PATH element, with Linux as the example:
> PATH=/usr/local/texlive/2006/bin/i386-linux:$PATH; export PATH
```

If in doubt, please ask any local system gurus to help you with problems; for example, the way to mount the TEX Live media, which directory or directories to use, and precise details of the changes to your personal initialization files can and do vary from site to site.

3.2.1 Non-interactive installation

It is possible to override the default directories with environment variables, and then install non-interactively. Example:

```
> TEXLIVE_INSTALL_PREFIX=/opt/texlive
> export TEXLIVE_INSTALL_PREFIX
> echo i | sh install-tl.sh
```

The `TEXLIVE_INSTALL_PREFIX` variable overrides the default location of `/usr/local/texlive`, leaving all else unchanged—so with the above invocation, the main installation will go to `/opt/texlive/2006`.

In the usual Unix way, the final `echo i` can be replaced by any sequence of input commands via a here document, so any sequence of commands can be scripted.

Here are all the possible overrides:

`TEXLIVE_INSTALL_PREFIX` Override `/usr/local/texlive`.

`TEXLIVE_INSTALL_TEXDIR` Override `\$TEXLIVE_INSTALL_PREFIX`/2006.

`TEXLIVE_INSTALL_TEXMFLOCAL` Override `\$TEXLIVE_INSTALL_PREFIX/texmf-var`.

`TEXLIVE_INSTALL_TEXMFSYSVAR` Override `\$TEXLIVE_INSTALL_TEXDIR/texmf-var`.

`TEXLIVE_INSTALL_TEXMFHOME` Override `\$HOME/texmf`.

It would be better to support a standard GNU-style `configure` with options, instead of these environment variables. Volunteers are welcome!

3.3 Installing individual packages to disk

You can add individual packages or collections from the current distribution to an existing non-TEX Live setup, or an earlier TEX Live installation.

To start, you mount the CD or DVD, with Rock Ridge extensions enabled. The exact command to do this varies from system to system; the following works under Linux, except the name of the device (`/dev/cdrom`, here) may vary. (All our examples will use `>` as the shell prompt; user input is `underlined`.)

```
> mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Change the current directory to the mount point:

```
> cd /mnt/cdrom
```

Under Mac OS X, the directory is typically under `/Volumes`, and the media will be mounted automatically.

Run the installation script `install-pkg.sh` (not `install-tl.sh`, which is intended for complete installations only):

```
> sh install-pkg.sh options
```

The first set of options controls what gets read:

`--package=`*pkgname* The individual package to work on.

`--collection=`*colname* The individual collection to work on.

`--nodoc` Exclude documentation files from the operation.

`--nosrc` Exclude source files from the operation.

`--cddir=`*dir* Source directory to read from; defaults to the current directory. If you followed the instructions above, that will be the distribution directory, and won't need to be changed.

`--listdir=`*dir* The so-called 'lists' directory within *cddir* from which to read the package information. The only reason to change the default is if you're making changes to TeX Live yourself.

What actually happens is controlled by the following options. If neither of these are specified, the default action is to install the selected files. The main destination tree is found by expanding `$TEXMFMAIN` with kpsewhich. You can override it by setting either the environment variable `TEXMFMAIN` or `TEXMF`.

`--listonly` List the files that would be installed, but don't actually install anything.

`--archive=`*tarfile* Instead of installing the files into the TeX system, make a tar archive.

Additional options:

`--config` After installation, run `texconfig init`.

`--nohash` After installation, don't run mktexlsr to rebuild the filename database.

`--verbose` Give more information as the script runs.

Here are some usage examples:

1. To see the files in the package fancyhdr without installing it:

   ```
   > sh install-pkg.sh --package=fancyhdr --listonly

   texmf/doc/latex/fancyhdr/README
   texmf/doc/latex/fancyhdr/fancyhdr.pdf
   ...
   ```

2. Install the LaTeX package natbib:

   ```
   > sh install-pkg.sh --package=natbib
   ```

3. Install the LaTeX package alg without source files or documentation:

   ```
   > sh install-pkg.sh  --package=alg --nosrc --nodoc
   ```

4. Install all the packages in the collection of additional plain TeX macros:

   ```
   > sh install-pkg.sh --collection=tex-plainextra
   ```

5. Write all files in the pstricks package to a tar file in `/tmp`:

   ```
   > sh install-pkg.sh --package=pstricks --archive=/tmp/pstricks.tar
   ```

If in doubt, please ask any local system gurus to help you with problems; for example, the way to mount the TeX Live media, which directory or directories to use, and precise details of the changes to your personal initialization files can and do vary from site to site.

4 Post-installation

After the main installation is done, for any operating system, the remaining steps are to configure the system for your local needs, and perform some basic tests.

Another sort of post-installation is to acquire packages, fonts, or programs that were not included in TeX Live. The basic idea is to install such additions in the `TEXMFLOCAL` tree (if you installed to disk), or `TEXMFSYSVAR` (if you are running live). See the "Installation directories" option on p. 9.

Unfortunately, the details can and do vary widely, and so we do not attempt to address them here. Here are some external links to descriptions:

- http://www.tex.ac.uk/cgi-bin/texfaq2html?label=instpackages
- http://www.ctan.org/installationadvice
- http://www.ctan.org/tex-archive/info/beginlatex/html/chapter5.html#pkginst
- http://www.ctan.org/tex-archive/info/Type1fonts for installation information for fonts in particular.

4.1 The texconfig program

At any time after installation, you can and should use the program texconfig to configure the system to fit your local needs. It is installed in the architecture-specific subdirectory `TEXDIR/bin/`*arch* along with everything else.

If you run it without arguments, it will enter full-screen mode and allow you to view and change options interactively.

You can also run it with various command-line options. Here are some of the most common (TeX Live is configured for the A4 paper size by default):

`texconfig paper letter` Set default paper size for various programs and drivers (pdftex, dvips, dvipdfm, xdvi) to be US letter. The other allowable size is a4, which is the default.

`texconfig rehash` Update all the TeX "filename databases", after adding or removing files.

`texconfig faq` Show the teTeX FAQ. (See also the main TeX FAQ in texmf-doc/doc/english/FAQ-en.)

`texconfig help` Output help information for texconfig.

Of course, texconfig can only support changing a few of the many options and configuration parameters in a TeX system. The main configuration file for the base Web2C programs is named `texmf.cnf`. You can find its location by running '`kpsewhich texmf.cnf`'; it contains many comments explaining the default settings and useful alternatives.

texconfig alters files in a user-specific directory, as in `$HOME/.texlive2006`. If you install TeX just for yourself, that is unlikely to make a difference. But if you install TeX on a multi-user system, you will want to change the configuration for the whole system. In this case, run texconfig-sys instead of texconfig.

Likewise, the updmap and fmtutil scripts were changed, to work under `$HOME/.texliveYYYY`. To alter system directories, use updmap-sys and fmtutil-sys.

In particular, for multi-user systems, you will probably want to pregenerate the standard formats with `fmtutil-sys -missing`. Otherwise, each user will end up with their own formats.

Also, if you have a personally-modified copy of `fmtutil.cnf` or `updmap.cfg`, instead of using the ones generated by installation, they must be installed in the tree referenced by the variable `TEXMFSYSCONFIG`.

The variables specifying the directories altered by these commands are listed in section 2.3, p. 5. You can see the actual directories by running `texconfig conf`, and you can change them by editing `texmf.cnf`.

4.2 Testing the installation

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

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

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

   ```
   > tex --version
   TeX 3.141592 (Web2C 7.5.5)
   kpathsea version 3.5.5
   ...
   ```

   If this comes back with 'command not found' instead of version and copyright information, most likely you don't have the correct `bin` subdirectory in your `PATH`. See the environment-setting information on p. 7.

2. Process a basic LATEX file:

   ```
   > latex sample2e.tex
   This is pdfeTeXk, Version 3.141592...
   ...
   Output written on sample2e.dvi (3 pages, 7496 bytes).
   Transcript written on sample2e.log.
   ```

   If this fails to find `sample2e.tex` or other files, perhaps you have interference from old environment variables or configuration files. For a deep analysis, you can always ask TEX to report on exactly what it is searching for, and finding; see "Debugging actions" on page 33.

3. Preview the result online:

   ```
   > xdvi sample2e.dvi
   ```

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

4. Create a PostScript file for printing or display:

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

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

   ```
   > pdflatex sample2e.tex
   ```

6. Preview the PDF file:

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

Unfortunately neither gv nor xpdf are currently included in TeX Live, so you must install them separately. See http://www.gnu.org/software/gv and http://www.foolabs.com/xpdf, respectively.

7. Other standard test files you may find useful:

   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'.

   You can process these in the same way as we did with sample2e.tex.

If you are new to TeX, or otherwise need help with actually constructing TeX or LaTeX documents, please visit http://tug.org/begin.html for some introductory resources.

5 Mac OS X installation

The recommended way to install TeX on Mac OS X is from the MacTeX distribution, new in 2005. This is provided on the live DVD in the top-level mactex/ directory. It contains its own (native) installer for a full TeX distribution, based on a combination of teTeX and TeX Live, along with many additional applications and documentation. The project web page is http://tug.org/mactex.

If you prefer, installation of TeX under Mac OS X can also be done directly from TeX Live, using the install* scripts, as follows.

In order to run the installation scripts under Mac OS X, you need to have the bash shell installed. If you are running Mac OS X 10.2 or later, you have bash, and can proceed. If you're running an earlier Mac OS X version, however, the default shell is zsh, which won't work; you'll need to get bash from the Internet, or more likely upgrade your system.

Once you have bash, the Unix installation documentation in the previous section can be followed. See section 3 on p. 6; Mac OS X-specific notes are included there where needed.

6 Windows installation

In this release of TeX Live, happily, the distribution once again has a native Windows installer, named tlpmgui.exe. (See section 2.1, p. 4, for an explanation of the different distributions.)

tlpmgui has essentially the same options as the Unix installer, only done through a GUI interface: selecting schemes, individual collections, installation directories, and so on. Section 3.2 on p. 8 describes the basic elements. It also allows some post-installation activities such as adding/removing packages, updating the filename database and building formats. Moreover, tlpmgui can setup the system for running programs directly from the DVD.

For those who like to look underneath the hood, tlpmgui uses as its "engine" a command-line Windows program named tlpm.

The Windows TeX system included in TeX Live is based on new binaries borrowed from the W32TEX distribution, kindly provided by Akira Kakuto. It also includes some older (but still working) tools made by Fabrice Popineau, and also a new dvi previewer, dviout by Toshio Oshima.

TeX Live can be installed on systems running Windows 98, ME, NT, 2K or XP. Older versions of Windows (3.1x) and MS-DOS are not supported.

Warning: Win9.x users must ensure they have enough environment space before undertaking installation. The tlpmgui.exe program won't change the environment size for them. A few environment variables will be created and it is possible you run out of environment space. Add `SHELL=<path>COMMAND.COM /E:4096 /P` in the config.sys file in order to increase your environment size.

6.1 Installing TEX Live to disk

After inserting the TEX Live CD into the CD drive, autostart should activate tlpmgui. If it does not, click Start→Run, then type `<drive letter>:\setup-win32\tplmgui.exe` (or `<drive letter>:\texlive\setup-win32\tplmgui.exe` if you are installing from the TeX Collection DVD), where `<drive letter>` is the drive letter with the TeX Live CD (or TeX Collection DVD), and then click OK.

The installation window titled TeX Live installation and maintenance utility should open. It contains the following sections: Main customization, Install, Select a scheme, Select systems, Directories and Options.

In the Directories section the installation drive (directory) next to the CD/DVD button should be displayed (e.g., `F:/` or `F:/texlive/` for the DVD), but if it is not, then click the CD/DVD button and select the CD/DVD drive, with the TEX Live CD (or TeX Collection DVD).

The directory in which you wish to install the software can be set by clicking the TLroot button. This directory will be set as `TLroot` environment variable for later usage. The `TEXMFTEMP` and `TEXMFCNF` environment variables as displayed next to the TEXMFTEMP and TEXMFCNF buttons will be adjusted automatically and set during installation, but they can also be adjusted manually now to suit special needs.

In the Select a scheme section the desired TEX Live installation scheme should be chosen by clicking the radio button labelled with the installation scheme name (e.g., scheme-gust). Each scheme is accompanied by an Info button which, when clicked, displays a short description of the relevant scheme.

A scheme is a large set of files targeted at some kind of usage. There are generic schemes for basic, medium and full installations. The remaining ones are either targeted at certain LUGs (i.e., what GUST or GUTenberg propose for their members) or application targeted (e.g., for XML and TEX cooperation). A preselected scheme can be refined. This is done in the Main customization section by choosing additional collections from Standard collections or Language collections. For example, by clicking the Select button labelled Standard collections, additional collections like Metapost, Omega or documentation in different languages can be selected.

Note: The Ghostscript, Perl and Wintools collections are selected by default and should be installed unless they are already installed and you really know what you are doing. This is because they are required by many important tools. The `PERL5LIB` and `GS_LIB` environment variables will be set too.

Next, clicking the Select button labelled Language Collections in the Main customization section opens the Language collections window in which the installation language can be chosen by ticking the box next to the language.

Next, click the Install button in the Install section to start the installation proper process.

The TEX Live system needs some post-processing steps (format files generation, ls-R databases generation, environment variables, etc.). All these operations are done there, some of them can be lengthy. So please wait until you see a statement about the successfully finished installation.

The shortcut for tlpmgui will be added to the Start→Programs→TeXLive2006 menu.

If it is needed (Win9x/WinME), you will be asked to reboot your computer.

6.2 Support packages for Windows

To be complete, a TEX Live installation needs support packages that are not commonly found on a Windows machine. Many scripts are written using the Perl language. Some important tools require the Ghostscript PostScript interpreter to render or to convert files. A graphic file toolbox is also needed in some cases. Last but not least, a TEX-oriented editor makes it easy to type in your TEX files.

All these tools are quite easy to find for Windows, but in order to try to make your life even easier, we have put such a set of tools on TEX Live:

- GNU Ghostscript 8.54
- a minimal Perl 5.8, sufficient to run all the TEX Live Perl scripts.
- win-tools is a set of small programs (bzip2, gzip, jpeg2ps and tiff2png)

Perl and Ghostscript are installed upon request; you may explicitly deselect them during installation if you already have them.

If you don't want to install this bundle, then you are on your own to install the required tools to complete your TEX Live system. Here is a list of places to get those tools:

Ghostscript http://www.cs.wisc.edu/~ghost/

Perl http://www.activestate.com/ (but you might need some supplementary packages from CPAN: http://www.cpan.org/)

ImageMagick http://www.imagemagick.com

NetPBM alternatively, you could use NetPBM instead of ImageMagick to process or convert your graphic files. NetPBM home page is http://netpbm.sourceforge.net/

TEX-oriented editors There is a wide choice, and it is a matter of the user's taste. Here is a selection:

- GNU Emacs is available natively under Windows, see http://www.gnu.org/software/emacs/windows/ntemacs.html
- XEmacs is available natively under Windows, see http://www.xemacs.org/
- WinShell is available on TEX Live in the `support` directory, see http://www.winshell.de
- WinEdt is shareware available from http://www.winedt.com
- Vim is available on TEX Live in the `support\vim` directory and the reference site is http://www.vim.org
- TeXnicCenter is free software, available from http://www.toolscenter.org and in the proTeXt distribution.
- LEd is available from http://www.ctan.org/support/LEd
- SciTE is available from http://www.scintilla.org/SciTE.html

You might want to install other tools that are not free[1] and therefore not included on TEX Live, such as GSView, the Ghostscript companion to more conveniently view PS/PDF files. GSView is available from http://www.cs.wisc.edu/~ghost/gsview/ or any CTAN site.

7 Maintenance of the installation in Windows

If you have TeX Live installed, you can use tlpmgui again for modifying and maintaining your installation.

[1] Not free, that is, in the sense of freedom to modify and redistribute, following free software guidelines. This does not mean you can't acquire them for no money.

7.1 Adding/removing packages

As the tlpmgui shortcut is available in the Start→Programs→TeXLive2006 menu, start it from here. The maintenance window titled TeX Live installation and maintenance utility shows. It contains several tabs: Add Packages, Remove packages, Manage installation, Remove installation.

Click the tab labelled Add packages or Remove packages to enter the relevant functionality and then:

- In the first tab select the proper CD drive (or DVD with `texlive` directory) by pressing the CD/DVD button.
- Click the Search button in the Buttons section to display or refresh in the Select packages to... section the list of packages to install or remove.

  When adding packages, the list of installed packages is compared to the list of packages available from your CD/DVD. Only packages not already installed are displayed. It is up to you to select which packages you want to install.

  When removing individual packages, only a list of installed packages will be displayed.

  Please note that for both Add packages and Remove packages the collections are listed first.
- Select a package by clicking on the name of the package. Clicking the Info button in the Buttons section displays a short description of the selected package in the window located in the Info about the selected item section. To select several packages at once hold down one of the Ctrl or Shift keys at your keyboard while holding down the left mouse button or drag the mouse pointer while holding down the left mouse button.
- Click the Install or Remove button in the Buttons section to perform the appropriate action.

7.2 Configuring and other management tasks

The functions available in the tab labelled Manage the installation are helpful in performing actions needed when you want to add support for a language which was not selected during the installation, or add/regenerate a format which was not selected during the installation or was modified after the installation.

The following actions can be performed:

- Refresh the `ls-R` database
- Create formats (All or Missing)
- Edit `language.dat`
- Edit `fmtutil.cnf`
- Edit `updmap.cfg`

Note: you can close the Edit... window with the Cancel or Done button; the latter will start rebuilding the format files (or the fontmap files if you have edited `updmap.cfg`), followed by a `ls-R` database files refresh.

For more information about the configuration see section 7.8, p. 20.

7.3 Uninstalling TEX Live

The tab labelled Remove the TeX Live installation opens a window which contains functionality not worth describing and we do not know who would need it and what it is for. . . :-)

Anyway, if you have the `texmf-local` directory for your local additions, the removal procedure will not wipe it out or delete things in it. The `setup-win32` directory containing tlpmgui and related files will not be deleted. You will have to do some manual cleanup to actually remove them.

7.4 Adding your own packages to the installation

First, whatever changes you make, **do not forget to rebuild the ls-R database files**. Otherwise, your new files will not be found. You can do this either via the tlpmgui run and selection of the appropriate action from the Manage the installation tab, or manually running the `mktexlsr` command.

If you want to add files that are not provided by the TEX Live distribution, it is recommended to put them in the `$TEXMFLOCAL` directory. This way, you will be safe against upgrades of the TEX Live software.

The directory pointed to by `$TEXMFLOCAL` is initially empty. If, for example, you want to add the support file for the Maple symbolic computation program, you will have to put the style files in:
`c:\TeXLive2006\texmf-local\tex\latex\maple\`
and the documentation files in:
`c:\TeXLive2006\texmf-local\doc\latex\maple\`

7.5 Running `tlmp.exe` from the command line

The tlpm.exe program which is used as engine by tlpmgui has a number of other useful options. You can get the list by running:

```
tlpm --help
```

More information and examples of use can be found in `tlpm.readme`.

7.6 Network installation

Kpathsea knows about UNC names, so you can use them to get your TEXMF tree from the network. But there is better than this. All the support files and configuration files, everything except the files in the `bin/win32` are shareable with a teTEX or Unix TEX Live installation. That means you can use Samba either to mount from an NT server to a Unix workstation or the converse. Several strategies are possible:

- Put everything on the server. Just add each set of files for the OS and architecture you want to use in the `bin` directory. That means for example `bin/win32` and `bin/i386-linux`. Next configure your main variables. You can use UNC names to point to the right directories under Win32.
- Install a local copy for the binaries and format files. In this case, assign `$TEXMFMAIN` to the main `texmf` tree that will be accessed remotely. Set `$TEXMFVAR` to be a local directory which will hold local configuration files and on-the-fly generated files.

7.7 What's different in Windows?

The Windows version of Web2C has some specific features that should be pointed out.

Kpathsea The hash-tables that Kpathsea builds are quite large for the TEX Live. In order to cut down the starting time of any Kpathsea-enabled program, these hash-tables have been put in shared memory. This way, when you chain the execution of several such programs, like `tex` calling `mpost` calling `tex`, the overhead when starting each of the subprograms is reduced. This change is hidden to the user, except if you set the debug flag of kpathsea to the `-1` value: you will then trace access to the shared memory, which is not what you want (it is accessed very often!). What is useful in a log trace of the shared memory access is still to be defined, so the situation might evolve in the future.

kpsecheck This command provides some options that did not fit well into kpsewhich. It will allow you to list all the files that occur multiple times across your texmf trees. This could be handy, but most of the time you will also get unwanted output (like dozens of `README` files). It is noticeable that all these files result in clashes inside the Kpathsea-hashing mechanism; fortunately, Kpathsea never looks for these files. For this reason, you can combine the `-multiple-occurrences` with 2 other options for including or excluding any filename that match some pattern (you can request for several patterns).

The kpsecheck command will also report the status of shared memory: in use or not used. That might be useful to know because if the status reported is '`in use`', that means one or several processes are working, and the effect of any mktexlsr command will be delayed until the next time where no Kpathsea linked process will be running.

Last, this same command will report about the location it thinks Ghostscript can be found. Under Win32, for many programs, it is easier to use the Ghostscript dll, and find it by using the Ghostscript registry key than to change the `PATH`, which has a limited length anyway.

Web2C The engines have a few more options than in Unix Web2C, and one option with a different behavior:

- `-halt-on-error` stop the compilation at the first error.
- `-job-time` set the job time to the same timestamp as the file given in argument.
- `-oem` use the DOS codepage for console output.
- `-output-directory` allow to write all the output files in the specified directory.
- `-time-statistics` print statistics about the job run time. It is to be noted that Win9x not being a true multitasking operating system, it has no reliable timer for short periods, so the value printed is an approximation. Under NT/2K/XP, the result is quite accurate with user time and system time values allocated for this run. For Unix users: the `time` command is not usually available to Windows users.

7.8 Personal configurations

7.8.1 Dvips

The configuration file for dvips can be found in
`C:\TeXLive2006\texmf-var\dvips\config\config.ps`
You may open it with any editor and change some parameters:

fonts you can change the default printer METAFONT mode or printer resolution in case dvips needs to generate PK fonts. By default it is configured to use Type 1 versions of the CM fonts, so it should not call mktexpk too often;

printer you can tell dvips where you want to print by default. If the `o` option is not followed by a printer name, then a `.ps` PostScript file is written. You can give dvips a printer name such as:

```
o lpt1:
% o | lpr -S server -P myprinter
% o \\server\myprinter
```

paper Next, you might want to change the paper size from European (A4) to US letter by making the US letter the first paper size mentioned in the file. Scroll to the group of lines beginning with `@`. Move the appropriate lines so that this section begins with the lines:

```
@ letterSize 8.5in 11in
@ letter 8.5in 11in
@+ %%BeginPaperSize: Letter
@+ letter
@+ %%EndPaperSize
```

The current TEX Live distribution implements the procedure of making always up-to-date fontmaps files for Dvips and Pdftex. This is done by the `updmap` program during installation, as well as during any font package addition. If you add new packages by hand, edit the file `updmap.cfg` in `$TEXMFVAR/web2c`.

7.8.2 PdfTeX

If you use the program pdflatex to write PDF format directly, and you are using US letter-size paper, edit the file `C:\TeXLive2006\texmf-var\tex\generic\config\pdftexconfig.tex` and change '`\pdfpagewidth`' and '`\pdfpageheight`'. These entries should read:

```
\pdfpagewidth=8.5 true in
\pdfpageheight=11 true in
```

Save the file and exit the editor.

7.8.3 GSView

GSView is now distributed under the Aladdin License, and therefore is no longer included in TEX Live.

If you may want to change the page size to US letter size. If so, open GSView from the Start menu, and select Media→Letter.

Also, there are menu settings that are supposed to give you the most readable screen image. On Media→Display Settings, set both `Text Alpha` and `Graphics Alpha` to 4 bits.

Note that the installation process has set all `.ps` and `.eps` files to automatically open with GSView.

For printing instructions, see section 7.10 below.

7.9 Testing

For generic verification procedures, see section 4.2 (p. 14). This section describes Windows-specific tests.

Open the file `sample2e.tex` in your editor (Xemacs, WinShell), found in `C:\TeXLive2006\texmf-dist\tex\latex\base`. The LATEX source should appear on the screen. Process it by clicking on the Command→LaTeX menu (XEmacs) or LATEX icon on the toolbar (WinShell), then view it by clicking on the Command→View DVI menu (XEmacs) or Preview (dviout) icon (WinShell).

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.

Hint for the future: If a LaTeX run stops because LaTeX cannot find a file, you can press Ctrl-z to quit.

7.10 Printing

It is possible to print from dviout. In this case, printing will be done using the Windows unified printer driver. By definition, it is compatible with all printers. However, there is some drawback: it can generate some huge spool files, and some (older) versions of Windows just don't like them. The advantage is that you can use features like embedding BMP or WMF images. You also need to make sure that the printer parameters are correctly set, else you will get scaled printing (printing at 600 dpi on a 300 dpi printer will give you only one quadrant of your page).

Printing is faster and more reliable if you run dvips to make a `.ps` file and then print from GSView. In GSview, select File→Print.... A Print window will appear.

If you are using a PostScript printer, *be sure to select PostScript Printer.* This is done in the Print Method box at the bottom left of the Print window. You can then select any of the printers that you have previously installed. If you fail to check the box for `PostScript Printer`, printing will not work.

If you will be using your own non-PostScript printer, select Ghostscript device in the Print Method box, then click on the button to the right labelled djet500 and select your printer type from the list that pops up. (In the older version of GSView, make sure `PostScript Printer` is *not* selected, then select your printer type from the Device list.)

7.11 Tips and tricks for Win32

7.11.1 Different flavors of Win32

What we call Win32 is not an operating system by itself. It is a large set of functions (around 12,000 in the header files of the Microsoft SDK) that you can use to write programs for different operating systems of the Windows family.

Windows comes in different flavors:

- Win95, Win98 and WinME, which *are not true multitasking, multithreading* environments, but rather updated incarnations of DOS. This can be more or less proven by the fact that when booting, the PC will load the `command.com` interpreter, and if you stop the boot process at this point, you can ask for the current (DOS) version and it will answer something like '`MS-DOS 7.0`' (at least for the old versions of Win9x).
- Windows NT, which is a new operating system written from scratch, capable of true multitasking behavior, and including many high level features.
- Windows 2000, based on NT, with all the bells and whistles of Win98.
- Windows XP, which comes with Personal and Pro flavors. This is the last step in merging both lines of products (Win9x based and NT based). XP is based on NT.

Win9x are able to run 32 bits programs and 16 bits programs concurrently. But the operating system by itself is not entirely written in 32 bits mode, and does not support memory protection: 16bits applications can overwrite parts of the operating system memory! Some parts of the system like the GDI (Graphical Device Interface) manage limited resources like bitmaps, fonts, pens and so on for the set of all programs that run concurrently. All the bitmaps headers available at the

same time can't amount for more than 64kB. This explains the performance tool and the fact that you can put your system on his knees by making intensive use of graphic objects for example.

NT, 2K and XP do not suffer from these limitations, and neither from other Win9x limitations. They are true multitasking environments, with protected memory. They are much more responsive than Win9x because of better memory management, better file system and so on.

7.11.2 Command line prompt

You may wonder, "Why would I need to use a command line prompt when I have Windows?"

Good question. The problem is of very general nature. Not all operations can be done easily using only a GUI. Command line gives you programming power — assuming a clever command interpreter.

But the problem here is more fundamental: TEX is *a batch* tool. Not an interactive one. TEX needs to compute the best layout for each page, resolve cross-references and so on. This can be done only by a global processing of the document. It is not (yet) a task that can be done interactively.

This means that you should use TEX from a command line. In fact the situation is not so bad. There is an advantage to write command line tools for complex processing: they are better debugged, because they are independent of any GUI problems, and GUI tools can be designed to interface to the command line tools. This is the case for TEX, where you will mostly interact with it through a GUI text editor.

However, you may need to use the command line prompt in a number of situations. One is when you are having difficulties and want to debug your setup.

Win9x You can open a command line prompt by looking either for the MS-DOS icon in the Start→Programs menu, or by choosing Start→Run menu typing in `command.com` as the progrm name.

NT, 2K, XP You can open a command line prompt by looking for Command Prompt in the Start→Accessories menu. You can also choose Start→Run and type in `cmd.exe`, which is the name of the brand new command interpreter for NT (thus, it is untrue to call this a *DOS* box!).

These locations may change across different Windows versions.

7.11.3 Path separators

The Win32 API understands both `/` and `\` characters as PATH separators. But the command interpreters do not! So whenever a path name is used programmatically, you can use both separators, and even mix them up in the same path name. But on the command line, you must type `\` as path separator. The reason is compatibility: the command processor used '`/`' to introduce arguments to commands.

All this to say: do not be surprised to read path names written using the Unix convention; fpTEX is a port of Web2C, and aims to be compatible across platforms. For this reason, all the configuration files that need to specify path names use the Unix convention.

7.11.4 File systems

The worst feature of Win9x with regard to TEX is probably the so-called FAT file system. TEX uses very many small files, with size around 1–3kB. The FAT file system is old, and predates by decades the multi-gigabytes hard disks we have today. As a result, it cannot manage efficiently the tens of thousands of TEX files found in TEX Live. The FAT file system allocates a minimum of 32kB for *any* file on a huge partition. It means that TEX will use much more disk space than it actually needs.

The other, more modern, file systems available, FAT32 and NTFS, do not have this drawback. They manage clusters of 4kB only. (You can lower the limit to 512 bytes on NTFS.)

7.11.5 How to add some directory to your PATH

There are pairs of variables and values which behave much like global variables inside your programs. The set of those variables is called the environment. Each program is initialized with a copy of the environment when it is run. It can request and change the value of any variable. The changes happen in the copy of the environment, and is not at all propagated to the other running programs.

Your PATH is a special environment variable used to search for programs you want to run. There is a different procedure to change it for Win9x, WinME and NT/2K/XP:

Windows 95/98 Edit your `autoexec.bat`. In this file should be a line starting with `PATH=` and followed by a list of directories separated by `;`. Please add the directory with the executables in this line. After this, this line could look as follows:

```
PATH=c:\windows;c:\windows\system;c:\TeXLive2006\bin\win32
```

Windows ME You need to run the special program `c:\windows\system\msconfig.exe` to be able to change any environment variable. From this program, select the 'Environment' tab, and then add or modify the variable you want. You will be asked to reboot the machine upon any change.

Windows NT/2K/XP Click left on Start→Settings→Control Panel. Now the window with the control panel icons opens. Double click on System. The System Properties window opens. Click on the tab Environment or look for a button named Environment Variables among the dialog boxes. Now you can change the environment variables for your user account. Note: There are also displayed the environment settings for the system. Normally, you can't change the system variables unless you have administrator rights on your machine. If you want to change the `PATH` for all users, you will have to contact your system administrator or be the system administrator yourself — in the latter case you should know what you are doing.

If there is already a `PATH` setting for your user account, left click on `PATH`. In the field Variable appears `PATH` while the field Value shows the current setting of `PATH` as a list of directories separated by `;`. Add the directory where the executables are located (e.g. `c:\TeXLive2006\bin\win32`). If there isn't a `PATH` variable for your user account, simply click in the field Variable and type in `PATH`, click in the field Value and type in the directory with the executables. Important: Click on the Apply button before clicking Ok, otherwise the changes to `PATH` won't apply to your system. Be careful when changing the environment settings.

The best way to be sure that a variable has been properly set is to open a console and type:

```
set VARIABLE
```

which should return the corresponding value.

7.11.6 TEX engines

If you have a look at the Web2C documentation, you will read that all the various TEX derived programs use the same base engine. For example, `tex.exe` and `latex.exe` are exact copies of the same program, but each one will use a different format file, based on its calling name.

Under Unix, this feature is implemented through *symbolic links*. It saves up a bit of disk space, because some engines are used with many different format files.

The Win32 API does not know about file links. So to save up almost the same amount of memory, all the TEX base engines have been put in DLLs (*Dynamic Linked Library*). This means that you will have the following layout:

```
18/09/2005  14:19                 3 584 latex.exe
18/09/2005  14:19                 3 584 pdfetex.exe
18/09/2005  14:19               524 288 tl90pdfetex.dll
```

and the `latex.exe` file is nothing but a rough copy of `pdfetex.exe` using the same core `tl90pdfetex.dll`. The same trick has been used for the `mktex*.exe` family of programs which are linked to the `mktex.dll` library.

In fact, a generic tool called `irun.exe` is provided to build the equivalent of Unix hard links for executable files only under Win32.

7.12 In case of problems

7.12.1 What to do if `latex` does not find your files?

- kpsewhich is the tool of choice to debug any problem. However, kpsewhich outputs debug information to stderr, and the older Windows consoles do not know how to redirect stderr to a file. (Windows NT and later do support redirections, but the trick below will work for any console.) For diagnostic purposes you can temporarily set an environment variable (in a DOS box):

  ```
  SET KPATHSEA_DEBUG_OUTPUT=err.log
  ```

  You can also set the debug level:

  ```
  SET KPATHSEA_DEBUG=-1
  ```

  Similarly, to redirect stderr to stdout:

  ```
  SET KPATHSEA_DEBUG_OUTPUT=con:
  ```

  This way you can capture both stdout and stderr in the same file.

- Assuming the installation has been done in `c:/TeX`, check the following values:

  ```
  kpsewhich -expand-path $SELFAUTOPARENT   c:/TeX
  kpsewhich -expand-path $TEXMF            c:/TeX/texmf....
  kpsewhich -expand-path $TEXMFCNF         .;c:/TeX/texmf-var/web2c;
  kpsewhich -expand-var $TEXINPUTS         .;c:/TeX/texmf/tex//
  ```

- If you have other TEX-related values already set in your environment, please, remove them. They are overriding the ones in texmf.cnf.

- Check the values from:

  ```
  kpsewhich cmr10.tfm   c:/TeX/texmf/fonts/tfm/public/cm/cmr10.tfm
  kpsewhich latex.fmt   c:/TeX/texmf/web2c/latex.fmt
  ```

- At this point, if everything is correct, TEX and friends should work. If it is not the case, you will need to play with the `-debug=n` option from `kpsewhich`, and check back all the values. Try to identify and report the problem.

7.12.2 What to do if your setup still does not work as expected?

Here are several questions to investigate:

1. Is `tex.exe` in my `PATH`?
2. Is the `TEXMFCNF` variable correctly set to `c:/TeXLive2006/texmf-var/web2c` (default value)?
3. Are there any errors in the log file generated by the `tlmpgui.exe` program? `tlmpgui.log` can be found in your `TEMP` directory. You can find this by searching for the string '`Error`'. Hint: the log file can show some errors after building all format files. Please do not panic: perhaps some formats weren't already installed.
4. Are there any relevant bug fixes at `http://tug.org/texlive/`? (Unlikely, but it doesn't hurt to check.)

The TEX Live software consists of hundreds of programs and tens of thousands of files, all from varying sources. So it is quite difficult to predict all possible causes for problems. Nevertheless, we will do our best to help you. (See section 1.2, p. 3.)

8 A user's guide to Web2C

Web2C is an integrated collection of TEX-related programs: TEX itself, METAFONT, MetaPost, BIBTEX, etc. It is the heart of TEX Live.

A bit of history: The original implementation was by Tomas Rokicki who, in 1987, developed a first TEX-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.

The Web2C system runs on Unix, 32-bit Windows systems, Mac OS X, and other operating systems. It uses Knuth's original sources for TEX and other basic programs written in `web` and translates them into C source code. The core TEX programs are:

bibtex Maintaining bibliographies.

dmp troff to MPX (MetaPost pictures).

dvicopy Expands virtual font references in DVI files.

dvitomp DVI to MPX (MetaPost pictures).

dvitype DVI to human-readable text.

gftodvi Generic font proofsheets.

gftopk Generic to packed fonts.

gftype GF to human-readable text.

makempx MetaPost label typesetting.

mf Creating typeface families.

mft Prettyprinting METAFONT source.

mpost Creating technical diagrams.

mpto MetaPost label extraction.

newer Compare modification times.

patgen Creating hyphenation patterns.

pktogf Packed to generic fonts.

pktype PK to human-readable text.

pltotf Plain text property list to TFM.

pooltype Display `web` pool files.

tangle `web` to Pascal.

tex Typesetting.

tftopl TFM to plain text property list.

vftovp Virtual font to virtual property list.

vptovf Virtual property list to virtual font.

weave `web` to TEX.

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

All programs honor these standard GNU options:

`--help` print basic usage summary.

`--verbose` print detailed progress report.

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

For locating files the Web2C programs use the path searching library Kpathsea. This library uses a combination of environment variables and a configuration files to optimize searching the (huge) collection of TEX files. Web2C can look at more than one directory tree simultaneously, which is useful in maintaining TEX's standard distribution and local extensions in two distinct trees. To speed up file searches the root of each tree has a file `ls-R`, containing an entry showing the name and relative pathname for all files under that root.

8.1 Kpathsea path searching

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

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

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

To check a particular path element p, Kpathsea first checks if a prebuilt database (see "Filename database" on page 30) applies to p, i.e., if the database is in a directory that is a prefix of p. If so, the path specification is matched against the contents of the database.

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

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

Although the simplest and most common path element is a directory name, Kpathsea supports additional features in search paths: layered default values, environment variable names, config file

values, users' home directories, and recursive subdirectory searching. Thus, we say that Kpathsea *expands* a path element, meaning it transforms all the specifications into basic directory name or names. This is described in the following sections in the same order as it takes place.

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

8.1.1 Path sources

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

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

You can see each of these values for a given search path by using the debugging options (see "Debugging actions" on page 33).

8.1.2 Config files

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

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

  variable`[.`*progname*`]` `[=]` *value*

  where the '`=`' and surrounding whitespace are optional.
- The *variable* name may contain any character other than whitespace, '`=`', or '`.`', but sticking to '`A-Za-z_`' is safest.
- If '`.`*progname*' is present, the definition only applies if the program that is running is named *progname* or *progname*`.exe`. This allows different flavors of TEX to have different search paths, for example.
- *value* may contain any characters except `%` and '`@`'. The `$`*var*`.`*prog* feature is not available on the right-hand side; instead, you must use an additional variable. A '`;`' in *value* is translated to '`:`' if running under Unix; this is useful to be able to have a single `texmf.cnf` for Unix, MS-DOS and Windows systems.

- All definitions are read before anything is expanded, so variables can be referenced before they are defined.

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

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

8.1.3 Path expansion

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

8.1.4 Default expansion

If the highest-priority search path (see "Path sources" on page 28) contains an *extra colon* (i.e., leading, trailing, or doubled), Kpathsea inserts at that point the next-highest-priority search path that is defined. If that inserted path has an extra colon, the same happens with the next highest. For example, given an environment variable setting

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

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

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

then the final value used for searching will be:

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

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

8.1.5 Brace expansion

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

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

Using this you can then write something like

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

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

8.1.6 Subdirectory expansion

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

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

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

8.1.7 List of special characters and their meaning: a summary

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

`:` Separator in path specification; at the beginning or the end of a path it substitutes the default path expansion.

`;` Separator on non-Unix systems (acts like `:`).

`$` Variable expansion.

`~` Represents the user's home directory.

`{...}` Brace expansion.

`//` Subdirectory expansion (can occur anywhere in a path, except at its beginning).

`%` Start of comment.

`\` Continuation character (allows multi-line entries).

`!!` Search *only* database to locate file, *do not* search the disk.

8.2 Filename databases

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

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

8.2.1 The filename database

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

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

`cd` */your/texmf/root* `&& \ls -1LAR ./ >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.

8.2.2 kpsewhich: Standalone path searching

The `kpsewhich` program exercises path searching independent of any particular application. This can be useful as a sort of `find` program to locate files in 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 more important options are described next.

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

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

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

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

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

`--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*.

8.2.3 Examples of use

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

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

We are looking for the file `article.cls`. Since the '`.cls`' suffix is unambiguous we do not need to specify that we want to look for a file of type `tex` (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
```

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 prolog `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
utmbo8r NimbusRomNo9L-Medi     ... <utmb8a.pfb
utmro8r NimbusRomNo9L-Regu     ... <utmr8a.pfb
```

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

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

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

8.2.4 Debugging actions

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

- `1` `stat` calls (disk lookups). When running with an up-to-date `ls-R` database this should almost give no output.
- `2` References to hash tables (such as `ls-R` databases, map files, configuration files).
- `4` File open and close operations.
- `8` General path information for file types searched by Kpathsea. This is useful to find out where a particular path for the file was defined.
- `16` Directory list for each path element (only relevant for searches on disk).
- `32` File searches.

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 LaTeX source file, `hello-world.tex`, which contains the following input.

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

```
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 1: Finding configuration files

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

Figure 2: Finding the prolog file

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

Figure 3: Finding the font file

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

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

In this case we have combined dvips's debug class 4 (font paths) with Kpathsea's path element expansion (see dvips Reference Manual, `texmf/doc/html/dvips/dvips_toc.html`). The output (slightly rearranged) appears in Figure 1.

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 TEX, PostScript and file system names of the fonts.

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

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

At this point dvips identifies itself to the user:

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

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

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

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

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

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

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

8.3 Runtime options

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

`main_memory` Total words of memory available, for TeX, METAFONT and MetaPost. You must make a new format file for each different setting. For instance, you could generate a "huge" version of TeX, 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" TeX data structures: boxes, glue, breakpoints, etc. Especially useful if you use PiCTeX.

`font_mem_size` Number of words for font information available for TeX. 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. Approximately 10,000 control sequences can be stored in the main hash table; if you have a large book with numerous cross-references, this might not be enough. The default value of `hash_extra` is `50000`.

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

9 Acknowledgements

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

- The English, German, Dutch, and Polish TeX user groups (TUG, DANTE e.V., NTG, and GUST, respectively), which together provide the necessary technical and administrative infrastructure. Please join your local user group!
- The CTAN team, which distributes the TeX Live images and provides the common infrastructure for package updates, upon which TeX Live depends.
- Peter Breitenlohner and the ε-TeX team for the stable foundation of future TeX's.
- Thomas Esser, without whose marvelous teTeX package TeX Live would certainly not exist, and whose continual help makes it a better product.
- Michel Goossens, who co-authored the original documentation.
- Eitan Gurari, whose TeX4ht was used to create the HTML version of this documentation, and who worked tirelessly to improve it at short notice.
- Hans Hagen, for major testing and making the ConTeXt format conform to TeX Live's needs.
- Hàn Thế Thành, Martin Schröder, and the pdfTeX team for continuing enhancements of TeX's abilities.
- Taco Hoekwater, for renewed development efforts on MetaPost and TeX itself.
- Paweł Jackowski, for the Windows installer tlpm, and Tomasz Łuczak, for tlpmgui.
- Akira Kakuto, for great assistance in incorporating the Windows binaries from his W32TEX distribution (http://www.fsci.fuk.kindai.ac.jp/kakuto/win32-ptex/).
- Jonathan Kew and his employer SIL, for the major new development of XeTeX and taking the time and trouble to integrate it in TeX Live.
- Reinhard Kotucha, for helping with the massive task of updating packages in TeX Live, as well as Windows research efforts, the `getnonfreefonts` script, and more.

- Petr Olsak, who coordinated and checked all the Czech and Slovak material very carefully.
- Toshio Oshima, for his dviout previewer for Windows.
- Fabrice Popineau, for the original Windows support in TEX Live.
- Norbert Preining, for helping with the TEX Live infrastructure and package updates, and coordinating the Debian version of TEX Live (together with Frank Küster), making many suggestions for improvement along the way.
- Staszek Wawrykiewicz, the principal tester for all of TEX Live, and coordinator of the many major Polish contributions: fonts, Windows installation, and more.
- Olaf Weber, for his patient assembly and maintenance of Web2C, upon which all else depends.
- Gerben Wierda, for creating and maintaining the Mac OS X support, and much integration and testing.
- Graham Williams, on whose work the catalogue of packages depends.

Builders of the binaries: Tigran Aivazian (x86_64-linux), Manfred Lotz (i386-freebsd), Fabrice Popineau (win32), Norbert Preining (alpha-linux), Vladimir Volovich (powerpc-aix, sparc-linux, sparc-solaris), Staszek Wawrykiewicz (i386-linux), Olaf Weber (mips-irix), Gerben Wierda (i386-darwin, powerpc-darwin).

Documentation and translation updates: Karl Berry (English), Daniel Flipo & Fabrice Popineau (French), Günter Partosch & Hartmut Henkel (German), Petr Sojka & Jan Busa (Czech/Slovak), Boris Veytsman (Russian), Staszek Wawrykiewicz (Polish).

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

10 Release history

10.1 Past

Discussion began in late 1993 when the Dutch TEX Users Group was starting work on its 4AllTEX 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 4AllTEX CD, but also the TUG Technical Council working group on a *TEX Directory Structure* (http://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 4AllTEX 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 teTEX 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 teTEX, and we decided to base the 2nd edition of the CD on the standard Web2C, with the addition of teTEX'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 teTEX was being made, and TEX Live included almost all of its features. The 4th edition followed the same pattern, using a new version of teTEX, and a new release of Web2C (7.3). The system now included a complete Windows setup.

For the 5th edition (March 2000) many parts of the CD were revised and checked, updating hundreds of packages. Package details were stored in XML files. But the major change for 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 (http://www.debian.org/intro/free); we have done our best to check the license conditions of all packages, but we would very much appreciate hearing of any mistakes.

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

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

10.1.1 2003

In 2003, with the continuing flood of updates and additions, we found that 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. 4). In addition:

- At the request of the LATEX team, we changed the standard latex and pdflatex commands to now use ε-TEX (see p. 5).
- 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 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`.
- 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 TEX Live 7, this translation was dependent on the system locale settings; now, locale settings do not influence the 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: TEX Live 2003.

10.1.2 2004

2004 saw many changes:

- If you have locally-installed fonts which use their own `.map` or (much less likely) `.enc` support files, you may need to move those support files.

  `.map` files are now searched for in subdirectories of `fonts/map` only (in each `texmf` tree), along the `TEXFONTMAPS` path. Similarly, `.enc` files are now searched for in subdirectories of `fonts/enc` only, along the `ENCFONTS` path. updmap will attempt to warn about problematic files.

  For methods of handling this and other information, please see http://tug.org/texlive/mapenc.html.

- The TEX Collection has been expanded with the addition of a MiKTEX-based installable CD, for those who prefer that implementation to Web2C. See section 2 (p. 4).

- Within TeX 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. 4), and the `README` files for each.
- All TeX-related input files are now collected in the `tex` subdirectory of `texmf*` trees, rather than having separate sibling directories `tex`, `etex`, `pdftex`, `pdfetex`, etc. See `texmf-doc/doc/english/tds/tds.html#Extensions`.
- Helper scripts (not meant to be invoked by users) are now located in a new `scripts` directory of `texmf*` trees, and searched for via `kpsewhich -format=texmfscripts`. So if you have programs which call such scripts, they'll need to be adjusted. See `texmf-doc/doc/english/tds/tds.html#Scripts`.
- Almost all formats leave most characters printable as themselves via the "translation file" `cp227.tcx`, instead of translating them with the `^^` notation. Specifically, characters at positions 32–256, plus tab, vertical tab, and form feed are considered printable and not translated. The exceptions are plain TeX (only 32–126 printable), ConTeXt (0–255 printable), and the Ω-related formats. This default behavior is almost the same as in TeX Live 2003, but it's implemented more cleanly, with more possibilities for customization. See `texmf/doc/web2c/web2c.html#TCX-files`. (By the way, with Unicode input, TeX may output partial character sequences when showing error contexts, since it is byte-oriented.)
- pdfetex is now the default engine for all formats except (plain) tex itself. (Of course it generates DVI when run as latex, etc.) This means, among other things, that the microtypographic features of pdftex are available in LaTeX, ConTeXt, etc., as well as the ε-TeX 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 LaTeX) or equivalent code, because simply testing whether `\pdfoutput` or some other primitive is defined is not a reliable way to determine if PDF output is being generated. We made this backward compatible as best we could this year, but next year, `\pdfoutput` may be defined even when DVI is being written.
- pdfTeX (`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 pdfTeX manual for more: `texmf/doc/pdftex/manual`.
- The `\input` primitive in tex (and mf and mpost) now accepts double quotes containing spaces and other special characters. Typical examples:

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

  See the Web2C manual for more: `texmf/doc/web2c`.
- 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://www.olsak.net/enctex.html`.

- Aleph, a new engine combining ε-TEX and Ω, is available. A little information is available in texmf-dist/doc/aleph/base and http://www.tex.ac.uk/cgi-bin/texfaq2html?label=aleph. 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 texmf/doc/man/man1/dvipng.1.
- We reduced the cbgreek package to a "medium" sized set of fonts, with the assent and advice of the author (Claudio Beccari). The excised fonts are the invisible, outline, and transparency ones, which are relatively rarely used, and we needed the space. The full set is of course available from CTAN (http://www.ctan.org/tex-archive/fonts/greek/cb).
- oxdvi has been removed; just use xdvi.
- The ini and vir commands (links) for tex, mf, and mpost are no longer created, such as initex. The ini functionality has been available through the command-line option -ini for years now.
- i386-openbsd platform support was removed. Since the tetex package in the BSD Ports system is available, and GNU/Linux and FreeBSD binaries were available, it seemed volunteer time could be better spent elsewhere.
- On sparc-solaris (at least), you may have to set the LD_LIBRARY_PATH environment variable to run the t1utils programs. This is because they are compiled with C++, and there is no standard location for the runtime libraries. (This is not new in 2004, but wasn't previously documented.) Similarly, on mips-irix, the MIPSpro 7.4 runtimes are required.

10.1.3 2005

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

- New scripts texconfig-sys, updmap-sys, and fmtutil-sys were introduced, which modify the configuration in the system trees. The texconfig, updmap, and fmtutil scripts now modify user-specific files, under $HOME/.texlive2005. See section 4.1, p. 13.
- Corresponding new variables TEXMFCONFIG and TEXMFSYSCONFIG to specify the trees where configuration files (user or system, respectively) are found. Thus, you may need to move personal versions of fmtutil.cnf and updmap.cfg to these places; another option is to redefine TEXMFCONFIG or TEXMFSYSCONFIG in texmf.cnf. In any case the real location of these files and the values of TEXMFCONFIG and TEXMFSYSCONFIG must agree. See section 2.3, p. 5.
- Last year, we kept \pdfoutput and other primitives undefined for DVI output, even though the pdfetex program was being used. This year, as promised, we undid that compatibility measure. So if your document uses \ifx\pdfoutput\undefined to test if PDF is being output, it will need to be changed. You can use the package ifpdf.sty (which works under both plain TEX and LATEX) to do this, or steal its logic.
- Last year, we changed most formats to output (8-bit) characters as themselves (see previous section). The new TCX file empty.tcx now provides an easier way to get the original ^^ notation if you so desire, as in:

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

- The new program dvipdfmx is included for translation of DVI to PDF; this is an actively maintained update of dvipdfm (which is also still available for now, though no longer recommended).
- The new programs pdfopen and pdfclose are included to allow reloading of pdf files in the Adobe Acrobat Reader without restarting the program. (Other pdf readers, notably xpdf, gv, and gsview, have never suffered from this problem.)
- For consistency, the variables `HOMETEXMF` and `VARTEXMF` have been renamed to `TEXMFHOME` and `TEXMFSYSVAR`, respectively. There is also `TEXMFVAR`, which is by default user-specific. See the first point above.

10.2 Present

In 2006, the major new addition to TeX Live was the XeTeX program, available as the `xetex` and `xelatex` programs; see http://scripts.sil.org/xetex.

MetaPost also received a notable update, with more planned for the future (http://tug.org/metapost/articles), likewise pdfTeX (http://tug.org/applications/pdftex)

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).

The installation scripts now accept various environment variables to allow for non-interactive installation; see section 3.2.1.

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

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

Finally, in May 2006 Thomas Esser announced that he would no longer be updating teTeX (http://tug.org/tetex). As a result, there was been 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.

10.3 Future

TeX Live is not perfect! (And never will be.) We intend to continue to release new versions, and would like to provide more help material, more utilities, more installation programs, and (of course) an ever-improved and checked tree of macros and fonts. This work is all done by hard-pressed volunteers in their limited spare time, and a great deal remains to be done. If you can help, don't hesitate to put your name forward!

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

tex-live@tug.org
http://tug.org/texlive

Happy TeXing!