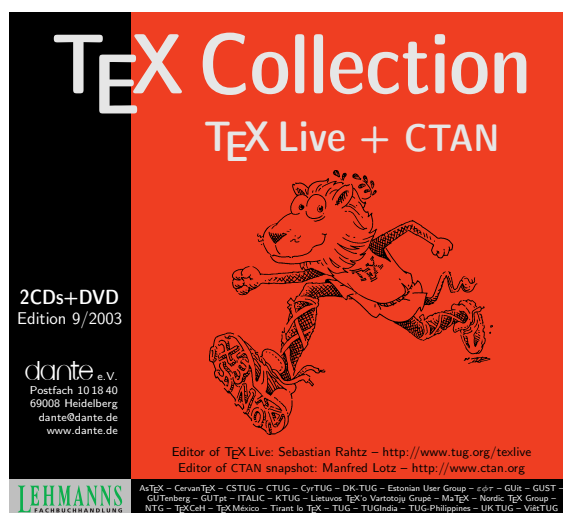


TEX-Collection 2003

The TEX Live Guide

Sebastian Rahtz, editor

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



Documentation contacts:

Czech/Slovak	Petr Sojka	sojka@fi.muni.cz
	Janka Chlebková	chlebkj (at) dcs.fmph.uniba.sk
English	Karl Berry	karl@freefriends.org
French	Fabrice Popineau	fabrice.popineau@supelec.fr
German	Volker RW Schaa	volker@dante.de
Polish	Staszek Wawrykiewicz	staw@gust.org.pl
Russian	Boris Veytsman	borisv@lk.net

9 January 2004

Contents

1	Introduction	3
1.1	Basic usage of T _E X Live	3
1.2	Getting help	3
2	Structure of T_EX Live	4
2.1	Multiple distributions: live, inst, demo	4
2.2	Top level directories	5
2.3	Extensions to T _E X	5
2.4	Other notable programs in T _E X Live	5
3	Unix installation	6
3.1	Running T _E X Live directly from media (Unix)	6
3.2	Installing T _E X Live to disk	8
3.3	Installing individual packages to disk	10
4	Post-installation	12
4.1	The texconfig program	12
4.2	Testing the installation	13
5	Mac OS X installation	14
5.1	i-Installer: Internet installation	14
5.2	install*.sh: T _E X Live installation	15
5.3	Mac OS X frontends	15
5.4	bash installation for older Mac OS X versions	15
6	Windows installation	16
6.1	The TeXLive.exe program	16
6.2	Running T _E X Live directly from media (Windows)	17
6.3	Support packages for Windows	17
6.4	Installing T _E X Live to disk	18
7	Windows maintenance and post-installation	21
7.1	What's different in Windows?	21
7.2	Adding packages to your installation	23
7.3	Uninstalling T _E X Live	23
7.4	Running TeXSetup.exe from the command line	23
7.5	Network installation	24
7.6	Personal configurations	25
7.7	Testing	26
7.8	Printing	26
7.9	Tips and tricks for Win32	26
7.10	In case of problems	29
7.11	Compiling the source files	30
7.12	Where to get more information	31
8	A user's guide to Web2C	31
8.1	Kpathsea path searching	32
8.2	Filename databases	35
8.3	Runtime options	42

9 Building on a new Unix platform	42
9.1 Prerequisites	42
9.2 Configuration	43
9.3 Running make	43
10 Last words	44
10.1 Acknowledgements	44
10.2 Release history	44
10.3 Future releases	46

List of Tables

1 Supported system architectures.	7
2 Main menu options for the installation.	8
3 Kpathsea file types.	36

1 Introduction

This document describes the main features of the T_EX Live software distribution—a T_EX and L^AT_EX distribution for 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.) It includes precompiled binaries for T_EX, L^AT_EX 2_ε, METAFONT, MetaPost, BIB_TE_X and many other programs; an extensive collection of macros, fonts and documentation; and support for typesetting in many different scripts from around the world.

For the 2003 release, the last update of packages and programs was made on September 3, 2003. For newer versions, please consult CTAN, <http://www.ctan.org>.

For a brief summary of the major changes in this edition of T_EX Live, see section 10.2 on p. 44.

1.1 Basic usage of T_EX Live

You can use T_EX Live in three principal ways:

1. You can run T_EX Live directly from the distribution media (except for the inst distribution; see section 2.1 on p. 4). This takes almost no disk space, as you might expect, and gives you immediate access to everything in T_EX Live. Of course performance will be worse than running on local disk, but you may well find it acceptable.
2. You can install all or part of T_EX Live to a local disk. This is the most common use of T_EX Live. You will need a minimum of 120 megabytes, 360 megabytes for a recommended system, and 800 megabytes for a full system.
3. You can integrate a particular package or collection into your existing T_EX system, either a T_EX Live system you installed earlier or a different system.

All of these are described in detail in the OS-specific installation sections.

1.2 Getting help

The T_EX 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 commercial-style support, you can forego T_EX Live completely and purchase a vendor's system; see <http://tug.org/interest.html#vendors> for a list.)

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

T_EX FAQ The T_EX FAQ is a huge compendium of answers to all sorts of questions, from the most basic to the most arcane. It is included on T_EX Live in [FAQ/english](#), and is available on the Internet through <http://faq.tug.org>. Please check here first.

T_EX Catalogue If you are specifically looking for a package, font, program, etc., the T_EX Catalogue is the place to look. It is a huge list of all T_EX-related items. See [texmf/doc/html/catalogue](#), or <http://www.ctan.org/tex-archive/help/Catalogue/catalogue.html>.

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

support archives The two principal forums for support are the Usenet newsgroup [news:comp.text.tex](#) and the mailing list texhax@tug.org. So, their archives have thousands of past questions and answers for your searching pleasure. See <http://groups.google.com/groups?group=comp.text.tex> and <http://tug.org/mail-archives/texhax>, respectively. A query on a general web search engine, such as <http://google.com>, never hurts.

posting questions If you cannot find an answer, you can post to [comp.text.tex](#) through Google or your newsreader, or through email to texhax@tug.org. But before you post, *please* read this FAQ entry for advice 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>.

T_EX Live support If you want to report a bug or have suggestions or comments on the T_EX 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 T_EX 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@tug.org are open to anyone, so feel free to join, start reading, and help out where you can. Welcome to T_EX!

2 Structure of T_EX Live

The main two installation scripts for Unix and Mac OS X are `install-tl.sh` and `install-pkg.sh`. We discuss them in section 3 on p. 6. Here, we describe the structure and contents of T_EX Live.

2.1 Multiple distributions: live, inst, demo

As of 2003, space limitations of CD-ROM format have forced us to divide T_EX Live into three distributions, as follows.

live a complete, runnable system on DVD; it is too large for CD-ROM. (The DVD also includes a snapshot of the CTAN repository, completely independent of T_EX Live.)

inst(allable) a complete system on CD; in order to make it fit, we had to compress everything we could. Therefore, it is not possible to run T_EX directly from the installable CD, you have to install it to disk (hence its name). Installation is described in subsequent sections.

demo a live system runnable directly from CD; in order to make this fit, we omitted the very large collection of CJK (Chinese, Japanese, Korean) language support, support for typesetting music, some less-commonly used fonts, and included executables only for Linux, Mac OS X, and Windows systems.

You can tell which type of distribution you're in by looking for a `00type.TL` file in this top-level directory.

2.2 Top level directories

Here is a brief listing and description of the top level directories in the T_EX Live distribution.

bin	The T _E X system programs, arranged by platform.
Books	Examples from some of the books about T _E X (see Books/README).
FAQ	Current versions of major FAQ collections.
info	A few manuals in GNU Info format, where available.
Mac OS X	Supporting software for Mac OS X (see section 5 on p. 14).
man	Unix man pages.
source	The source of all programs, including the main Web2C T _E X and METAFONT distributions. These are stored in a bzip2-compressed tar archive.
support	assorted auxiliary packages and programs. These are <i>not</i> installed by default. This includes Ghostscript, netpbm, and assorted editors and T _E X shells.
texmf	root of installed packages, fonts, config files, etc.
usergrps	Material about a few of the T _E X user groups. (Visit http://tug.org/usergroups.html for a current list.)
xemtex	The XEmacs editor and other support programs for Windows (see section 6.3 on p. 17). These programs generally come pre-installed on Unix systems, or are at least easy to compile.

2.3 Extensions to T_EX

T_EX Live contains three extended versions of T_EX:

ε-T_EX 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 T_EX--X_ET extensions for bidirectional typesetting. In default mode, ε-T_EX is 100% compatible with ordinary T_EX. See [texmf/doc/etex/base/etex_man.pdf](#). ε-T_EX is now the default for L^AT_EX and pdfL^AT_EX.

pdfT_EX writes Acrobat PDF format as well as DVI. The L^AT_EX hyperref package has an option ‘`pdftex`’ which turns on all the program features. See [texmf/doc/pdftex/pdftex-1.pdf](#) and [texmf/doc/pdftex/base/example.tex](#).

Ω (Omega) 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/doc/omega/base/doc-1.8.tex](#) (not completely up-to-date).

2.4 Other notable programs in T_EX Live

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

bibtex	bibliography support.
makeindex	index support.
dvips	convert DVI to PostScript.
xdvi	DVI previewer for the X Window System.
dvilj	HP LaserJet driver.
dv2dt, dt2dv	convert DVI to/from plain text.
dviconcat, dviselect	cut and paste pages from DVI files.
dvipdfm	convert DVI to PDF, an alternative approach to pdfT _E X (mentioned above). See the <code>ps4pdf</code> and <code>pdftricks</code> packages for still more alternatives.

psselect, psnup, ... PostScript utilities.
 lacheck L^AT_EX syntax checker.
 texexec ConT_EXt and PDF processor.
 tex4ht T_EX to HTML converter.

3 Unix installation

As introduced in section 1.1 on p. 3, T_EX Live has three principal uses:

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

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

Warning: The T_EX-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 T_EX-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 CDs on one which does support Rock Ridge, and use this with the others.

Linux, FreeBSD, Sun, SGI and Alpha systems should be able to use the CDs without problems. We would appreciate receiving detailed advice from users on other systems who succeed, for future versions of this documentation.

The discussion below assumes you have been able to mount the CDs with full Rock Ridge compatibility.

3.1 Running T_EX Live directly from media (Unix)

Only Linux, Mac OS X, and Windows binaries are included on the demo CD; if you want to run live on other Unix systems, you'll need to use the DVD.

To start, we must 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. 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 `/usr/local/texmf-local` in this example. (If the default value of `/usr/TeX` works for you, then you can skip this step.)

```
Enter command: d
Current directories setup:
<1> TEXDIR:      /usr/TeX
...
Enter command: 1
New value for TEXDIR [/usr/TeX]: /usr/local/texmf-local
...
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 the TeX Live system!
>
```

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 `VARTEXMF`, to the value specified above. See table 1 for a list of the architecture names for the different systems, and whether they are available on the demo CD. All systems are available in the `inst` and `live` distributions. (In addition to the version-specific names listed here, there are generic names without the version numbers. For instance, `sparc-solaris` links to `sparc-solaris2.7`. The generic names can be used to protect against the version numbers changing in the future, if you wish.)

After the main installation has completed, and environment variables have been set, the next step is to run `texconfig` to customize your installation for your needs. This is explained in section 4.1, p. 12.

Table 1: Supported system architectures.

<code>alpha-linux</code>	HP Alpha Linux	
<code>alphaev5-osf4.0d</code>	HP Alphaev5 OSF	
<code>i386-freebsd4.8</code>	Intel x86 FreeBSD	
<code>i386-linux</code>	Intel x86 GNU/Linux	demo CD
<code>i386-openbsd3.3</code>	Intel x86 OpenBSD	
<code>i386-solaris2.8</code>	Intel x86 Solaris	
<code>mips-irix6.5</code>	SGI IRIX	
<code>powerpc-aix4.3.3.0</code>	IBM RS/6000 AIX	
<code>powerpc-darwin6.3</code>	Mac OS X	demo CD
<code>sparc-solaris2.7</code>	Sun Sparc Solaris	
<code>sparc64-linux</code>	Sun Sparc Linux	
<code>win32</code>	Windows (32-bit)	demo CD

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
VARTEXMF=/usr/local/texmf-local/texmf-var; export VARTEXMF
```

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

```
setenv PATH /mnt/cdrom/bin/archname:$PATH
setenv VARTEXMF /usr/local/texmf-local/texmf-var
```

Then log out, log back in, and test your installation (see section 4.2 on p. 13).

If in doubt, please ask any local system gurus to help you with problems; for example, the way to mount the T_EX 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 T_EX Live to disk

It is possible, indeed typical, to install the T_EX system from the T_EX Live to disk. This can be done either from the live DVD, or the inst CD. It can also be done from the demo CD, if you don't need the omitted packages or systems. (See section 2.1 on p. 4 for an explanation of the different distributions.)

To start, we must 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. Either uppercase or lowercase is ok; we'll use lowercase in our examples.

Here is an introductory list of 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 system you are running on.
- b The architectures to install binaries for.
- s The base installation scheme to use (minimal, recommended, full, etc.)
- c Override the installation scheme for individual collections.
- l Override for language collections.
- d Directories in which to install.
- o General 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, this is generally unnecessary to override. It's there in case the automatic detection fails.

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 not install the current platforms). This is often useful if you are sharing a T_EX tree across a network of heterogenous machines. For a list of the supported architectures, see table 1, p. 7.

s – Base installation scheme. From this menu, you can choose an overall common set of packages. The default is a recommended set for typical needs, but you can also choose a basic set to conserve disk space, or a full set to get absolutely everything. The Live scheme is used for creating the T_EX Live demo distribution itself, and isn't generally useful to select for a particular site. There are also specific sets for Omega and XML users.

c – Individual collections. From this menu, you can override the basic scheme's choice of which collections to install. Each collection — TeX macro files, Metafont font families, and so on — consists of several packages. In this menu, selection letters are case-sensitive.

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

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

Language collections typically includes 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 hyphenations are loaded.

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

TEXDIR By default, the top-level directory under which everything else will be installed. The default value is `/usr/TeX`, and is often changed. For example, changing to a value such as `/usr/local/texlive2003` would make it possible to keep different releases of T_EX Live separate. You may then wish to make `/usr/local/texlive` a symbolic link, after testing the new release.

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

TEXMFLOCAL This tree is where the T_EX system scripts install non-version-specific files, primarily fonts. The default value is `TEXDIR/texmf-local`. It's also the recommended location to put any local packages or configuration settings. Therefore, it's desirable to change it to a location independent of the current T_EX Live release; for example, `/usr/local/texmf-local`.

VARTEXMF This tree is where the scripts install files that *are* version-specific, primarily format files and the configuration files which are modified by `texconfig` (see section 4.1, p. 12). The default value is `TEXDIR/texmf-var`, and there's generally no reason to change it.

o – General options. From this menu, you can select three general options affecting the installation:

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

- 1 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.)
- d Skip installation of the font/macro documentation tree. This is useful 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 or automount.

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.

The last 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. 7, or you can simply list the directory `TEXDIR/bin`.

The syntax for doing this, and the initialization file to put it 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=/TEXDIR/bin/archname:$PATH; export PATH
```

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

```
setenv PATH /TEXDIR/bin/archname:$PATH
```

After the main installation has completed, and environment variables have been set, the next step is to run `texconfig` to customize your installation for your needs. This is explained in section 4.1, p. 12.

Here is a brief annotated example which selects a full installation, with binaries for the current system only, with directory changes as recommended above. The prompts and RETURN keys are omitted for brevity.

```
> sh install-tl.sh
s b r                # scheme, full, return to main
d                  # change directories
1 /usr/local/texlive2003 # top-level dir
2 /usr/local/texmf-local # TEXMFLOCAL outside TEXDIR
r                  # return to main
i                  # perform installation
> texconfig ...
# New PATH, assuming Linux:
> PATH=/usr/local/texlive2003/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.

3.3 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. You can do this from the demo CD or the live DVD, but *not* from the inst CD. (See section 2.1 on p. 4.)

To start, we must 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
```

This 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 default is cddir/texmf/tpm/lists; the only reason to change it 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.dvi
texmf/doc/latex/fancyhdr/fancyhdr.pdf
...
```

2. Install the L^AT_EX package `natbib`:

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

3. Install the L^AT_EX 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 T_EX 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 T_EX 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 T_EX Live. The basic idea is to install such additions in the TEXMFLOCAL tree (if you installed to disk), or VARTEXMF (if you are running live). See “Installation directories” on p. 9.

Unfortunately, the details can vary widely, and so we do not attempt to address them here. Please see <http://www.ctan.org/tex-archive/info/beginlatex/html/chapter5.html#pkginst> for a good description, and <http://www.ctan.org/tex-archive/info/Typefonts> for font creation and installation information 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 commonly used:

```
texconfig dvips paper letter Set default paper size for dvips to be letter-size.
```

```
texconfig xdvi us Likewise, for xdvi.
```

```
texconfig rehash Update all the TEX system filename databases.
```

```
texconfig faq Show the teTEX FAQ. (See also the main TEX FAQ in the FAQ subdirectory of TEX Live.)
```

```
texconfig help Output help information.
```

Of course, texconfig only supports changing a few of the many options and configuration parameters in a T_EX 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.

4.2 Testing the installation

After installing T_EX 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 described the 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 (Web2c 7.5.2) 3.141592
kpathsea version 3.5.2
Copyright (C) 1997-2003 D.E. Knuth.
...
```

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 L^AT_EX file:

```
> latex sample2e.tex
>TeX (Web2c 7.5.2) 3.141592
...
Output written on sample2e.dvi (3 pages, 7256 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 T_EX to report on exactly what it is searching for, and finding; see "Debugging actions" on page 39.

3. Preview the result online:

```
> xdvi sample2e.dvi
```

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

4. Create a PostScript file for printing or display:

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

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

```
> pdflatex sample2e.tex
```

6. Preview the PDF file:

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

Unfortunately neither `gv` nor `xpdf` are currently included in T_EX Live, so you must install them separately. See <http://wwwthep.physik.uni-mainz.de/~plass/gv> and <http://www.foolabs.com/xpdf>, respectively.

7. Other standard test files you may find useful:

`docstrip.tex` Produce T_EX source or documentation from a `.dtx` file.
`small2e.tex` A simpler document than `sample2e`, to reduce the input size if you're having troubles.
`testpage.tex` Test if your printer introduces any offsets.
`nfssfont.tex` For printing font tables and tests.
`testfont.tex` Also for font tables, but using plain T_EX.
`story.tex` The most canonical (plain) T_EX 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 T_EX, or otherwise need help with actually constructing T_EX or L^AT_EX documents, please visit <http://tug.org/begin.html>. We especially recommend the *Formatting Information* manual by Peter Flynn, available at <http://www.ctan.org/tex-archive/documentation/beginlatex>.

5 Mac OS X installation

T_EX Live supports Mac OS X, but no prior Macintosh versions. (If you are running an older Mac version, you can view the files by installing the Joliet system extension available from <http://www.tempel.org/joliet>; however, the T_EX Live binaries will not run.)

Installation of T_EX under Mac OS X can be done in two ways:

1. With the `install*` scripts, as with Unix.
2. With the i-Installer included in `MacOSX/II2.dmg`.

Each of these is described in a following section.

In addition, typical use of T_EX under Mac OS X goes through a *frontend*. This is also described below.

5.1 i-Installer: Internet installation

The i-Installer is included in the T_EX Live distribution as an alternative to normal installation. It does not use the contents of the T_EX Live distribution at all; instead, the system (approximately 70 megabytes) is downloaded over the Internet.

One advantage of i-Installer is that it makes updates relatively painless. If you are interested, please see the i-Installer T_EX home page at <http://www.rna.nl/tex.html>.

To use it, mount `./MacOSX/II2.dmg`. Read the documentation, launch it, and install at least *TeX Foundation* and *TeX Programs*. The first will finish without configuration, as soon as the second is installed you will be presented with a graphical interface to setting up your T_EX system.

The i-Installer distribution uses the `teTEX` `texmf` tree with some additions. Due to differences between T_EX Live and `teTEX` you cannot update a T_EX Live installation with an i-Installer *TeX Programs* i-Package.

5.2 install*.sh: T_EX Live installation

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; please see the subsection 5.4 (p. 15) below for instructions on installing `bash`.

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 where needed.

5.3 Mac OS X frontends

Using T_EX on a Macintosh typically goes through a front end program, comprising an execution shell, editor, previewer, and other facilities. Here are the principal frontends:

TeXShop Included in T_EX Live as `./MacOSX/texshop.dmg`. See <http://www.uoregon.edu/~koch/texshop/texshop.html>.

ITeXMac Included in T_EX Live as `./MacOSX/iTeXMac-*.dmg`. See <http://itexmac.sourceforge.net>.

Mac-emacs A port of Emacs to Mac OS X, integrating AucTeX. See <http://www.cs.man.ac.uk/~franconi/mac-emacs>.

The frontends use `/usr/local/tETeX` as the default location; therefore, you must either install T_EX Live there, or change the configuration of the frontend.

5.4 bash installation for older Mac OS X versions

Mac OS X versions 10.1 and earlier do not include `bash` by default, and the default shell does not run the T_EX Live installation scripts properly. This section explains how to install `bash`.

First, check if `bash` is already installed. 'Launch Terminal' (`/Applications/utilities/Terminal`) and type `rehash; which bash`. If the answer is a filename (for example, `/bin/bash`), then `bash` is already installed, and you're done here; go back to the main installation instructions. If the answer is `bash: command not found`, proceed here.

There are two ways to install `bash` if you need it — via the GUI or via command line.

To install via GUI, double-click the `MacOSX/bash.dmg` file in T_EX Live. The disk image (volume) will be mounted. Then start the `i-Installer` application on that volume. You will be asked to authenticate; if you have never seen that before, you might not have enough privileges to install. Just enter your own user name and password. Hit `install`. `Bash` will be installed on your system.

To install via command line:

1. ensure you have admin privileges: log in as the `admin` user, the System Administrator, a user with Admin privileges, using `sudo`, etc.
2. copy `MacOSX/bash.tar.gz` from the T_EX Live distribution to your home directory.
3. Launch Terminal, and run:

```
(cd /usr/local/; sudo tar xvzf ~/bash.tar.gz)
```

You'll be asked for your password, and then `bash` will be installed.

4. Quit Terminal.

After using either installation method, be sure to recheck that `bash` is installed with `rehash; which bash` in a new Terminal window.



Figure 1: “Welcome to TeX Live” window

6 Windows installation

TeX Live can be installed on systems running Windows 9x, ME, NT, 2K or XP. Older versions of Windows (3.1x) and MS-DOS are not supported.

It is necessary to have your Windows set up so that it uses the Microsoft Joliet extensions for reading CDs; simply look at the CD in Explorer and see whether it shows long, mixed-case, file names. If it does not, you must install the Joliet extensions.

The Windows TeX system included in TeX Live is no more and no less than the fpTeX distribution. It includes a dvi previewer, Windvi, which is similar in usage to the established Unix xdvi. The documentation can be found in texmf/doc/html/windvi/windvi.html.

6.1 The TeXLive.exe program

If your computer is configured to let the CD autostart, then a dialog box with a menu bar will popup on the screen, and you will have several choices from there:

- Install TeX on your hard disk
- Do maintenance on your TeX system.
- Remove the TeX system.
- Use TeX off the TeX Live CD or DVD.
- Browse documentation: TeX Live documentation, TUG web pages, fpTeX web pages.

- Run the TeXdocTK application to find specific documentation.

If your CD does not autostart, you can explicitly run the program by double clicking on `bin/win32/TeXLive.exe` on the CD from the explorer window.

6.2 Running TeX Live directly from media (Windows)

It is possible to use the TeX system directly from the demo CD or 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 inst CD (see section 2.1 on p. 4).

To run live under Windows, from the menu, chose `Explore CD-Rom`, then `Run TeX off CD-Rom`. This will launch the XEmacs editor.

XEmacs startup file will set the environment variables needed. XEmacs will also setup a temporary TDS compliant texmf tree in the temporary area of your computer. It is needed to store files that may be built on the fly, such as `pk` bitmap fonts, or format files. Configuration files are copied from the CD to this texmf tree, so that you can edit them if needed. The `ls-R` database is computed for this texmf tree. The AUC-TeX mode will be entered Whenever you will visit or create a TeX file, providing all the bells and whistles for TeX support XEmacs is capable of.

If you run ‘M-x shell’ within XEmacs, then you will have access to all the TeXLive tools from the command line provided by XEmacs.

[For advanced users:] You can also use the small batch file `mkloctex.bat` to be called in a directory `bin\win32` of the CD. From the Start menu select ‘Run’, then browse the CD drive and select `mkloctex.bat`. Before starting it, you should add two parameters separated by a space: the letter of your CD drive and the letter of the drive where you want to install the TeX directory. The whole line should read, e.g., `d:\bin\win32\mkloctex.bat d c`. When installation is complete, please read carefully the information on screen. If you are running Windows 9x/ME, then you will have to restart Windows.

6.3 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:

- Ghostscript 7.07
- a minimal Perl 5.8, sufficient to run all the TeX Live Perl scripts
- a subset of ImageMagick 5.5.6
- the ISpell checker
- XEmacs 21.5.14 with a selection of packages to support TeX typesetting.

These packages should be installed only all at once. The bundle is known as the XEmTeX bundle.

If you didn’t install XEmTeX, Perl and Ghostscript are installed anyway if they are not detected on your machine. This is because they are required by many important tools. The `PERL5LIB` and `GS_LIB` environment variables will be set too.

When you are under XEmacs, the tools provided shadow any other version you might have installed previously. The point here is that `TeXSetup.exe` will not install anything outside of the TeX Live location, neither will it run foreign installers, nor will it try to autodetect installed products using unreliable rules

of thumb. The XEmTeX bundle is standalone and is meant to evolve in the future. The reference site is <http://www.fptex.org/xemtex/>.

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, the reference url is <http://www.gnu.org/software/emacs/windows/ntemacs.html>
- XEmacs is available natively under Windows, the reference url is <http://www.xemacs.org/>
- WinShell is available on TeX Live in the support directory, the reference url is <http://www.winshell.de>
- WinEdt is shareware available from <http://www.winedt.com>
- TeXnicCenter is available from <http://www.toolscenter.org/products/texniccenter/>
- Vim is available on TeX Live in the support\vim directory and the reference site is <http://www.vim.org>
- SciTE is available from <http://www.scintilla.org/SciTE.html>

You might want to install other tools that are not free¹ 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.

6.4 Installing TeX Live to disk

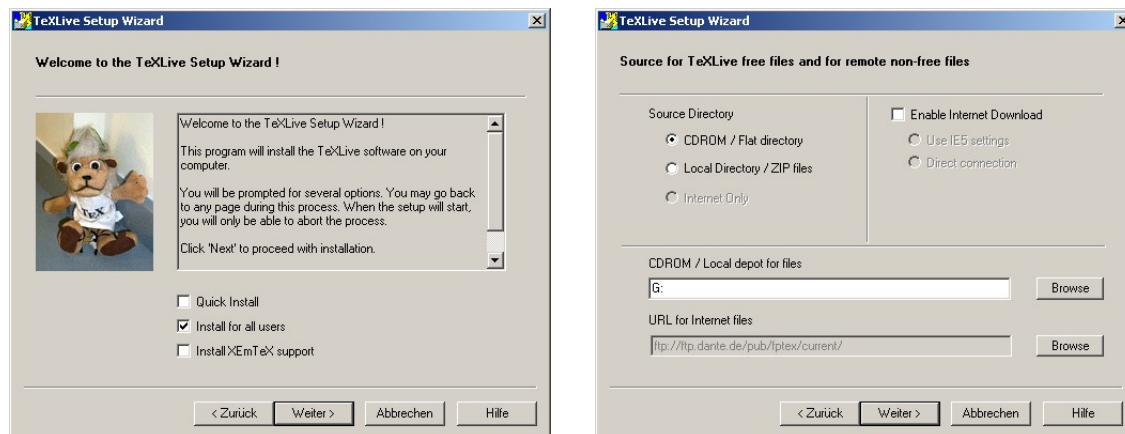
Warning: Win9x users must ensure they have enough environment space before undertaking installation. The TeXSetup.exe program won't change the environment size for them. A few environment variables will be created and it is possible you run out of environment space. Add "SHELL=<path>COMMAND.COM /E:4096 /P" in the config.sys file in order to increase your environment size.

Installation is started by letting the media autostart. The program run is TeXSetup.exe if you are using the inst disk. If you are using the live or the demo disk, you will have to select the item TeXLive Software from the menu, then the subitem Install on Hard Disk. This will invoke TeXSetup.exe. You can also find it in the bin/win32 directory and run it, if the autostart fails for any reason. TeXSetup.exe is a Windows wizard and it will display several pages while running.

Welcome Page You can choose a *quick* installation from here. In this case, the installation will run without any human assistance from beginning to end, with all the default settings (Figure 2, on the left). If you have enough privileges (administrator or power user rights) under a Windows version where this is applicable, then you can decide to install the TeX Live for all users or for yourself only by checking the appropriate box. For installing the whole XEmTeX bundle (XEmacs, Ghostscript, Perl, ImageMagick and Ispell) you can check *Install XEmTeX Support* box.

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

The TeXSetup Wizard

Source directories for the T_EX Live filesFigure 2: The T_EX Live setup wizard

Source Page This page is a bit complex. It will allow you to select two source directories for your T_EX Live system (Figure 2, on the right). You will need a *local source directory* and possibly a *remote source directory*.

Why do we need both these directories? The files consisting of the T_EX Live system are on the CD, of course, but some other useful packages are not, either because of space lacking or because their license was not compatible with T_EX Live. You need to enable Internet downloading if you want to install these packages that are marked with a specific “remote” icon.

However, don’t panic: the default parameters of the setup will allow you to install a full system using the CD only. Simply, you won’t have WinEdt (for example), but you will be able to install it later.

So you can take your files from:

- the CD or any similar flat tree of files, available through some standard directory (this means the CD can be mounted on some remote machine and be made available through network sharing),
- a set of .zip files (this is the case when you are installing from the demo CD or the fpT_EX distribution),
- the Internet; in this case, the program takes care of downloading the .zip files it needs for you.

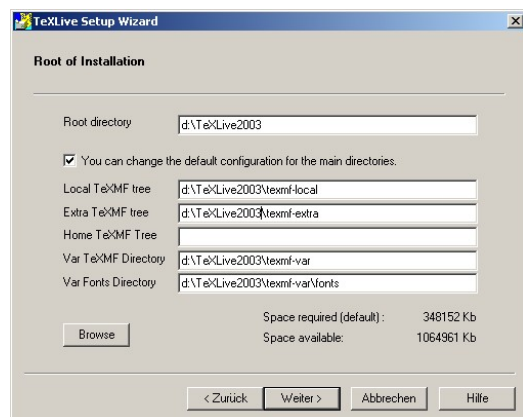
This option is available only if you enable Internet file downloading in the right part of the page. You also need to configure this Internet access by selecting to connect either using Internet Explorer 5 `wininet.dll`, or using a direct connection (`ftp`, `http`).

Root Page On this page, you will tell where you want the files to be installed (Figure 3, on the left). Only the root directory really matters, the other ones are set according to the root one. You may want to make `$TEXMFEXTRA` point to some TDS compliant directory with other T_EX files or assign a different value to `$HOMETEXMF`, which is set by default to whatever Windows think is your ‘HOME’ location.

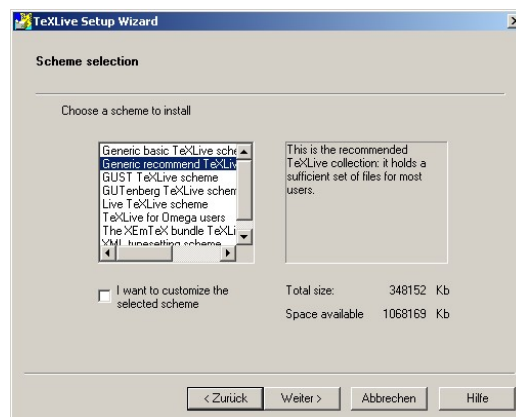
Get TPM Page This page does not require any manual intervention. The .tpm files which describe collections and packages are retrieved (possibly from the Internet), unzipped if needed and parsed.

Schemes Page On this page, you will select the global scheme of your installation (Figure 3, on the right). A scheme is a large set of files targeted at some kind of usage. There are 3 generic schemes for basic,

Root and directories



Scheme selection

Figure 3: T_EX Live-Setup: Root and directories / Schemes

recommended and full installation. The other ones are devoted to LUGs (what GUST or GUTenberg propose for their members) or applications (XML and T_EX). When a scheme is selected, it is still possible to refine the selection by checking the appropriate box. If doing so, you will be presented the packages page to change your selection, else you will jump to the review page.

Packages Page Collections and packages are presented in a tree form (Figure 4, on the left). The links in the tree are dependency links. Collections *depend on* packages and maybe other collections, and it is the same for each package. You can select any package or collection individually, but your request will be granted only if the object is not requested by another one which is selected. For example, you can't deselect `tex-basic` without deselecting all the collections that request it.

The `tex-xemtex` collection displayed on the picture is Win32 specific. It holds a number of bonus packages (Figure 4, on the right) which can be installed automatically and individually: Ghostscript (the PostScript interpreter), XEmacs editor preconfigured for T_EX, tools like Perl, ImageMagick, Ispell. *None of these packages are selected by default.*

On this page, you also have the information about disk space needed, for each object, and for the set of those who are selected, and also the disk space available on the partition selected for the installation. Last, you can choose to install or not the documentation files and source files associated with each package.

Review Page You will find there a summary of your choices (Figure 5, on the left). It is still possible to go back to change them.

Files Copy Page The selected files are copied on your hard disk (Figure 5, on the right). All the files not yet available on your local disk are first downloaded from the remote source directory on the Internet. Then every package is unpacked (if `.zip` files), or copied from the CD.

Configuration Page Several packages need some configuration step to make them usable (Figure 6, on the left). Also the T_EX Live system needs some post-processing step (format files generation, `ls-R` databases generation, environment variables, etc.). All these operations are done there, some of them can be lengthy.

Final Page The installation being over, you may want to display the Windows specific documentation (HTML format) and / or the log file of the setup process (Figure 6 on the right). If it is needed (Win9x/WinME), you will be asked to reboot your computer.

Packages Page

Win32 Support

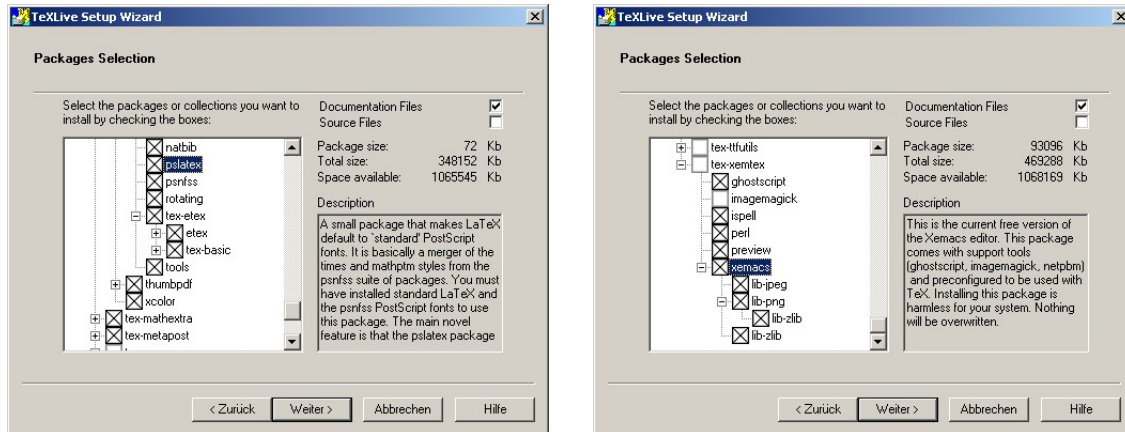


Figure 4: Packages Page / Win32 goodies

Review Page

File Copy Page

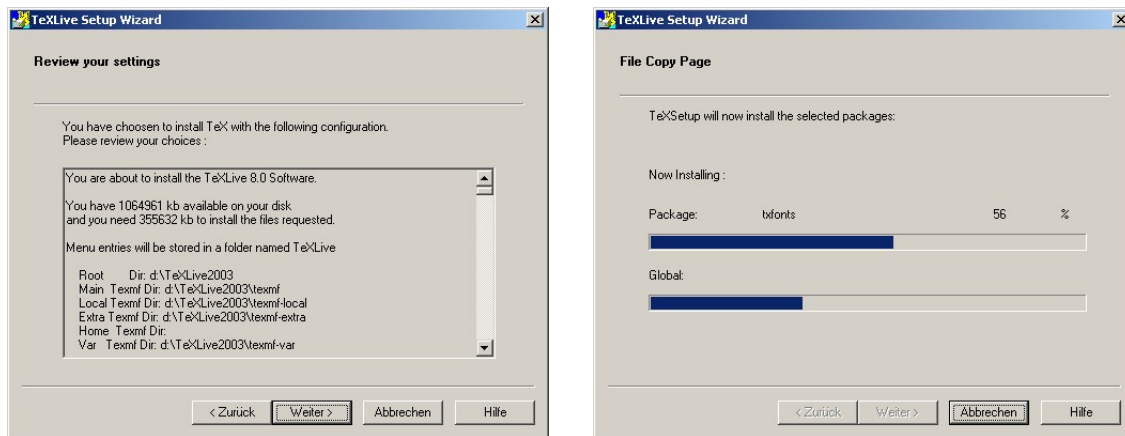


Figure 5: Review Page / File Copy Page

Please be aware that the choice of cluster size on DOS disk partitions can radically affect the size of your T_EX installation. The support tree has hundreds of small files, and it is not unusual for a complete installation to take up to four times the amount of space used on the CD.

7 Windows maintenance and post-installation

7.1 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 T_EX 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

Configuration Page

Final Page

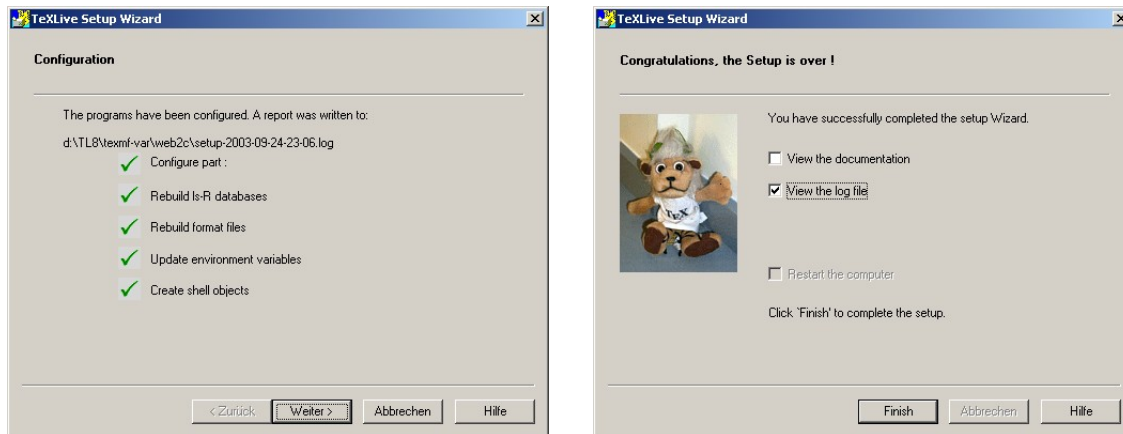


Figure 6: Configuration Page / Final page

`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)². 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.

²It is noticeable that all these files result in clashes inside the `Kpathsea`-hashing mechanism; fortunately, `Kpathsea` never looks for these files.

7.2 Adding packages to your installation

You will find an option in the TeXLive menu (or go to Start→Programs→TeXLive→Add TeX package menu) to run TeXSetup.exe in maintenance mode. The steps are almost the same as in the initial installation.

We'll discuss differences below, but 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 GUI (Start→Programs→TeXLive→Maintenance→Rebuild ls-R), or manually running the `mktexlsr` command.

The main difference is in the packages selection page. In maintenance mode, the list of installed packages is compared to the list of packages available from your source directories. Packages that are not installed will be displayed in green, out of date packages will be displayed in red and up-to-date, installed packages are displayed in black.

This way, you can choose to add or upgrade components, either from TeX Live or from the Internet, where you are likely to find some more recent version of your packages.

It is up to you to select which packages you want to install. The rest of the process is similar to the first installation.

If you want to add files that are not provided by the TeX Live (or fpTeX) 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:\Program Files\TeXLive\texmf-local\tex\latex\maple\
```

and the documentation files in:

```
c:\Program Files\TeXLive\texmf-local\doc\latex\maple\
```

7.3 Uninstalling TeX Live

The uninstall procedure is available either from the TeXLive.exe program, from the TeXLive menu or from the control panel (Start→Control Panel, Add/Remove Programs). This procedure will cleanup your hard disk of most of the TeX Live files. However, TeX is a system that is creating files and there is no mechanism to keep track of all of them. Moreover, Win32 support packages have their own uninstall procedure, which you will have to run separately (provided you want to get rid of them). Last, the files you may have stored in `$TEXMFLOCAL` won't be removed. So, even if the vast majority of files are automatically removed by the uninstall procedure, you will have to do some manual cleanup to actually remove all of them.

7.4 Running TeXSetup.exe from the command line

The TeXSetup.exe program has a number of other useful options. You can get the list by running:

```
c:\>TeXSetup --help
```

Here is the description:

- automatic-reboot reboot without waiting user confirmation once installation is over;
- dry-run do nothing, just log everything that will be done without this option;
- quick use the recommended installation and default directories, ask nothing up to rebooting;
- with-xemtex selects the XEmTeX bundle;
- net-method (=ie5/direct) enable to download components with restricted licenses from the Internet (either using direct connection of Internet Explorer 5 DLLs): you need to have an available network connection and some of the packages are huge;


```

--remote-source-directory <url> this is the base url for the remote packages;
--local-source-directory <dir> by default, TeXSetup.exe will guess the root directory of the
    set of files you want it to act on, if you ever upgrade TeXSetup.exe, you won't be able to copy the
    new version onto your CD, so you will need to use this option to specify the root of the CD;
--installation-directory <dir> this is the root of your installation, all files will be copied under
    this location. The default value is c:\Program Files\TeXLive;
--texmfmain-directory <dir>
--texmflocal-directory <dir>
--texmfextra-directory <dir>
--texmfhome-directory <dir>
--vartexmf-directory <dir>
--vartexfonts-directory <dir> these are the directories used to configure the location of your files.
    They map directly to the texmf.cnf variables.
--with-source-files(=yes/no) copy the source files for TEX packages, default value is no;
--with-documentation-files(=yes/no) copy documentation files for TEX packages. Default value
    is yes. Beware: this is only documentation about specific packages, general documentation will be
    installed anyway;
--program-folder <folder> the name of the folder under which you will find the menus;
--add-package <pkg> this is used to add or update a specific package after a first regular installation;
--scheme <pkg> install the named scheme instead of the default texlive-recommended scheme;
--maintenance mostly the same as --add-package without specifying a package to add;
--uninstall this option will remove anything TEX related coming from the CD, which means there can
    be files left if you added style files or format files, and also that supplementary tools will not be
    removed. This option is still a bit crude in this release;
--help this option opens up a box with the list of options.

```

7.5 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 teT_EX or Unix T_EX 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 \$VARTEXMF to be a local directory which will hold local configuration files and on-the-fly generated files.

7.6 Personal configurations

7.6.1 Dvips

The configuration file for dvips can be found in

```
C:\Program Files\TeXLive\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 `$VARTEXMF/web2c`.

7.6.2 Pdftex

If you use the program `pdflatex` to write PDF format directly, and you are using US letter-size paper, edit the file `C:\Program Files\TeXLive\texmf-var\pdftex\config\pdftex.cfg` and change ‘`page_width`’ and ‘`page_height`’. These entries should read:

```
page_width 8.5 true in
page_height 11 true in
```

Save the file and exit the editor.

7.6.3 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 7.8 below.

7.6.4 WinDvi

The TeXSetup.exe program takes care of associating the files with the .dvi extension with Windvi.

Open it with Start→Programs→TeXLive→DVI Viewer.

You can set it for US letter-size paper by going to View→Options→Papertype and selecting US (8.5" x 11") (and then OK. Exit windvi.

You can change other parameters from there as well, such as the ability to execute system commands specified by the document (disabled by default for security reasons). Also, the first time you view any .dvi file, you may find the magnification too large. Zoom out until you get an appropriate size.

All configuration settings for Windvi are stored in a file named windvi.cnf file. You can find it by running this command at the prompt:

```
c:\>kpsewhich --expand-var $HOME/windvi.cnf
```

Should you have problems with Windvi, please remove the configuration file and test your problem against a vanilla configuration.

7.7 Testing

For generic verification procedures, see section 4.2 on p. 13. This section describes Windows-specific tests.

Open the file sample2e.tex in your editor (Xemacs, WinShell), found in C:\Local\TeX\texmf\tex\latex\base. The L^AT_EX source should appear on the screen. Process it by clicking on the Command→LaTeX menu (XEmacs) or L^AT_EX icon on the toolbar (WinShell), then view it by clicking on the Command→View DVI menu (XEmacs) or Preview (Windvi) icon (WinShell).

At first, when you preview files with Windvi, 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 L^AT_EX run stops because L^AT_EX cannot find a file, you can press Ctrl-z to quit.

7.8 Printing

It is possible to print from Windvi. 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 (subsection 7.6.4), else you will get scaled printing (printing at 600dpi on a 300dpi 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.)

7.9 Tips and tricks for Win32

7.9.1 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. They are the latest — and hopefully last — metamorphosis 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.

7.9.2 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 program 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.

7.9.3 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; fp \TeX 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.

7.9.4 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-gigabyte 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.)

7.9.5 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:\"Program Files"\TeXLive\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:\Program Files\TeXLive\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.

7.9.6 T_EX engines

If you have a look at the Web2C documentation, you will read that all the various T_EX 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 T_EX base engines have been put in DLLs (*Dynamic Linked Library*). This means that you will have the following layout:

```
13/05/2002 17:06          3 584 latex.exe
13/05/2002 17:06        266 240 tex.dll
13/05/2002 17:06          3 584 tex.exe
```

and the `latex.exe` file is nothing but a rough copy of `tex.exe` using the same core `tex.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 `lnexe.exe` is provided to build the equivalent of Unix hard links for executable files only under Win32.

7.10 In case of problems

7.10.1 What to do if latex does not find your files?

- `kpsewhich` is the tool of choice to debug any problem. Unfortunately, `kpsewhich` outputs debug information to `stderr`, and the Windows console does not know how to redirect `stderr` to a file³. 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
```

If you want to redirect `stderr` to `stdout`, which is not possible Windows, then do:

```
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:/Program Files/TeXLive`, check the following values:


```
kpsewhich -expand-path $SELFAUTOPARENT c:/Program Files/TeXLive
kpsewhich -expand-path $TEXMF c:/Program Files/TeXLive/texmf
kpsewhich -expand-path $TEXMFCNF .;c:/Program Files/TeXLive/texmf/web2c;
c:/Program Files/TeXLive/bin/win32;
c:/Program Files/TeXLive/bin;
c:/Program Files/TeXLive
kpsewhich -expand-var $TEXINPUTS .;c:/Program Files/TeXLive/texmf/tex//
```

³Well, NT and Win2k consoles know how to do that. But the trick will work for any console.

- 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:/Program Files/TeXLive/texmf/fonts/tfm/public/cm/cmr10.tfm
kpsewhich latex.fmt c:/Program Files/TeXLive/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.

7.10.2 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:/Program Files/TeXLive/texmf-var/web2c` (default value)?
3. Are there any errors in the log file generated by the `TeXSetup.exe` program? You can find this by searching for the string 'Error'.
4. Are there any relevant bug fixes at <http://tug.org/texlive/>? (Unlikely, but it doesn't hurt to check.)
5. Check the web pages at <http://www.fptex.org/>, or consider subscribing to the `fpTeX` mailing list, via <http://tug.org/mailman/listinfo/fptex>.

The \TeX Live software consists of hundreds and 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 on p. 3.)

7.11 Compiling the source files

You have the whole set of source files, in the `source/source.tar.bz2` archive available in the distribution together with a patch file for Win32. To be able to compile the whole distribution for Windows, you will need:

- Windows 2K/XP
- Microsoft Visual Studio .Net 2003
- a set of Unix tools (`sed`, `grep`, `gawk`, etc.) and also Perl, Flex and Bison,
- to adjust the paths in the `win32/make/common.mak` file according to your installation
- to adjust the paths in the Perl script file `win32/perl/build.pl`,
- run the compilation from the `mwin32/` directory using this command:

```
c:\texlive\source\mwin32>perl ./perl/build.pl --install --log=install.log
```

There is a lot of work to do to make this process easier and cleaner.

7.12 Where to get more information

The Windows T_EX system included on T_EX Live is also known as fpT_EX. The packaging differs, but fpT_EX is no more no less than the current T_EX Live release for Windows.

The fpT_EX home page is <http://www.fptex.org>.

The current fpT_EX release is available from any CTAN site in <http://www.ctan.org/tex-archive/systems/win32/fptex>.

The main ftp site for fpT_EX is <ftp://ftp.dante.de/pub/fptex/> from where beta versions of fpT_EX and additional tools are available. This main site is (partially) mirrored daily by all CTAN sites at [systems/win32/fptex](http://www.ctan.org/tex-archive/systems/win32/fptex).

The T_EX Users Group is kindly hosting a mailing list dedicated to fpT_EX. It is used for announcements, bug reports, enhancement requests, and general discussion about fpT_EX. To subscribe, visit <http://tug.org/mailman/listinfo/fptex>. The mailing list address is fptex@tug.org.

8 A user's guide to Web2C

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

A bit of history: The original implementation was by Tomas Rokicki who, in 1987, developed a first T_EX-to-C system adapting change files under Unix, which were primarily the 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 T_EX and other basic programs written in web and translates them into C source code. The core T_EX 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 T_EX.

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 T_EX files. Web2C can look at more than one directory tree simultaneously, which is useful in maintaining T_EX'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 35) applies to *p*, i.e., if the database is in a directory that is a prefix of *p*. If so, the path specification is matched against the contents of the database.

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

The "file must exist" condition comes into play with '.vf' files and input files read by T_EX'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 39).

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

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

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

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

then the final value used for searching will be:

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

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

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 can be used to implement multiple \TeX hierarchies, by assigning a brace list to `$TEXMF`. For example, in `texmf.cnf`, you find the following definition:

```
TEXMF = {$HOMETEXMF,$TEXMFLOCAL,!!$VARTEXMF,!!$TEXMFMAIN}
```

Using this you can then write something like

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

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

8.1.6 Subdirectory expansion

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

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

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

8.1.7 List of special characters and their meaning: a summary

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

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

8.2 Filename databases

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

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

8.2.1 The filename database

As explained above, the name of the main filename database must be `ls-R`. You can put one at the root of each T_EX 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's `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 L^AT_EX 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 T_EX 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 found in the first column of Table 3, which lists currently recognized names, a description, possible file extensions, and associated environment variables.

Table 3: Kpathsea file types.

<i>Name</i>	<i>Description</i>	<i>Variables</i>	<i>Suffixes</i>
afm	Adobe font metrics	AFMFonts	.afm
base	Metafont memory dump	MFBASES, TEXMFINI	.base
bib	BIB _T E _X bibliography source	BIBINPUTS, TEXBIB	.bib
	bitmap fonts	GLYPHFonts, TEXFonts	
bst	BIB _T E _X style files	BSTINPUTS	.bst
cnf	Runtime configuration files	TEXMFCNF	.cnf
dvips config	dvips configuration files, e.g., config.ps and psfonts.map	TEXCONFIG	.map
fnt	_T E _X memory dump	TEXFORMATS, TEXMFINI	.fnt, .efnt, .efm
gf	generic font bitmap	GFFonts	.gf
graphic/figure	Encapsulated PostScript figures	TEXPICTS, TEXINPUTS	.eps, .epsi
ist	makeindex style files	TEXINDEXSTYLE, INDEXSTYLE	.ist
ls-R	Filename databases	TEXMFDBS	
map	Fontmaps	TEXFONTMAPS	.map
mem	MetaPost memory dump	MPMEMS, TEXMFINI	.mem
mf	Metafont source	MFINPUTS	.mf
mfpool	Metafont program strings	MFPOOL, TEXMFINI	.pool
mft	MFT style file	MFTINPUTS	.mft
	miscellaneous fonts	MISCFonts	
mp	MetaPost source	MPINPUTS	.mp
mppool	MetaPost program strings	MPPPOOL, TEXMFINI	.pool
MetaPost support	MetaPost support files, used by DMP	MPSUPPORT	
ocp	Ω compiled process files	OCPINPUTS	.ocp
ofm	Ω font metrics	OFMFonts, TEXFonts	.ofm, .tfm
opl	Ω property lists	OPLFonts, TEXFonts	.opl
otp	Ω translation process files	OTPIINPUTS	.otp
ovf	Ω virtual fonts	OVFFonts, TEXFonts	.ovf
ovp	Ω virtual property lists	OVPFFonts, TEXFonts	.ovp
pk	packed bitmap fonts	<i>program</i> Fonts (<i>program</i> being XDVI, etc.), PKFonts, TEXPKS, GLYPHFonts, TEXFonts	.pk
PostScript header	downloadable PostScript	TEXPSHEADERS, PSHEADERS	.pro, .enc

Kpathsea file types *continued*

<i>Name</i>	<i>Description</i>	<i>Variables</i>	<i>Suffixes</i>
tex	T _E X source	TEXINPUTS	.tex, .cls, .sty, .clo, .def
TeX system documentation	Documentation files for the T _E X system	TEXDOCS	
TeX system sources	Source files for the T _E X system	TEXSOURCES	
texpool	T _E X program strings	TEXPOOL, TEXMFINI	.pool
tfm	T _E X font metrics	TFMFonts, TEXFonts	.tfm
Troff fonts	Troff fonts, used by DMP	TRFonts	
truetype fonts	TrueType outline fonts	TTFFonts	.ttf, .ttc
Type 1 fonts	Type 1 PostScript outline fonts	T1Fonts, T1INPUTS, TEXPSHEADERS, DVIPSHEADERS	.pfa, .pfb
type42 fonts	Type 42 PostScript outline fonts	T42Fonts	
vf	virtual fonts	VFFonts, TEXFonts	.vf
web2c files	Web2C support files	WEB2C	
other text files	text files used by 'foo'	FOOINPUTS	
other binary files	binary files used by 'foo'	FOOINPUTS	

The last two entries in Table 3 are special cases, where the paths and environment variables depend on the name of the program: the variable name is constructed by converting the program name to upper case, and then appending INPUTS.

The environment variables are set by default in the configuration file `texmf.cnf`. It is only when you want to override one or more of the values specified in that file that you might want to set them explicitly in your execution environment.

The `--format` and `--path` options are mutually exclusive.

`--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/tex/latex/base/article.cls
```

We are looking for the file `article.cls`. Since the `.cls` suffix is unambiguous we do not need to specify that we want to look for a file of type `tex` (T_EX source file directories). We find it in the subdirectory `tex/latex/base` below the 'TEXMF' root directory. Similarly, all of the following are found without problems thanks to their unambiguous suffix.

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

The latter is a BIB_TE_X 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 on our system (since we use the Type 1 versions included in T_EX Live).

```
> kpsewhich ecrm1000.pk
  /usr/local/texmf/fonts/pk/ljfour/jknappen/ec/ecrm1000.600pk
```

For the extended Computer Modern files we had to generate `.pk` files, and since the default METAFONT mode on our installation is `ljfour` with a base resolution of 600dpi (dots per inch), this instantiation is returned.

```
> kpsewhich -dpi=300 ecrm1000.pk
```

In this case, when specifying that we are interested in a resolution of 300dpi (`-dpi=300`) we see that no such font is available on the system. In fact, a program like `dvips` or `xdvi` would go off and actually build the `.pk` files at the required resolution 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 T_EX support, before turning our attention to the generic configuration file (`config.ps`) and the PostScript font map `psfonts.map`. 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/config/config.ps
> kpsewhich psfonts.map
  /usr/local/texmf/dvips/base/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/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 --format="dvips config" utm.map
/usr/local/texmf/dvips/psnfss/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/fonts/type1/urw/utm/utmr8a.pfb
```

It should be evident from these few 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 the `ls-R` database, 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 value to use.

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}
```

This little file only uses the font `cmr10`, so let us look 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 7.

`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.
```



```

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 7: 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 8: 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 9: Finding the font file

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` (compare the generic value and the huge one instantiated by `hugetex`, etc.).

`extra_mem_bot` Extra space for “large” \TeX data structures: boxes, glue, breakpoints, etc. Especially useful if you use $\text{P}\text{C}\text{T}\text{E}\text{X}$.

`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. You can see that both the `hugetex` and `pdflatex` program invocations ask for an extra 15,000 control sequences (the default value of `hash_extra` is zero).

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

9 Building on a new Unix platform

If you have a platform for which we have not created executables, you will need to compile \TeX and friends from scratch. This is not as hard as it sounds. What you need is all in the directory `source` in the distribution.

9.1 Prerequisites

You will need at least 100 megabytes of disk space to compile all of \TeX and its support programs. You’ll also need an ANSI C compiler, a make utility, a lexical scanner, and a parser generator. We recommend the GNU version of these programs (`gcc`, GNU `make`, `m4`, `flex`, `bison`). You may be able to work with other C compilers and make programs, but you will need a good understanding of building Unix programs to sort out problems.

Also, the command `uname` must return a sensible value.

9.2 Configuration

To begin, perform a normal installation of T_EX Live to your disk (see section 3.2 on p. 8). You may wish to skip installing any of the prebuilt binaries.

Then, unpack the source from the compressed tar file in the directory `source` to your disk and change directory to where you placed it.

Next, run `configure` with a command line like this:

```
> sh configure -prefix=/usr/local/TeX
```

The `-prefix` directory is the one where you installed the support tree; the directory layout will be as follows (where `$TEXDIR` stands for the directory you chose):

```
$TEXDIR/share/texmf    main tree with fonts,
                      macros, etc
```

```
$TEXDIR/man           Unix manual pages
```

```
$TEXDIR/info          GNU style Info manuals
```

```
$TEXDIR/bin/$PLATFORM binaries
```

If you want to leave out the `$PLATFORM` directory level, i.e., put the binaries directly into `$TEXDIR/bin`, specify the `--disable-multiplatform` option to `configure`.

Have a look at the output of `./configure --help` for more options you can use. For example, you can skip building of Ω and ϵ -T_EX.

9.3 Running make

Make sure the shell variable or option `noclobber` is not set. Then, run the main `make` like this:

```
> make world
```

and relax...

Alternatively, you want to log all the output, as in:

```
> sh -c "make world >world.log 2>&1" &
```

Before you believe that everything went ok, please check the log file for errors: GNU `make` always uses the string `***` whenever a command fails. Also, check if all the programs were built:

```
> cd $TEXDIR/bin/archname
```

```
> ls | wc
```

The result should be over 200 (you can check the exact number with the `bin` directory contents in the distribution).

If you need special privileges for `make install`, you can separate the `'make world'` into two different runs, like this:

```
> make all
```

```
> su
```

```
> make install strip
```

After you've installed your new binaries, you should follow the normal post-installation procedures, given in section 4 on p. 12.

10 Last words

This edition of T_EX Live is edited by Sebastian Rahtz, with the major contributors being Fabrice Popineau, who has worked away unceasingly at the Windows part of the package (especially the setup!) and contributed in many different ways with ideas, advice and code, and Staszek Wawrykiewicz, who provided great testing feedback, and co-ordinated the Polish contributions.

Kaja Christiansen performed a vital role in endless recompilations on assorted Unix platforms. Vladimir Volovich did great work on cleaning the source and making other improvements, while Gerben Wierda did all the work for Mac OS X.

10.1 Acknowledgements

T_EX Live is a joint effort by virtually all T_EX Users Groups. In particular, we are very grateful for past and present help from:

- The German-speaking T_EX Users (DANTE e.V.), who provide the hardware the T_EX Live source repository; and Rainer Schöpf and Reinhard Zierke who look after it. DANTE president Volker Schaa coordinated production with the Lehmann's Bookstore (<http://www.lob.de>).
- The Perforce corporation (<http://www.perforce.com>), for providing a free copy of their excellent change management system, which we use to manage.
- Karl Berry, who provided the original Web2C distribution, and has continued to give invaluable advice, encouragement, and help.
- Mimi Burbank, who arranged access at the Florida State University School of Computational Science and Information Technology to a slew of different computers to compile T_EX on, and acted as an essential guinea-pig whenever asked.
- Thomas Esser, without whose marvelous teT_EX package T_EX 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 T_EX4ht was used to create the HTML version of this documentation, and who worked tirelessly to improve it at short notice.
- Petr Olsak, who coordinated and checked all the Czech and Slovak material very carefully.
- Olaf Weber, for his patient assembly and maintenance of Web2c.
- Graham Williams, on whose work the catalogue of packages depends.

In addition, Gerhard Wilhelms, Volker Schaa, Fabrice Popineau, Janka Chlebíková, Staszek Wawrykiewicz, Erik Frambach, and Ulrik Vieth kindly translated documentation at various times into their respective languages, checked other documentation, and provided very welcome feedback.

10.2 Release history

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

working group's deliberations. It was also clear that the success of the 4All \TeX CD showed that Unix users would benefit from a similarly easy system, and this is the other main strand of \TeX Live.

We first undertook to make a new Unix-based TDS CD in the autumn of 1995, and quickly identified Thomas Esser's $\text{te}\TeX$ as the ideal setup, as it already had multi-platform support and was built with portability across file systems in mind. Thomas agreed to help, and work began seriously at the start of 1996. The first edition was released in May 1996. At the start of 1997, Karl Berry completed a major new release of Web2c, which included nearly all the features which Thomas Esser had added in $\text{te}\TeX$, and we decided to base the 2nd edition of the CD on the standard Web2C, with the addition of $\text{te}\TeX$'s `texconfig` script. The 3rd edition of the CD was based on a major revision of Web2C, 7.2, by Olaf Weber; at the same time, a new revision of $\text{te}\TeX$ was being made, and \TeX Live included almost all of its features. The 4th edition followed the same pattern, using a new version of $\text{te}\TeX$, 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 $\text{te}\TeX$, to correct the drift apart in versions 5 and 6.

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 on p. 4). In addition:

- At the request of the $\text{L}\mathcal{\text{A}}\TeX$ 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 8-bit input characters 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.

10.3 Future releases

TeX Live is not perfect! (And never will be.) We intend to continue to release new versions yearly, 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:

Sebastian Rahtz / 7 Stratfield Road / Oxford OX2 7BG / UK
tex-live@tug.org
<http://tug.org/texlive>

Happy TeX'ing!