The TEX Live Guide, 7th edition

Sebastian Rahtz
sebastian.rahtz@oucs.ox.ac.uk

May 2002

Contents

List of Tables

1 Introduction

This documentation describes the main features of the **TEX Live** 7 CD-ROM—a TEX/LATEX distribution for Unix, Linux, MacOSX, and Windows32 systems that includes TEX, LATEX 2ε, METAFONT, MetaPost, Makeindex, and BIBTEX; and a wide-ranging set of macros, fonts and documentation conforming to the *TEX Directory Standard* (TDS)—which can be used with nearly every TEX setup.

This TEX package uses the Web2c version 7.3.7 implementation of the programs, which tries to make TEXing as easy as possible, and takes full advantage of the efficient and highly customizable Kpathsea library from Karl Berry and Olaf Weber. It can be run either directly from the CD-ROM or installed on a hard disk.

Most of the runnable systems on the CD-ROM include a large set of drivers and support programs for TEX, including dvips (PostScript driver), dvipdfm (dvi to PDF), xdvi (X Windows previewer), dvilj (HP LaserJet driver), lacheck (LATEX syntax checker), tex4ht (TEX to HTML converter), dviconcat and dviselect, dv2dt and dt2dv (`dvi` to ASCII and vice versa), and Angus Duggan's PostScript utilities.

1.1 Extensions to TEX

The **TEX Live** runnable systems contain three extended versions of standard TEX:

1. ε-TeX, which adds a small but powerful set of new primitives, and the TeX--XeT extensions for left to right typesetting; in default mode, ε-TeX is 100% compatible with ordinary TeX. See `texmf/doc/etex/base/etex_man.pdf` on the CD-ROM for details.
2. pdfTeX, which can optionally write Acrobat PDF format instead of DVI. You will find the user manual in `texmf/doc/pdftex/pdftex-l.pdf`. The file `texmf/doc/pdftex/samplepdf/samplepdf.tex` shows how it is used. The LaTeX hyperref package has an option '`pdftex`', which turns on all the program features.
3. Ω (Omega), which works internally with 16-bit characters, using Unicode; this allows it to directly work with almost all the world's scripts simultaneously. It also supports dynamically loaded 'Ω Translation Processes' (OTPs), which allow the user to define complex transformations to be performed on arbitrary streams of input. See `texmf/doc/omega/base/doc-1.8.tex` for some (not necessarily up to date) documentation.

ε-TeX (version 2.1) is stable, although subsequent releases will add new functionality. pdfTeX (version 1.00b) is also stable, but is still being improved. Ω (version 1.23) is under development; the version on this CD-ROM is that current as of May 2002.

2 Structure and contents of the CD-ROM

The important CD-ROM top-level directories are listed below.

bin The TeX family programs, arranged in separate platform directories.

Books Examples related to some books about TeX.

FAQ Frequently Asked Questions, in English, French, and German.

MacOSX Support programs for MacOSX users

info Documentation in GNU 'info' format for the TeX system.

man Documentation in the form of Unix man pages for the TeX system.

setupw32 directory contains material for installation and use under Windows (see section 4 on p. 13).

source The source of all programs, including the main Web2c TeX and METAFONT distributions. These are stored in a bzip2-compressed tar archive.

support Various bits of TeX-related software which are *not* installed by default, support programs, and a complete distribution of Ghostscript version 7.05. You can find here some other programs (editors, TeX shells), which are usually absent from Windows installations, dedicated for beginners. They can be installed with the `TeXSetup.exe` Windows installation program.

texmf The main support tree of macros, fonts and documentation;

usergrps Material about TeX User Groups.

There are also two installation scripts for Unix systems, `install-cd.sh` and `install-pkg.sh`; we discuss them on in section 3 on p. 6.

2.1 Packages and collections

The **TEX Live** `texmf` tree consists of various 'collections', each of which refers to a set of 'packages', of which there are over 700 on the CD-ROM. Normal installation allows the user to copy all of a collection to a local hard disk from the CD-ROM, but it is also possible to install just one package of a collection.

The collections add functionality to a TEX system. One of them, named 'tex-basic', is necessary for almost all TEX tasks, and two others, called 'tex-latex' and 'tex-pdftex' are highly recommended for most users. All others are optional. The collections (defined by XML files in `texmf/tpm/collections`) and their short descriptions are listed below.

tex-basic These files are regarded as basic for any TEX system, covering plain TEX macros, Computer Modern fonts, and configuration for common drivers.

tex-bibtexextra Additional, extensive libraries of BibTeX styles and bibliographies.

tex-chemistry Essential chemistry

tex-context Hans Hagen's powerful macro package, ConText

tex-documentation Assorted useful documentation and guides

tex-etex Support files for an extended TEX

tex-extrabin Various useful, but non-essential, support programs. Includes programs and macros for texinfo system; programs for dvi file manipulation, etc.

tex-fontbin Programs for conversion between font formats, testing fonts (virtual fonts stuff, .gf and .pk manipulation, mft, fontinst, etc.)

tex-fontsextra All sorts of extra fonts

tex-formatsextra A collection of TEX 'formats', ie large-scale macro packages designed to be dumped into .fmt file

tex-games Setups for typesetting various board games, including chess

tex-genericextra This is a mixed bag of macro packages and fonts which do not seem to belong elsewhere

tex-htmlxml Packages to convert LATEX to XML/HTML, and typeset XML/SGML

tex-langafrican Support for some African scripts

tex-langarmenian Essential armenian

tex-langcjk Essential CJK (Chinese, Japanese, Korean) macros and fonts

tex-langcroatian Essential croatian

tex-langcyrillic Fonts and macro packages to typeset Cyrillic texts.

tex-langczechslovak Pick this if you want Czech/Slovak fonts and other packages

tex-langdanish Essential Danish

tex-langdutch Essential Dutch

tex-langfinnish Essential Finnish

tex-langfrench Essential French

tex-langgerman Essential German

tex-langgreek Essential Greek

tex-langhungarian Essential Hungarian

tex-langindic Essential Indic

tex-langitalian Essential Italian

tex-langlatin Essential Latin

tex-langmanju Essential Manju

tex-langmongolian Essential mongolian

tex-langnorwegian Essential Norwegian

tex-langother Other languages

tex-langpolish Pick this if you want Polish fonts and other packages

tex-langportuguese Essential Portuguese

tex-langspanish Essential Spanish

tex-langswedish Essential Swedish

tex-langtibetan Fonts and support for typesetting Tibetan

tex-langukenglish Essential UK English

tex-langvietnamese Essential Vietnamese

tex-latex These packages are either mandated by the core LaTeX team, or very commonly recommended

tex-latexextra A large collection of add-on packages for LaTeX

tex-mathextra Extra math

tex-metapost MetaPost (and MetaFont) drawing packages

tex-music Music typesetting packages

tex-omega Omega, a 16-bit extended TeX by John Plaice and Yannis Haralambous

tex-pdftex Support files for Han The Thanh's variant of TeX which can generate PDF output

tex-pictures Essential graphics

tex-plainextra Plain TEX extra macros

tex-psfonts Essential psfonts

tex-psutils Utilities to manipulate PostScript files

tex-publishers Essential publishers

tex-t1utils Utilities to manipulate Type1 fonts

tex-texbooks Examples and other material from various books about TeX/LaTeX.

tex-theses Macro packages from various Universities for their thesis styles

tex-ttfutils Utilities to manipulate TrueType fonts

win32-support You can choose individual tools from this collection. There are many TEX oriented editors, graphics files toolsets, etc.

The directory `texmf/tpm/packages` contains lists of all files in each package (used by the installation programs).

3 Installation and use under Unix

You can use the **TEX Live** CD-ROM in three ways:

1. You can mount the CD-ROM on your file system, run the `install-cd.sh` script, and select the option `<R>` (‘do not install files, set up to run off CD-ROM’), and run everything off the CD-ROM; this takes very little disk space, and gives you immediate access to everything on the CD-ROM; although the performance will not be optimal, it is perfectly acceptable on, for instance, PCs running Linux. You could also copy the entire CD contents to your hard disk and work in this way.

2. You can install all or part of the system to your local hard disk; this is the best method for many people, if they have enough disk space to spare (a minimum of about 100 megabytes, or 300 megabytes for a recommended good-sized system).

3. You can install selected packages to work either with your existing TEX system or a **TEX Live** system you installed earlier.

Each of these methods is described in more detail in the following sections.

Warning: This CD-ROM is in ISO 9660 (High Sierra) format, with Rock Ridge and Joliet extensions. In order to take full advantage of the CD-ROM on a Unix system, your system needs to be able to use the Rock Ridge extensions. Please consult the documentation for your mount command to see if it is possible. If you have several different machines on a local network, see if you can mount the CD-ROM on one which *does* support Rock Ridge, and use this with the others.

Linux, FreeBSD, Sun, SGI and DEC Alpha systems should be able to use the CD-ROM with no problems. We would appreciate receiving detailed advice from other system users who also succeed, for future versions of this documentation.

The discussion below about installation assumes you have been able to mount the CD-ROM with full Rock Ridge compatibility.

3.1 Pre-installation procedure for MacOSX users

If you do not run MacOSX, you should skip this section.

The `install-cd.sh` script is a *sh* script (begins with "`#!/bin/sh`"), but on MacOSX sh is unable to run it because sh is emulated. However, bash will run it. Unfortunately (again) bash is not installed by default on MacOSX[1].

1. (optional) See if `bash` is already installed. Launch `Terminal` (`/Applications/utilities/Terminal`) and type in a window

   ```
   >> rehash; which bash
   ```

   the answer will be:

   - the bash location (e.g. `/bin/bash` or `/usr/local/bin/bash`) if installed;
   - `bash: command not found` if not installed.

   If bash is already installed, skip to 4

2. bash installation:

   Mac friendly procedure Look in the `MacOSX` directory of the CD-ROM for an image disk named `bash.dmg`. mount that file by double-clicking it. 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.

   Terminal procedure (a) Log in as an admin user, at least a user with Admin privileges or sudo user or as the System Administrator.

   (b) Open the `MacOSX` directory on the CD-ROM and copy `bash.tar.gz` to your home directory (`~/`)

   (c) Then launch `Terminal` and type or copy/paste the line below in a Terminal window):

[1] there are some suggestions this will change in the future and even that bash will be used to emulate sh, in which case it might be true that in future versions of MacOSX, the script will just work.

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

type a carriage-return: you will be asked for your password, and `bash` will be installed.

(d) Quit `Terminal`.

3. Now after using either install method, goto 1: you must obtain `/usr/local/bin/bash`... (if not, try to log out and in).

4. The installation procedure is the same on MacOSX than on other UNIX platforms (as MacOSX is UNIX, that's quite normal!). Nevertheless, you should read what follows:

   - Note that in all commands of the following sections, `sh` must be replaced by `sudo bash`:

     ```
     >> sh install-cd.sh
     ```

     becomes

     ```
     >> sudo bash install-cd.sh
     ```

     etc.

   - On MacOSX, cd are auto-mounted: you *don't* need to use `mount`. The CD-ROM will be mounted in the `/Volumes/` directory: in order to make it the current directory, you just have to type in `Terminal`:

     ```
     >> cd /Volumes/TeXLive-7...
     ```

     (complete this line with the real name of the CD-ROM: using "auto completion" by pressing the `<tab>` key will do it).

3.2 Running TEX Live from the CD-ROM

The organisation of Web2c means that you can run programs simply by adding the appropriate directory under `bin` on the CD-ROM to your `PATH`, and the support files will all be found with no further ado. The following shows the list of available systems and the corresponding directories. **Only x86 Linux, Mac OSX, and Windows are available on the default CD. You need to ask for the Unix CD if you need the other systems.**

Compaq Alpha Linux	`alpha-linux`	CD2
Compaq Alphaev5 OSF 4.0d	`alphaev5-osf4.0d`	CD2
HP9000 HPUX 10.20	`hppa2.0-hpux10.20`	CD2
IBM RS 6000 AIX 4.2.*	`rs6000-aix4.2.1.0`	CD2
Intel x86 Solaris 2.8	`i386-solaris2.8`	CD2
Intel x86 with GNU/Linux	`i386-linux`	CD1
Mac OSX	`powerpc-darwin5.3`	CD1
Sun Sparc Solaris 2.7	`sparc-solaris2.7`	CD2
Windows 9X/ME/NT/2K/XP	`win32`	CD1

You may worry that when you subsequently make fonts or change configuration, things will go wrong because you cannot change files on the CD-ROM. However, you can maintain a parallel, writeable, TEX tree on your hard disk; this is searched before the main tree on the CD-ROM. The default location is `texmf-var` on the CD (which does not exist!), so you *must* override this by setting the `VARTEXMF` environment variable.

Thus *sh* or *bash* users on an Intel PC running Linux can mount the **TeX Live** CD-ROM on `/mnt/cdrom` by issuing the command:

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

Then they should change the current directory to `/mnt/cdrom`, run

```
>> sh install-cd.sh
```

and select the option `<R>` (*do not install files, set up to run off CD-ROM*). After that, they should include the directory containing the binaries for the given architecture into the search path by updating the `PATH` variable.

```
PATH=/mnt/cdrom/bin/i386-linux:$PATH
export PATH
VARTEXMF=/usr/TeX/texmf-var
export VARTEXMF
```

MacOSX users On MacOSX, the default shell is `tcsh`:

```
    setenv PATH /Volumes/<cd-name>/bin/powerpc-darwin5.3:${PATH}
    setenv VARTEXMF /usr/TeX/texmf-var
```

For convenience, these statements can also be entered into the `.profile` script. (for `tcsh` on MacOSX, `~/Library/init/tcsh/rc.mine`).

If in doubt, ask your local system support guru to help you work out how to mount your CD-ROM or which directory to use for your system.

Appropriate support files will be installed on your hard disk the first time you need them. You can edit and change local configuration files which are stored to the directory designated by `$VARTEXMF`. Any format file that is needed will be generated and stored here.

3.3 Installing TeX Live to a hard disk

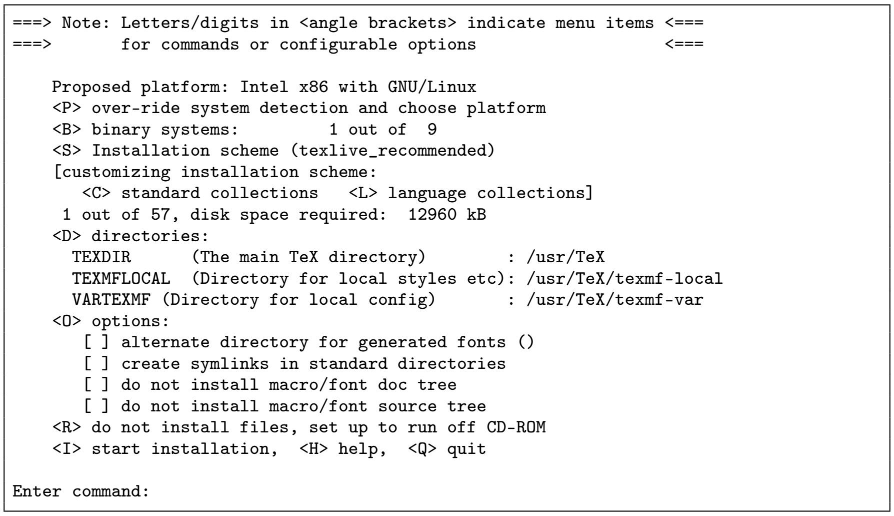
All of the necessary steps to install all or part of the distribution on your hard disk are achieved by mounting the CD-ROM, changing to the top-level directory, and typing:

```
>> sh install-cd.sh
```

(On some Unix systems, you may need to use sh5 or bash.) This script works by accessing lists of collections and packages from the CD-ROM, and trying to guess what sort of computer system you are on. It should start by displaying the following:

```
Initializing collections... Done initializing.
Counting selected collections... Done counting.
Calculating disk space requirements for collections...Done calculating that.
Initializing system packages... Done initializing system.
```

It will then show the main control screen (Figure 1), which lets you change five things:



```
===> Note: Letters/digits in <angle brackets> indicate menu items <===
===>       for commands or configurable options                  <===

    Proposed platform: Intel x86 with GNU/Linux
    <P> over-ride system detection and choose platform
    <B> binary systems:         1 out of  9
    <S> Installation scheme (texlive_recommended)
    [customizing installation scheme:
       <C> standard collections   <L> language collections]
     1 out of 57, disk space required:  12960 kB
    <D> directories:
      TEXDIR      (The main TeX directory)        : /usr/TeX
      TEXMFLOCAL  (Directory for local styles etc): /usr/TeX/texmf-local
      VARTEXMF (Directory for local config)       : /usr/TeX/texmf-var
    <O> options:
       [ ] alternate directory for generated fonts ()
       [ ] create symlinks in standard directories
       [ ] do not install macro/font doc tree
       [ ] do not install macro/font source tree
    <R> do not install files, set up to run off CD-ROM
    <I> start installation,  <H> help,  <Q> quit

Enter command:
```

Figure 1: Main control screen

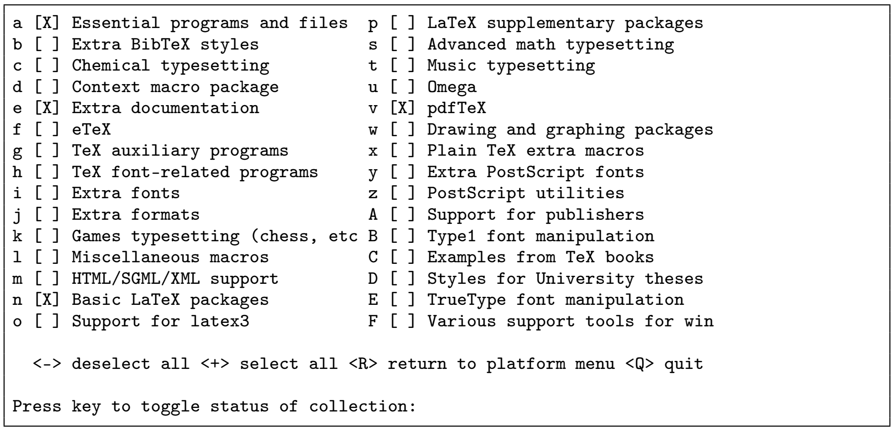
1. the type of system you are on, or want to install for;
2. the *installation scheme* you want to use (eg full, recommended, basic etc)
3. the collections you want to change from the installation scheme (they are organised into two sets: *standard collections* and *language collections*);
4. the location on your hard disk to put the files;
5. some runtime behaviour features.

You choose options by typing a letter or number and pressing ‘return’. In the example, a Linux GNU/Linux system has been detected, the default set of collections will be installed, and the default installation directory is /usr/TeX; note that the disk space required for the current installation configuration is also displayed. If you make a suggested setup, you need about 60 megabytes of disk free; however, the basic setup will only take about 30 megabytes, and you can enhance it with selected packages as you need them.

MacOSX users Most frontends (*TeXShop*, *ITeXMac*. . .) use the teTeX default location which is /usr/local/teTeX, so, Mac users could find interest in installing **TeX Live** in /usr/local/teTeX rather than in /usr/TeX.

Under the directory you choose for installation, the installation script will put the binaries in a subdirectory of bin, and the support tree in texmf. An additional tree texmf-var will contain copies of configuration files (except the main texmf.cnf), which are to be modified by texconfig program. This tree will also store generated format files for TeX, METAFONT, etc.

When you choose <C> for *standard collections*, you will see the display of available collections (Figure 2). Each collection — TeX macro files, Metafont font families, and so on — consists of several

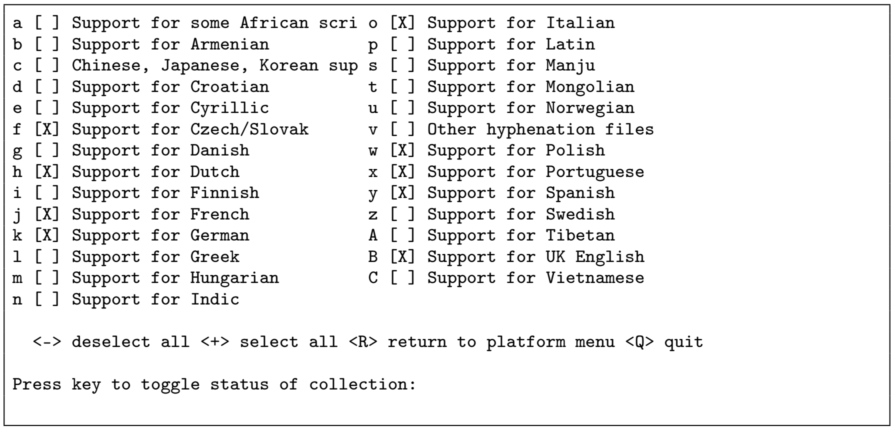


```
a [X] Essential programs and files  p [ ] LaTeX supplementary packages
b [ ] Extra BibTeX styles           s [ ] Advanced math typesetting
c [ ] Chemical typesetting          t [ ] Music typesetting
d [ ] Context macro package         u [ ] Omega
e [X] Extra documentation           v [X] pdfTeX
f [ ] eTeX                          w [ ] Drawing and graphing packages
g [ ] TeX auxiliary programs        x [ ] Plain TeX extra macros
h [ ] TeX font-related programs     y [ ] Extra PostScript fonts
i [ ] Extra fonts                   z [ ] PostScript utilities
j [ ] Extra formats                 A [ ] Support for publishers
k [ ] Games typesetting (chess, etc B [ ] Type1 font manipulation
l [ ] Miscellaneous macros          C [ ] Examples from TeX books
m [ ] HTML/SGML/XML support         D [ ] Styles for University theses
n [X] Basic LaTeX packages          E [ ] TrueType font manipulation
o [ ] Support for latex3            F [ ] Various support tools for win

  <-> deselect all <+> select all <R> return to platform menu <Q> quit

Press key to toggle status of collection:
```

Figure 2: Selecting standard collections



```
a [ ] Support for some African scri o [X] Support for Italian
b [ ] Support for Armenian          p [ ] Support for Latin
c [ ] Chinese, Japanese, Korean sup s [ ] Support for Manju
d [ ] Support for Croatian          t [ ] Support for Mongolian
e [ ] Support for Cyrillic          u [ ] Support for Norwegian
f [X] Support for Czech/Slovak      v [ ] Other hyphenation files
g [ ] Support for Danish            w [X] Support for Polish
h [X] Support for Dutch             x [X] Support for Portuguese
i [ ] Support for Finnish           y [X] Support for Spanish
j [X] Support for French            z [ ] Support for Swedish
k [X] Support for German            A [ ] Support for Tibetan
l [ ] Support for Greek             B [X] Support for UK English
m [ ] Support for Hungarian         C [ ] Support for Vietnamese
n [ ] Support for Indic

  <-> deselect all <+> select all <R> return to platform menu <Q> quit

Press key to toggle status of collection:
```

Figure 3: Selecting language collections

packages. You can toggle their inclusion on or off by pressing the key. Note that the selection letter keys are case sensitive.

When you choose <L> for *language collections*, you will see the display of available language support collections (Figure 3). Each collection consists of several packages, which provide features like hyphenation files and fonts.

The <O> for *options* item lets you decide whether to make new fonts be created in another location (if you want the main package mounted read-only for most users), and whether to make symbolic links for the binaries, man and GNU info pages in the 'standard' locations; you'll need 'root' permissions for tasks to do this, of course.

When you are finished, return to the main screen, and ask the installation to start. It will take each of the collections and systems that you requested, consult the list of files on the CD-ROM, and build a master list of files to transfer. These will then be copied to your hard disk. If you installed a system, an initialization sequence is now run (creating format files, etc.). When this has finished, all you need do is add the correct subdirectory of `bin` in the TEX installation to your path, and start using TEX. If you want, you can move the binaries up one level, e.g. from `/usr/local/bin/alpha-osf3.2` to `/usr/local/bin`; if you do this, however, you must edit `texmf/web2c/texmf.cnf` (see Appendix 11) and change the line near the start which reads

```
  TEXMFMAIN = $SELFAUTOPARENT
```

to

```
  TEXMFMAIN = $SELFAUTODIR
```

If you move the whole installation to another directory tree entirely, you need to edit `TEXMFMAIN` to specify the support tree explicitly, and set `TEXMFCNF` in your environment to `$TEXMFMAIN/texmf/web2c`.

3.4 Installing individual packages from TEX Live to a hard disk

You may want to use the **TEX Live** CD-ROM to either update an existing setup, or add features to an earlier installation from the CD-ROM. The main installation program is intended for the first time only, and subsequently you should use the `install-pkg.sh` script on the CD-ROM. Run this by mounting the CD-ROM, changing to the mounted directory, and typing

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

The script supports nine options; the first four let you set the individual package you want to install, the whole collection (i.e., `tex-mathextra`), the name of the mounted CD-ROM directory, and the name of the directory containing the list files (normally these latter two will be set automatically):

`--package=`*name*
`--collection=`*name*
`--cddir=`*name*
`--listdir=`*name*

What actually happens is controlled by four more switches; the first two allow you to exclude documentation or source files from the installation, the third stops the default action of running `mktexlsr` on completion to rebuild the file database, and the last does nothing but list the files that would be installed:

`--nodoc`
`--nosrc`
`--nohash`
`--listonly`

Finally, you can specify that, instead of installing the files, the script should make a tar archive in a specified location:

`--archive=`*name*

Thus, if we simply wanted to see the files that make up the package fancyhdr before we installed it, our command and output would be as follows:

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

texmf/doc/latex/fancyhdr/fancyhdr.dvi
texmf/doc/latex/fancyhdr/fancyhdr.tex
texmf/lists/fancyhdr
texmf/source/latex/fancyhdr/README
texmf/source/latex/fancyhdr/fancyheadings.new
texmf/tex/latex/fancyhdr/extramarks.sty
texmf/tex/latex/fancyhdr/fancyhdr.sty
texmf/tex/latex/fancyhdr/fixmarks.sty
```

Other examples of usage are:

- Install the LaTeX package `natbib`:

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

- Install the LaTeX package `alg` with no source files and no documentation:

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

- Install all the packages available in the collection containing additional Plain TeX macros:

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

- Place all files which are needed for PSTricks in a tar file in `/tmp`:

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

3.5 The texconfig program

After the installation program has copied all files to their final locations, you can use a program called texconfig that allows you to configure the system to fit your local needs. This can be called at any other time to change your setup, with a full-screen (which requires the dialog program, included as part of the binary packages) or command-line interface. It should be used for all maintenance, such as changes of installed printers, or rebuilding the file database. Both modes have help text to guide you through the facilities.

4 Installation and use under Windows

This section only applies to systems running Windows 9x, ME, NT, 2K or XP.

It is also necessary to have your Windows set up so that it uses the Microsoft Joliet extensions for reading CD-ROMs; simply look at the CD-ROM in Explorer and see whether it shows long, mixed-case, file names. If it does not, you cannot use the ready-to-run system on the CD-ROM.

This Win32 TeX systems 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`.

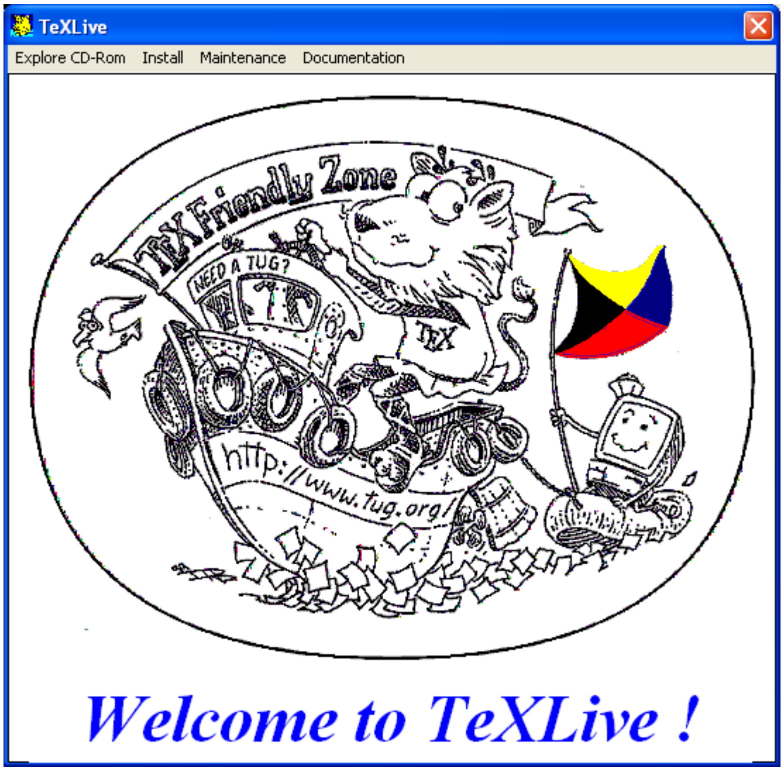


Figure 4: "Welcome to **TEX Live**" window

4.1 The `TeXLive.exe` program

If your computer is configured to let the CD-ROM 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,
- Install TeX-oriented editors on your hard disk,
- Install support packages on your hard disk (Ghostscript, NetPBM, . . .)
- Do some maintenance on you TEX system,
- Remove the TEX system,
- Use TEX off the CD-ROM,
- Cleanup the temporary files created on your hard disk when using TEX off the CD-ROM,
- Update some of the DLLs on your system,

- Browse some documentation: **TEX Live** documentation, TUG web pages, fpTEX web pages,
- Run the TeXdocTK application to find specific documentation.

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

4.2 Running from the CD-ROM

You can run all the TEX programs directly off the CD-ROM, and have access to all the macros and fonts immediately, at the price of a slower performance than if you install on the hard disk. To work effectively, one needs to modify environment variables and to create some small auxiliary directories on a hard disk. These directories will contain necessary configuration files allowing the user to modify programs settings and to generate a necessary format file. Moreover, automatically generated font files will be stored there too.

Should you want to run TEX this way, you will have to follow these steps:

1. from the menu, chose `Explore CD-Rom`, then `Select a text editor`, a dialog box will open to select some `.exe` program.

   This program needs to be a TEX oriented editor. It must be able to run the TEX compiler, previewer and any other needed tool. If you don't have one already installed on your system, you can install one from the CD-ROM, details section 4.3.

   There is no way we can guess if the program you will select is actually a text editor, so be careful. Here is a list of frequently used TEX editors:

GNU Emacs	`c:\Program Files\NTEmacs\bin\runemacs.exe`
XEmacs	`c:\Program Files\XEmacs\XEmacs-21.2\i586-pc-win32\xemacs.exe`
WinShell	`c:\Program Files\WinShell\WinShell.exe`
WinEdt	`c:\Program Files\WinEdt Team\WinEdt\WinEdt.exe`
TeXnicCenter	`c:\Program Files\TeXnicCenter\TEXCNTR.exe`

   The program selected will be memorized as the editor to use for future runs.

2. from the menu, chose `Explore CD-Rom`, then `Run TeX off CD-Rom`. The environment will be modified, a small temporary directory created and some configuration files copied there. Then, the selected editor selected will be launched, and you will be able to type in some text, let TEX typeset it and the view it or print it.

   If Ghostscript is not detected on your machine, you will be warned that rendering your DVI files might fail. You can install it from the `Install`, `Support` menu item. See section 4.3 for details.

3. you can select a different text editor any time you want.

4. if you chose `Cleanup CD-Rom setup`, everything TEX needed will be removed, comprised the selection of your text editor, but not the extra packages you may have downloaded and installed. If you installed WinShell or NTEmacs, they won't be removed.

The editor is run inside a modified environment. A temporary TDS compliant texmf tree is build in the temporary area of your computer. It is needed to store files that may be build on the fly like pk font files or format files. Configuration files are copied from the CD-ROM to this texmf tree, so that you can edit them if needed. The `ls-R` database is computed for this texmf tree. Then the `PATH` and `TEXMFCNF` environment variables are set locally, and the editor is run in this local environment. From within your editor[2], you have access to a full **TEX Live** environment, all files referenced on the CD-ROM.

[For advanced users:] You can also use the small batch file `mkloctex.bat` to be called in a directory `setupw32` of the CD-ROM. From the Start menu select 'Run', then browse 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 hard disk where you want to install the TEX directory. The whole line should read, e.g., `d:\setupw32\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.

4.3 Installing editors or support packages

You can already use the `TeXSetup.exe` program to install a single, not TEX dependent package. This might be either an editor like WinShell or NTEmacs, or also a support package like NetPBM (graphics formats conversion) or Ghostscript.

Some of the packages are not free, or not with the same meaning as for the rest of the CD-ROM. These packages have been made available through the Internet. You need to enable an Internet connection in order to install them. Chosing the `Enable Internet access` subitem will search your system for an active Internet connection, or start one if possible. *If your computer is not connected to the Internet, then the timeout might be long, 30s or more*. So try to enable it only if you know you have a connection.

Only a few packages are available from the CD-ROM, but the most important of them: NTEmacs and WinShell for the editors, Ghostscript and NetPBM for the other support packages. NetPBM is needed for running TEX4ht.

The downloadable packages are sometimes huge: Perl is 10Mb, XEmacs is 50Mb, so be warned that it can take a lot of time to install such things. `TeXSetup.exe` does not yet provide an estimation of the time needed to complete the download.

When installing these packages, `TeXSetup` is working in unattended mode. However, the programs that have their own installer like WinEdt or Ghostscript for example will require human interaction.

Those packages who have no specific installer will be unpacked and configured for your system. You will be required to select some directory where they will be installed. The directory to select should be the root directory of the whole installation. Assuming you want to install NTEmacs and NetPBM, the archive files already contain the `NTEmacs\` and `NetPBM\` part, so the directory you have to select is something like `c:\Local` or `c:\Program Files`.

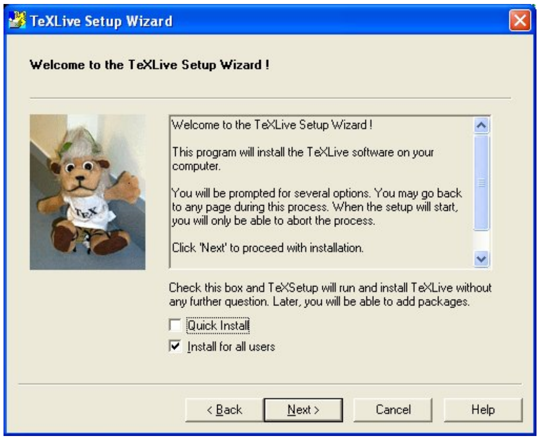
4.4 Installing to your hard disk

Installation is started by letting the CD autostart, and selecting the item `Install` from the menu, then the subitem `TeXLive`. This will invoke the `TeXSetup.exe`. You can also find it in the `bin/win32` directory and run it. `TeXSetup.exe` is a Windows wizard and it will display several pages while running.

[2] Actually, you can state any other program than a text editor, and select your command processor for example. You will then get a console with the right settings to use TEX from the CD-ROM.

Welcome Page You can choose a *quick* installation from there, in this case, the installation will run without any human assistance from beginning to end, with all the default settings (Figure 5, on the left). However, if you chose to install any support program that has its own installer like WinEdt or Ghostscript, your intervention might be required. If you have enough privileges (administrator or power user rights) under a Windows version where it is applicable, then you can decide to install the TEX **Live** for all users or for yourself only by checking the appropriate box.

The TeXSetup Wizard



TeXLive Setup Wizard

Welcome to the TeXLive Setup Wizard !

Welcome to the TeXLive Setup Wizard !

This program will install the TeXLive software on your computer.

You will be prompted for several options. You may go back to any page during this process. When the setup will start, you will only be able to abort the process.

Click 'Next' to proceed with installation.

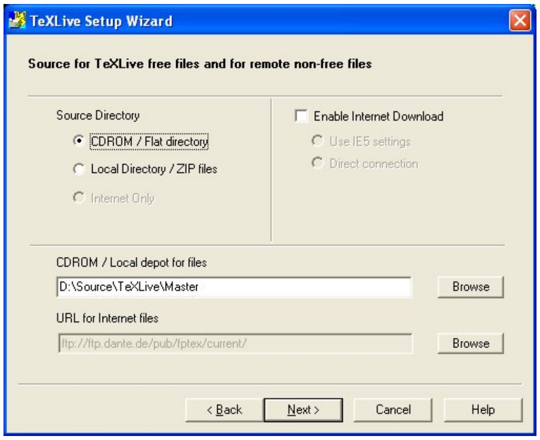
Check this box and TeXSetup will run and install TeXLive without any further question. Later, you will be able to add packages.

□ Quick Install

☑ Install for all users

< Back | Next > | Cancel | Help

Source directories for the **TEX Live** files



TeXLive Setup Wizard

Source for TeXLive free files and for remote non-free files

Source Directory

● CDROM / Flat directory

○ Local Directory / ZIP files

○ Internet Only

□ Enable Internet Download

○ Use IE5 settings

○ Direct connection

CDROM / Local depot for files

D:\Source\TeXLive\Master

Browse

URL for Internet files

ftp://ftp.dante.de/pub/fptex/current/

Browse

< Back | Next > | Cancel | Help

Figure 5: The **TEX Live** setup wizard

Source Page This page is a bit complex. It will allow you to select two source directories for your **TEX Live** system (Figure 5, on the right). You will need a *local source directory* and possibly a *remote source directory*.

Why do we need both these directories? The very files of the **TEX Live** system are on the CD-ROM, but some other packages useful under a Win32 system are not, either because of space lacking or because their license was not compatible with the **TEX Live**'s one. You need to enable Internet downloading if you want to install these support packages.

However, don't panic: the default parameters of the setup will allow you to install a full system using the CD-ROM 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:

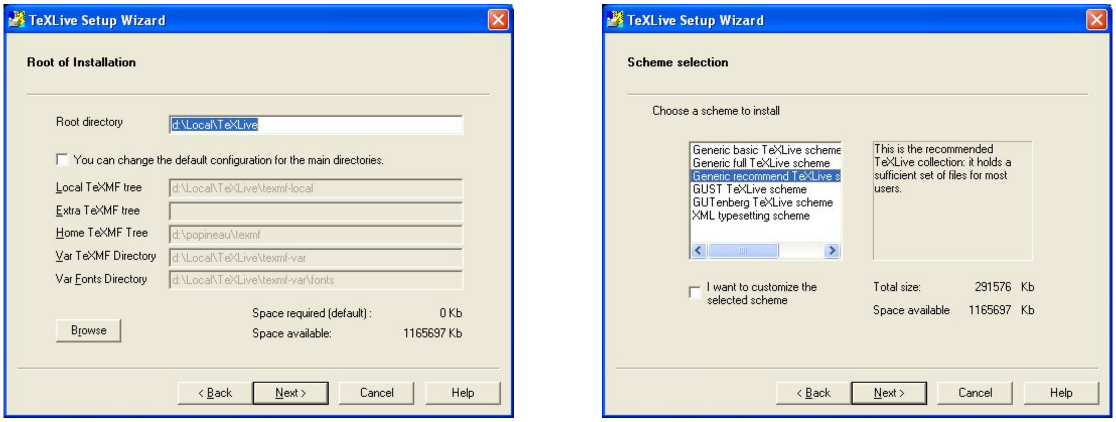
- the CD-ROM or any similar flat tree of files, available through some standard directory (this means the CD-ROM can be mounted on some remote machine and be made available through network sharing),
- a set of `.zip` files (this is the case when you are installing the fpTEX distribution),
- the Internet, 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 6, on the left). Only the root directory really matters, the other ones are set according to the root one. You may want to make `$TEXMFEXTRA` point to some TDS compliant directory with other TEX files or assign a different value to `$HOMETEXMF`, which is set by default to whatever Windows think is your 'HOME' location.

Root and directories

Scheme selection



TeXLive Setup Wizard

Root of Installation

Root directory: d:\Local\TeXLive

□ You can change the default configuration for the main directories.

Local TeXMF tree: d:\Local\TeXLive\texmf-local

Extra TeXMF tree:

Home TeXMF Tree: d:\popineau\texmf

Var TeXMF Directory: d:\Local\TeXLive\texmf-var

Var Fonts Directory: d:\Local\TeXLive\texmf-var\fonts

Space required (default) : 0 Kb

Space available: 1165697 Kb

TeXLive Setup Wizard

Scheme selection

Choose a scheme to install

- Generic basic TeXLive scheme
- Generic full TeXLive scheme
- Generic recommend TeXLive s
- GUST TeXLive scheme
- GUTenberg TeXLive scheme
- XML typesetting scheme

This is the recommended TeXLive collection: it holds a sufficient set of files for most users.

□ I want to customize the selected scheme

Total size: 291576 Kb

Space available 1165697 Kb

Figure 6: **TEX Live**-Setup: Root and directories / Schemes

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 6, on the right). A scheme is a large set of files targeted at some kind of usage. There are 3 generic schemes for basic, recommended and full installation. The other ones are devoted to LUGs (what GUST or GUTenberg propose for their members) or applications (XML and TEX). 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.

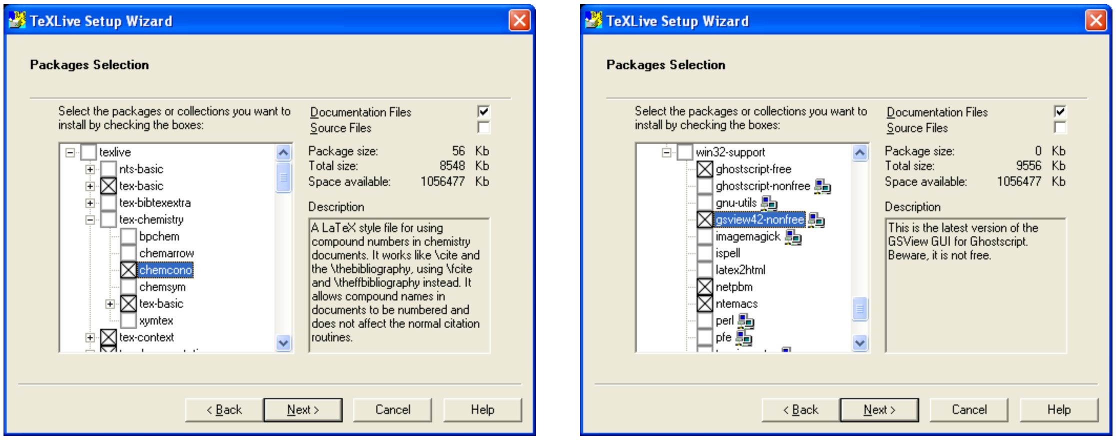
Packages Page Collections and packages are presented in a tree form (Figure 7, on the left). The links in the tree are dependency links. Collections *depend on* packages and maybe other collections, and it is the same for each package. You can select any package or collection individually, but your request 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 `win32-support` collection displayed on the picture is Win32 specific. It holds a number of bonus packages (Figure 7, on the right) which can be installed automatically and individually:

Ghostscript, the PostScript interpreter, TEX oriented editors, tools like Perl, LATEX2HTML, etc. *None of these packages are selected by default*. Some of them have an Internet Explorer icon on their right, this means that they are not on the CD-ROM and they will be available only if you previously enabled Internet downloading. *This collection cannot be selected entirely at once: you need to select the packages individually*. This is to avoid unwanted downloads of huge files.

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 8, on the left). It is still time to go back to change them.



Packages Page

TeXLive Setup Wizard

Packages Selection

Select the packages or collections you want to install by checking the boxes:

- □ texlive
  - □ nts-basic
  - ☒ tex-basic
  - □ tex-bibtexextra
  - □ tex-chemistry
    - □ bpchem
    - □ chemarrow
    - ☒ chemcono
    - □ chemsym
    - ☒ tex-basic
    - □ xymtex
  - ☒ tex-context

Documentation Files ☑
Source Files □

Package size:	56	Kb
Total size:	8548	Kb
Space available:	1056477	Kb

Description

A LaTeX style file for using compound numbers in chemistry documents. It works like \cite and the \thebibliography, using \fcite and \theffbibliography instead. It allows compound names in documents to be numbered and does not affect the normal citation routines.

Win32 Support

TeXLive Setup Wizard

Packages Selection

Select the packages or collections you want to install by checking the boxes:

- □ win32-support
  - ☒ ghostscript-free
  - □ ghostscript-nonfree
  - □ gnu-utils
  - ☒ gsview42-nonfree
  - □ imagemagick
  - □ ispell
  - □ latex2html
  - ☒ netpbm
  - ☒ ntemacs
  - □ perl
  - □ pfe

Documentation Files ☑
Source Files □

Package size:	0	Kb
Total size:	9556	Kb
Space available:	1056477	Kb

Description

This is the latest version of the GSView GUI for Ghostscript. Beware, it is not free.

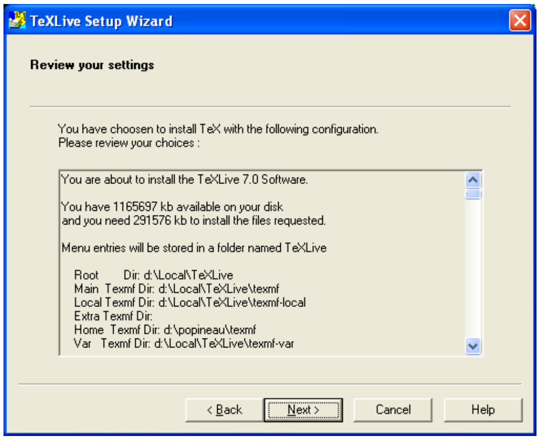
Figure 7: Packages Page / Win32 goodies

Files Copy Page The selected files are copied on your hard disk (Figure 8, 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-ROM.

Configuration Page Several packages need some configuration step to make them usable (Figure 9, on the left). Also the **TEX 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 9 on the right). If it is needed (Win9x/WinME), you will be asked to reboot your computer.

Review Page



Review your settings

You have choosen to install TeX with the following configuration.
Please review your choices :

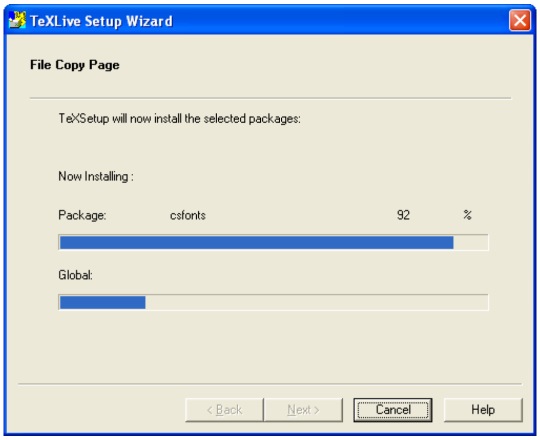
```
You are about to install the TeXLive 7.0 Software.

You have 1165697 kb available on your disk
and you need 291576 kb to install the files requested.

Menu entries will be stored in a folder named TeXLive

   Root       Dir: d:\Local\TeXLive
   Main  Texmf Dir: d:\Local\TeXLive\texmf
   Local Texmf Dir: d:\Local\TeXLive\texmf-local
   Extra Texmf Dir:
   Home  Texmf Dir: d:\popineau\texmf
   Var   Texmf Dir: d:\Local\TeXLive\texmf-var
```

File Copy Page



File Copy Page

TeXSetup will now install the selected packages:

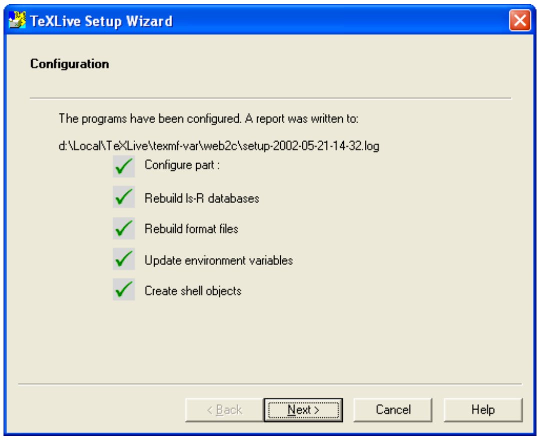
Now Installing :

Package: csfonts 92 %

Global:

Figure 8: Review Page / File Copy Page

Configuration Page



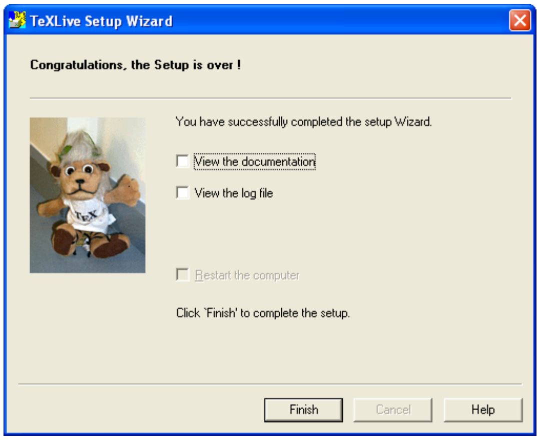
Configuration

The programs have been configured. A report was written to:

d:\Local\TeXLive\texmf-var\web2c\setup-2002-05-21-14-32.log

- ✓ Configure part :
- ✓ Rebuild ls-R databases
- ✓ Rebuild format files
- ✓ Update environment variables
- ✓ Create shell objects

Final Page



Congratulations, the Setup is over !

You have successfully completed the setup Wizard.

□ View the documentation

□ View the log file

□ Restart the computer

Click 'Finish' to complete the setup.

Figure 9: Configuration Page / Final page

Please be aware that the choice of cluster size on DOS disk partitions can radically affect the size of your TEX installation. The support tree has hundreds of small files, and it is not unusual for a complete installation to take up to 4 times the amount of space used on the CD-ROM.

5 Maintenance and other aspects of the TeX Live installation under Windows

5.1 What's different under Win32 from the standard Web2c?

The Win32 version of Web2c has some specific features that need to be noticed.

Kpathsea the hash-tables that Kpathsea builds are quite large for the **TeX Live**. In order to cut down the starting time of any Kpathsea-enabled program, these hash-tables have been put in shared memory. This way, when you chain the execution of several such programs, like `tex` calling `mpost` calling `tex`, the overhead when starting each of the programs but the first will be 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 option 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)[3]. For this reason, you can combine the `-multiple-occurences` 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 the ones from regular Web2c, and one option with a different behaviour:

- the `-fmt` option behaves differently. Previously and with the regular Web2c distribution, this option has 2 different meanings when you are in "ini" or "vir" mode. Under Win32, it has the same meaning: preload the format file specified with the argument. The meaning of building a format of such name in "ini" mode is obtained by the new `-job-name` option.
- `-job-name` option: allows to set the name of the file resulting from the compilation process. In normal mode, it will change the base name of all files produced (`.dvi`, `.aux`, etc.), while in "ini" mode, it will set the name of the format file written.
- `-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.
- `-output-directory` allow to write all the output files in the specified directory.

[3] It is noticeable that all these files result in clashes inside the Kpathsea-hashing mechanism, fortunately, Kpathsea never look for these files.

- `-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.

5.2 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 again `TeXSetup.exe`, but in maintenance mode this time. The steps you will go through are almost identical to the ones the first time you run it.

The only different step is about 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 your CD-ROM 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 ones of the 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 you want to add there the support file for Maple (symbolic computation program) for example, 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\`

Next, **do not forget to rebuild the ls-R databases files**, either by using the right menu (Start -> Programs -> TeXLive -> Maintenance), either by manually running the `mktexlsr` command.

5.3 Removing TEX Live from your hard disk

The uninstall procedure is available either from the `TeXLive.exe` program, from the `TeXLive` menu or from the control panel (Start menu -> Control Panel, Add/Remove Programs option). 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.

5.4 Running `TeXSetup.exe` from the command line

The `TeXSetup.exe` program has a number of other interesting 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;

`--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-ROM, so you will need to use this option to specify the root of the CD-ROM;

`--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-ROM, which means there can be files left if you added style files or format files, and also that supplementary tools will not be removed[4];

`--help` this option opens up a box with the list of options.

[4] This option is still a bit crude as of May 28, 2002

5.5 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 teTeX or Unix **TeX Live** installation. That means you can use Samba either to mount from an NT server to a Unix workstation or the converse. Several strategies are possible:

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

5.6 Personal Configurations

5.6.1 WinShell

Start this program from the Start menu or from the Desktop shortcut. Go to **Options** -> **Program Calls**.

- In the DVIWin tab, if the filename is **yap**, replace it with **windvi.exe**.
- In Ghostview tab, make sure it gives the correct path for gsview32.exe:
  `C:\ghostgum\gsview\gsview32.exe` (for 3.6, the current version)
  or
  `C:\gstools\gsview\gsview32.exe` (for the older version)

Click **OK**.

Note that the install process sets all files with the `.tex` extension to open with WinShell. Unless you plan to use another editor (such as WinEdt or Emacs), this is appropriate.

Unfortunately, WinShell does not have a spell-checking feature. However, if you have installed the `tex-extrabin` collection, your installation includes **Ispell** (a spell checking program found on most Unix systems). The executable is in your PATH so `ispell.exe` will be found if you invoke it from a DOS window. If you installed documentation, look at
`C:\Program Files\TeXLive\texmf\doc\html\manpages\ispell.html`
for information on Ispell. (Otherwise, you can find `ispell.html` on the CD.) If you rely on spell checking, you may want to add an Ispell icon to WinShell. See subsection 5.9.4 of this document for how to do that.

For an excellent commercial (but inexpensive) spelling checker, see `http://www.microspell.com`.

WinShell also has on-line help, which you can access via the **?** on the menu bar.

Other information about using WinShell can be found in section 5.9 on p. 27.

5.6.2 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 (WinShell will do fine) 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 type1 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
```

Note: The current **TEX Live** distribution has implemented (for the first time!) the procedure of making always up-to-date fontmaps files for Dvips and Pdftex. This is done by 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`.

5.6.3 Pdftex

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

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

Save the file and exit the editor.

5.6.4 GSView

Starting from versions compatible with Ghostscript 6.50, GSView is not free anymore, but shareware. So it is not on the CD-ROM anymore.

You may want to change the papersize to US letter size. If so, open GSView from the Start menu. From the **Media** menu, select **Letter**.

Also, there are menu settings that are supposed to give you the most readable screen image:

From **Media** -> **Display Settings**, set **Text Alpha** and **Graphics Alpha** both 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 5.8 below.

5.6.5 WinDvi

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

Open it from the Start menu (Programs -> TeXLive -> DVI Viewer). You can set it for US letter paper by going to **View** -> **Options** and next to **Papertype**, selecting US (8.5" x 11"). Click **OK**. Exit Windvi. You can change other parameters from there, like the ability to execute commands included in path`\special{}`. 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 the configuration for Windvi is stored in the `$HOME/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.

5.7 Testing

You can test WinShell by opening the file `sample2e.tex`, found in `C:\Local\TeX\texmf\tex\latex\base`. The LATEX source should appear on the screen. Process it by clicking on the LATEX icon on the toolbar, then view it by clicking on the Preview (Windvi) icon.

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. Return to WinShell and try dvips, then GSView.

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

5.8 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 5.6.5), 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. To print from GSView, first select **Print...** from the **File** menu. A Print window will appear.

If you will be using a PostScript printer, *be sure to select **PostScript Printer***. In the newer version 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 on your PC. 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.)

If you use WinShell and a PostScript printer, probably the most convenient way to print is to add an icon to the WinShell toolbar that invokes dvips in a way that sends the output directly to a default printer. For detailed instructions on how to do this, see 5.9.3 on p. 27 (*More About WinShell*).

5.9 More About WinShell

5.9.1 Installing Bug Fixes

WinShell's author (Ingo de Boer, thanks to him) sometimes releases beta versions of the next WinShell version which are also bug fixes. You can grab them from `http://www.winshell.de`. Usually they are `.zip` files that only require to be unpacked in WinShell directory (`c:\Program Files\WinShell` by default), either using WinZip or a similar tool, or by using `unzip` on the command line. If you got some `winshellbugfix.zip` file and that you saved it in the WinShell directory, then you need to run:

```
c:\>cd c:\"Program Files"\WinShell
c:\>c:\local\bin\unzip winshellbugfix.zip
```

Say 'yes' if you are asked if some files should be overwritten. The `unzip.exe` programme can be found in the `support/gnu-utils` package. If you do not have it on your machine, you can use any archiver tool like WinZip to achieve the same effect.

5.9.2 Using the Project Feature

If your document is split into several files (for example a thesis), look into WinShell's "Project" feature. From the **Project** menu, you give the project a name (e.g., Thesis), supply the name of the main (or root) file, and then "add" other files. These filenames display on the left of the screen where you can double click the names to view and switch between them. Clicking the **LaTeX** icon always processes the main file.

Note the icons on the toolbar for toggling the project space (on the left) and the log space (at the bottom). If you are not using the Project feature, you may want to toggle off the space on the left, using the full screen width to display your file.

5.9.3 Printing from WinShell to a PostScript Printer

The Dvips icon on the WinShell toolbar puts the PostScript output in a file, which you can then view with GSView and print from there if you choose. However, it's convenient to add a WinShell "program call" to dvips which sends the output directly to a designated PostScript printer. The steps below show how to do this for the printer **vclw**; you should substitute the name of your most frequently-used printer for **vclw**.

1. Make the program aware of the printer:
   - Open WinShell, go to **Options** -> **Program Calls** -> **User defined**.
   - Click on **Tool 1** in the list on the right and fill in the fields to the left as follows:
     Name: `Print`
     exe file: `dvips`
     cmd-line: `-D600 %m -o vclw`
     Uncheck the box for "DVIPS first"
   - Click **OK**
2. Add Print to the toolbar:
   - Go to **Options** -> **View** -> **Customize**.
   - In the Category box, select **User-Programs**.

- Select **Print** and drag it to the toolbar, placing it just to the right of the GSView icon.
- You then have a choice of "Image only", "Text only", or "Image and Text". The easiest is to select "Text only" and click **OK**. You should then see **Print** on the toolbar. (If you prefer, you can select "Image only", then "Edit", and edit the displayed picture to your satisfaction.)

Now, to print a LaTeX document, just click on the **Print** icon to send to your selected printer. To use a different printer, you will need to click on the **Dvips** icon to print to a file. Then click on the GSView icon and use GSView to send to any printer you have installed on your PC.

5.9.4 Adding Ispell to WinShell

1. Add Ispell to User Tools:

   - Open WinShell, go to **Options** -> **Program Calls** -> **User defined**.
   - In the list on the right, click on **Tool 1** (or **Tool 2** if you have already used **Tool 1**) and fill in the fields to the left as follows:
     Name: `Ispell`
     exe file: `ispell`
     cmd-line: `-t -d american %c.tex`
     Uncheck the boxes for "LaTeX first" and "DVIPS first"
   - Click **OK**

2. Add Ispell to the toolbar:

   - Go to **Options** -> **View** -> **Customize**.
   - In the Category box, select **User-Programs**.
   - Select **Ispell** and drag it to the toolbar, placing it just to the right of the GSView icon (or the last icon you added).
   - You then have a choice of "Image only", "Text only", or "Image and Text". The easiest is to select "Text only" and click **OK**. You should then see **Ispell** on the toolbar. (If you prefer, you can select "Image only", then "Edit", and edit the displayed picture to your satisfaction.)

Now, when you have a LaTeX document open, you can click on **Ispell** to perform spell checking. Ispell will open another window and display the first misspelled word on the left with the filename on the right. Below that you will see the context in which the misspelling appears; often several suggestions for replacements are also displayed. To replace the word, enter the number corresponding to the desired replacement. Other possible responses are listed below; for example, you can press the space bar to ignore the misspelled word. For more information on Ispell, read the manual page: `C:\Program Files\TeXLive\texmf\doc\html\manpages\ispell.html`.

Note that when you replace a word, you will not see the correction in your WinShell window until you close the file (click the X in the upper right corner) and then open it again (use the File menu).

5.10 Tips and tricks about the Win32 platform

5.10.1 Different flavors of Win32

What we call Win32 is not an operating system by itself. It is a set of functions – and a large one[5] – 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 behaviour, and loaded with high level features;
- Windows 2K, written on an NT basis, 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 written on an NT basis.

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.

5.10.2 Command line prompt

You will 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 not tied to GUI problems, and GUI tools can be designed to interface the command line tools. This is the case for TEX where you will interact with it most of the time through a GUI text editor – see section 5.6.1 for example.

However, you may need to use the command line prompt in a number of situations, by example in case of problems and you want to debug your setup – see section 5.11.

[5] Around 12000 functions in the header files of the Microsoft SDK

Win9x You will open a command line prompt by looking either for the MS-DOS icon in the "Start->Programs" menu, either by choosing "Start->Run" menu and typing in `command.com`

NT, 2K, XP You will open a command line prompt by looking for the "Command Prompt" in the "Start->Accessories" menu[6]. You can also choose the "Start->Run" menu and type in `cmd.exe`, which is the name of the brand new command interpreter for NT[7].

5.10.3 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 the `/` to introduce arguments to commands.

All this to say: do not be surprised to read path names written using the Unix convention; fpTEX is a port of Web2c, and aims to be compatible across platforms. For this reason, all the configuration files that need to specify path names use the Unix convention.

5.10.4 File systems

The worse feature of Win9x with regard to TEX is probably the so-called FAT file system. TEX uses many many small files, with size around 1kb – 3kb. The FAT file system is old, and predates by far the multi-gigabytes hard disks we have today. It means it can't manage efficiently the 30000 TEX files found on the CD-ROM. The FAT file system will allocate 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 – namely FAT32 and NTFS – do not have this drawback. They manage clusters of 4kb only[8].

5.10.5 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.

[6] These locations may change across different OS versions.

[7] Which explains why it is untrue to call this a *DOS* box under NT!

[8] You can lower the limit to 512 bytes on NTFS

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 later 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.

5.10.6 TEX engines

If you have a look at the Web2c documentation, you will read that all the various TEX derived programs use the same base engine. For example, `tex.exe` and `latex.exe` are exact copies of the same program, but each one will use a different format file, based on its calling name.

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

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

```
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.

5.11 In case of problems

5.11.1 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 file[9]. For

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

diagnostic purposes you can temporarily set an environment variable (in 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 under either W9x or NT/2K/XP, then just 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//
```

- 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.exe` and co. 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.

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

There are several questions to ask about:

1. Is `tex.exe` on 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? Errors are flagged with the sequence `Error`.

4. One can also go to `http://www.tug.org/tex-live.html` and check for any bug fix.

5. The Windows distribution on the CD-ROM is no more no less than the fpTeX distribution, so you can also go to the Web pages at `http://www.fptex.org`, or consider subscribing to the fpTeX mailing-list by consulting `http://www.tug.org/mailman/listinfo/fptex` .

The **TEX Live** software is complex and made of more than 250 programs and around 40000 files from various sources. It is quite difficult to predict all possible causes for problems. Nevertheless, we will do our best to help you in every case.

5.12 Compiling the source files

You have the whole set of source files, comprised for Windows in the `source/source.tar.bz2` archive available on the CD-ROM. To be able to compile the whole distribution for Windows, you will need:

- Windows 2K/XP
- Microsoft Visual Studio .Net
- 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
- adjust the paths in the Perl script file `win32/perl/build.pl`,
- run the compilation from the `win32/` directory using this command:

```
c:\texlive\source\win32>perl ./perl/build.pl --install --log=install.log
```

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

5.13 Where to get more information?

The Win32 TEX distribution on the CD-ROM is also known as fpTEX. Only the packaging differs, but fpTEX is no more no less than the current **TEX Live** release for Windows.

The fpTEX home on the Web is at:
`http://www.fptex.org/`

The current fpTEX release is available from any CTAN site in the directory :
`ftp://ctan.tug.org/tex-archive/systems/win32/fptex/`.

The main ftp site for fpTEX is `ftp://ftp.dante.de/pub/fptex/` from where beta versions of fpTEX and additionnal tools are available. This main site is (partially) mirrored daily by the CTAN backbones in their `systems/win32/fptex` directory.

The TEX Users Group is kindly hosting a mailing-list dedicated to fpTEX. This is a very low volume one. It is used for announcements, bugs reports or as well to discuss about improvements or various users problems. To subscribe, read the page at `http://www.tug.org/mailman/listinfo/fptex`. The mailing list address is `fptex@tug.org`.

6 Building on a new Unix platform

If you have a platform for which we have not provided binary sources, 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` on the CD-ROM.

You should first install the support tree from the **TEX Live** CD-ROM (do a basic install, with no system binaries chosen).

6.1 Prerequisites

You will need about 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. The GNU utilities (gcc, GNU make, m4, flex, bison) are the most widely tested on different platforms. gcc-2.7.* flex-2.4.7 and GNU make-3.72.1 or newer should work well. 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. The command `uname` must return a sensible value.

6.2 Configuration

First, unpack the source from the compressed `tar` file in the directory `source` to your disk and change directory to where you placed it. Decide where the 'root' of the installation will be, e.g. `/usr/local` or `/usr/local/TeX`. Obviously you should use the same location that you specified when you installed the support tree.

Now, start the build process by running configure with a command-line like

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

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

`$TEXDIR/man`	Unix manual pages
`$TEXDIR/share/texmf`	main tree with fonts, macros, etc
`$TEXDIR/info`	GNU style info manuals
`$TEXDIR/bin/$PLATFORM`	binaries

You can omit the use of '`share/`' part for the `texmf` directory if you want, as `$TEXDIR/share/texmf` and `$TEXDIR/texmf` are auto-detected by configure. If you choose something different, you have to specify that directory with the `--datadir` option of configure.

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

Have a look at the output of `./configure -help` for more options you can use (such as omitting optional packages such as Ω or ε-TeX).

6.3 Running make

Make sure the shell variable `noclobber` is not set, and then type

```
>> make world
```

and relax....

It could also be useful to log all the output, e.g. by typing

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

Before you think that everything is ok, please check the log file for errors (GNU make always uses the string "`Error:`" whenever a command returns an error code) and check if all binaries are built:

```
>> cd /usr/local/TeX/bin/i686-pc-linux-gnu
>> ls | wc
```

The result should be 209.

If you need special privileges for `make install`, you can run two make jobs in separate runs:

```
>> make all
>> su
>> make install strip
```

6.4 Final configuration steps

Set up your `PATH` to include the directory containing the just-installed binaries (e.g. `/usr/local/TeX/bin/mips-sgi-irix6.5`); similarly, `MANPATH` and `INFOPATH` to include the relevant newly installed subdirectories, i.e. `$TEXDIR/man` and `$TEXDIR/info`.

The program texconfig allows you to set the defaults for hyphenation, paper size, print command, METAFONT mode, etc. You can run this command interactively and see what options it offers, or type

```
>> texconfig help
```

For example, if you are not using A4 format paper, you can make 'lettersize' the default using:

```
>> texconfig dvips paper letter
>> texconfig xdvi paper us
```

7 A user's guide to the Web2c system

Web2c contains a set of TEX-related programs, i.e., TEX itself, METAFONT, MetaPost, BIBTEX, etc. The original implementation was by Tomas Rokicki who, in 1987, developed a first TEX-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 latest result is Web2c Version 7.3, which was released in March 1999. Our version has some updates for the forthcoming new release, and identifies itself as 7.3.7

The Web2c 7.3 system runs on Unix, Windows 3.1, 9x/ME/NT/2K/XP, DOS, and other operating systems. It uses Knuth's original sources for TEX and other basic programs written in `web` and translates them into C source code. Moreover, the system offers a large set of macros and functions developed to augment the original TEX software. The core TEX family components are:

bibtex Maintaining bibliographies.

dmp troff to MPX (MetaPost pictures).

dvicopy Produces modified copy of DVI file.

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 Property list to TFM.

pooltype Display `web` pool files.

tangle `web` to Pascal.

tex Typesetting.

tftopl TFM to property list.

vftovp Virtual font to virtual property list

vptovf Virtual property list to virtual font.

weave `web` to TEX.

The precise functions and syntax of these programs are described in the documentation of the individual packages or of Web2c itself. However, knowing a few principles governing the whole family of programs will help you to benefit optimally from your Web2c installation.

All programs honor the 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 few configuration files to optimize searching the TEX directory tree. Web2c can handle more than one directory tree simultaneously, which is useful if one wants to maintain TEX's standard distribution and local extensions in two distinct trees. To speed up file searches the root of each tree has a file `ls-R`, containing an entry showing the name and relative pathname for all files "hanging" under that root.

7.1 Kpathsea path searching

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

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

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

To check a particular path element *p*, Kpathsea first checks if a prebuilt database (see "Filename database" on page 40) applies to *p*, i.e., if the database is in a directory that is a prefix of *p*. If so, the path specification is matched against the contents of the database.

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

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

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

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

7.1.1 Path sources

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

1. A user-set environment variable, for instance, `TEXINPUTS`. Environment variables with a period and a program name appended override; e.g., if "`latex`" is the name of the program being run, then `TEXINPUTS.latex` will override `TEXINPUTS`.

2. A program-specific configuration file, for example, a line "`S /a:/b`" in dvips's `config.ps`.

3. A Kpathsea configuration file `texmf.cnf`, containing a line like "`TEXINPUTS=/c:/d`" (see below).

4. The compile-time default.

You can see each of these values for a given search path by using the debugging options (see "Debugging actions" on page 45).

7.1.2 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`.

While reading the description of the format of the file `texmf.cnf` below, please also refer to appendix 11, starting on page 51, which lists the `texmf.cnf` file on the CD-ROM.

- 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, MSDOS 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;}//
```

7.1.3 Path expansion

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

7.1.4 Default expansion

If the highest-priority search path (see “Path sources” on page 37) contains an *extra colon* (i.e., leading, trailing, or doubled), Kpathsea inserts at that point the next-highest-priority search path that is defined. If that inserted path has an extra colon, the same happens with the next highest. For example, given an environment variable setting

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

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

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

then the final value used for searching will be:

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

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

7.1.5 Brace expansion

A useful feature is brace expansion, which means that, for instance, `v{a,b}w` expands to `vaw:vbw`. Nesting is allowed. This 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-ROM, 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.

7.1.6 Subdirectory expansion

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

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

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

7.1.7 List of special characters and their meaning: a summary

The following list summarises the meaning of 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, e.g., `a{1,2}b` will become `a1b:a2b`.

`//` 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.

7.2 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 "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 adapt to DOS-like "8.3" filename conventions in source files.

7.2.1 The filename database

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

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

```
cd /your/texmf/root && ls -LAR ./ >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 for changes in the installed files—perhaps after installing or updating a LaTeX package—the file `ls-R` is automatically updated.

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

7.2.2 kpsewhich: Standalone path searching

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

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

The options specified in "*option*" can 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 1, which lists currently recognized names, a description, associated environment variables[10], and possible file extensions.

Table 1: Kpathsea file types

Name	*Description*	*Variables*	*Suffixes*
afm	Adobe font metrics	AFMFONTS	.afm
base	Metafont memory dump	MFBASES, TEXMFINI	.base
bib	BibTeX bibliography source	BIBINPUTS, TEXBIB	.bib
	bitmap fonts	GLYPHFONTS, TEXFONTS	
bst	BibTeX style files	BSTINPUTS	.bst
cnf	Runtime configuration files	TEXMFCNF	.cnf
dvips config	dvips configuration files, e.g., `config.ps` and `psfonts.map`	TEXCONFIG	.map
fmt	TeX memory dump	TEXFORMATS, TEXMFINI	.fmt, .efmt, .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

[10] You can find definitions for these environment variables in the file `texmf.cnf` (page 51)

Kpathsea file types *continued*

Name	*Description*	*Variables*	*Suffixes*
mft	MFT style file	MFTINPUTS	.mft
	miscellaneous fonts	MISCFONTS	
mp	MetaPost source	MPINPUTS	.mp
mppool	MetaPost program strings	MPPOOL, 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	OTPINPUTS	.otp
ovf	Ω virtual fonts	OVFFONTS, TEXFONTS	.ovf
ovp	Ω virtual property lists	OVPFONTS, TEXFONTS	.ovp
pk	packed bitmap fonts	*program*FONTS (*program* being XDVI, etc.), PKFONTS, TEXPKS, GLYPHFONTS, TEXFONTS	.pk
PostScript header	downloadable PostScript	TEXPSHEADERS, PSHEADERS	.pro, .enc
tex	TeX source	TEXINPUTS	.tex, .cls, .sty, .clo, .def
TeX system documentation	Documentation files for the TeX system	TEXDOCS	
TeX system sources	Source files for the TeX system	TEXSOURCES	
texpool	TeX program strings	TEXPOOL, TEXMFINI	.pool
tfm	TeX font metrics	TFMFONTS, TEXFONTS	.tfm
Troff fonts	Troff fonts, used by DMP	TRFONTS	
truetype fonts	TrueType outline fonts	TTFONTS	.ttf, .ttc
type1 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 1 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.

Note that 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 “`.`*`prognam`*” feature in configuration files. The default is “`kpsewhich`”.

`--show-path=`*`name`*

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

`--debug=`*`num`*

sets the debugging options to “*`num`*”.

7.2.3 Examples of use

Let us now have a look at Kpathsea in action.

```
>> 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” (TeX 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 BibTeX bibliography database for *TUGBoat* articles.

```
>> kpsewhich cmr10.pk
```

Font bitmap glyph files of type .pk are used by display programs like dvips and xdvi. Nothing is returned in this case since there are no pre-generated Computer Modern ".pk" files on our system (since we use the Type1 versions on the CD-ROM).

```
>> 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 600 dpi (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 TeX 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 name for these in Berry's 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 Type1 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 Regular instance `utmr8a.pfb` and find its position in the `texmf` directory tree by using a search for Type1 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.

7.2.4 Debugging actions

Sometimes it is necessary to investigate how a program resolves file references. To make this feasible in a convenient way Kpathsea offers various debug levels:

1 `stat` calls (file tests). When running with an up-to-date `ls-R` database this should almost give no output.

2 References to hash tables (like `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 you will probably always use these levels if you need any debugging.

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 Type1 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 10. 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 7.2.3 for more details about PostScript map file handling).

At this point dvips identifies itself to the user...

```
This is dvips 5.78 Copyright 1998 Radical Eye Software (www.radicaleye.com)
```

...and then 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":

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

Now the search is on for the file `cmr10.tfm`, which is found, then a few more prolog files (not shown) are referenced, and finally the Type1 instance `cmr10.pfb` of the font is located and included in the output file (see last line).

```
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 10: 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 11: 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 12: Finding the font 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]
```

7.3 Runtime options

Another of the nice features 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 listing of `texmf.cnf` is shown in Appendix 11, starting on page 51; the settings of all parameters can be found in Part 3 of that file. The more important control variables 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 PICTeX.

`font_mem_size` Number of words for font information available for TeX. This is more or less the total size of all TFM files read.

`hash_extra` Additional space for the hash table of control sequence names. Approximately 10,000 control sequences can be stored in the main hash table; if you have a large book with numerous cross-references, this might not be enough. 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 present TeX, these runtime parameters provide a practical compromise allowing some flexibility.

8 Acknowledgements

This edition of TeXLive is edited by Sebastian Rahtz, with the major contributors being Fabrice Popineau, who has worked away unceasingly at the Win32 part of the package (especially the setup!) and contributed in many different ways with ideas, advice and code; and Staszek Wawrykiewicz, who provided great checking feedback, and co-ordinated the Polish contributions. Kaja Christiansen performed a vital role in endless recompilations on assorted Unix platforms, and Robin Laakso coordinated production for

TUG. Vladimir Volovich did great work on cleaning the source and making other improvements, while Gerben Wierda did all the work for MacOSX.

We are very grateful for past and present help from

- The German-speaking TEX Users (DANTE e.V.), who provided a machine on which the master of the CD-ROM is developed and maintained; and Rainer Schöpf and Reinhard Zierke who look after it;
- The Perforce company, for providing a free copy of their excellent change management system, which we have used to manage the CD-ROM contents;
- 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 TEX on, and acted as an essential guinea-pig whenever asked;
- Kaja Christiansen, who provided essential feedback, compilation, and documentation preparation;
- Thomas Esser, without whose marvellous teTEX package this CD-ROM would certainly not exist, and whose continual help makes it a better product;
- Michel Goossens, who co-authored the documentation;
- Eitan Gurari, whose TEX4ht was used to create the HTML version of this documentation, and who worked tirelessly to improve it at short notice;
- Art Ogawa and Pat Monohon, who coordinated releases for TUG;
- Petr Olsak, who coordinated and checked all the Czech/Slovak material very carefully;
- Olaf Weber, for his patient assembly and maintenance of Web2c;
- Graham Williams, on whose work the catalogue of packages depends.

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.

9 History

This CD-ROM distribution is a joint effort by many TEX Users Groups, including those from Germany, the Netherlands, the UK, France, the Czech Republic/Slovakia, India, Poland and Russia, as well as the international TUG. Discussion began in late 1993 when the Dutch TEX Users Group was starting work on its 4AllTEX CD-ROM for MS-DOS users, and it was hoped at that time to issue a single, rational, CD-ROM for all systems. This was far too ambitious a target, but it did spawn not only the very successful 4AllTEX CD-ROM, but also the TUG Technical Council working group on a *TEX Directory Structure*, which specified how to create consistent and manageable collections of TEX support files. The final 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-ROM. The CD-ROM you now have is a very direct result of the working group's deliberations. It was also clear that the success of the 4AllTEX CD-ROM showed that Unix users would benefit from a similarly easy system, and this is the other main strand of **TEX Live**.

We undertook to make a new Unix-based TDS CD-ROM in the autumn of 1995, and quickly identified Thomas Esser's teTEX as the ideal setup, as it already had multi-platform support and was built with portability across file systems in mind. Thomas agreed to help, and work began seriously at the start of 1996. The first edition was released in May 1996. At the start of 1997, Karl Berry completed a major new release of his Web2c package, which included nearly all the features which Thomas Esser had added in teTEX, and we decided to base the 2nd edition of the CD-ROM on the standard Web2c, with the addition of teTEX's `texconfig` script. The 3rd edition of the CD-ROM was based on a major revision of Web2c, 7.2, by Olaf Weber; at the same time, a new revision of teTEX was being made, and **TEX Live** shares almost all of its features. The 4th edition followed the same pattern, using a new version of teTEX, and a new release of Web2c (7.3). The system now included a complete Windows setup.

For the 5th edition (March 2000) many parts of the CD-ROM 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 on this CD-ROM should 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 a lot 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 as a major addition a setup for MacOSX, and the usual myriad of updates to all sorts of packages and programs. An important goal was integration of the source back with teTeX, to correct the drift apart in versions 5 and 6.

10 Future versions

This CD-ROM is not a perfect product! We plan to re-issue it once a year, 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!

Corrections, suggestions and additions for future revisions should be sent to:

Sebastian Rahtz
7 Stratfield Road
Oxford OX2 7BG
United Kingdom
`rahtz@tug.org`

Updates, notes, and suggestions will be made available on CTAN in `info/texlive`. A WWW page for information and ordering details is at `http://www.tug.org/tex-live.html`.

11 The texmf.cnf file

```
% TeX Live texmf.cnf
% What follows is a super-summary of what this .cnf file can
% contain. Please read the Kpathsea manual for more information.
%
% texmf.cnf is generated from texmf.in, by replacing @var@ with the
% value of the Make variable ‘var’, via a sed file texmf.sed, generated
% (once) by kpathsea/Makefile (itself generated from kpathsea/Makefile.in
% by configure).
%
% Any identifier (sticking to A-Za-z_ for names is safest) can be assigned.
% The ‘=’ (and surrounding spaces) is optional.
% No % or @ in texmf.in, for the sake of autogeneration.
% (However, %’s and @’s can be edited into texmf.cnf or put in envvar values.)
% $foo (or ${foo}) in a value expands to the envvar or cnf value of foo.
%
% Earlier entries (in the same or another file) override later ones, and
% an environment variable foo overrides any texmf.cnf definition of foo.
%
% All definitions are read before anything is expanded, so you can use
% variables before they are defined.
%
% If a variable assignment is qualified with ‘.PROGRAM’, it is ignored
% unless the current executable (last filename component of argv[0]) is
% named PROGRAM.  This foo.PROGRAM construct is not recognized on the
% right-hand side. For environment variables, use FOO_PROGRAM.
%
% Which file formats use which paths for searches is described in the
% various programs’ and the kpathsea documentation.
%
% // means to search subdirectories (recursively).
% A leading !! means to look only in the ls-R db, never on the disk.
% A leading/trailing/doubled ; in the paths will be expanded into the
%   compile-time default. Probably not what you want.
%
% You can use brace notation, for example: /usr/local/{mytex:othertex}
% expands to /usr/local/mytex:/usr/local/othertex.  Instead of the path
% separator you can use a comma: /usr/local/{mytex,othertex} also expands
% to /usr/local/mytex:/usr/local/othertex.  However, the use of the comma
% instead of the path separator is deprecated.
%
% The text above assumes thet path separator is a colon (:).  Non-UNIX
% systems use different path separators, like the semicolon (;).

%  Part 1: Search paths and directories.

% You can set an environment variable to override TEXMF if you’re testing
% a new TeX tree, without changing anything else.
%
% You may wish to use one of the $SELFAUTO... variables here so TeX will
% find where to look dynamically.  See the manual and the definition
% below of TEXMFCNF.

% The main tree, which must be mentioned in $TEXMF, below:
TEXMFMAIN =                        $SELFAUTOPARENT/texmf
% A place for local additions to a "standard" texmf tree.
TEXMFLOCAL =                        $SELFAUTOPARENT/texmf-local

% User texmf trees can be catered for like this...
HOMETEXMF=$HOME/texmf

% A place where texconfig stores modifications (instead of the TEXMFMAIN
% tree).  texconfig relies on the name, so don’t change it.
VARTEXMF =                        $SELFAUTOPARENT/texmf-var

% Now, list all the texmf trees. If you have multiple trees,
% use shell brace notation, like this:
```

```
%   TEXMF =                        {$HOMETEXMF,!!$VARTEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN}
% The braces are necessary.
%
% A place where to store other TeX support files. It can be a remote
% texmf tree, or a tree to store non-free stuff, or ...
%   TEXMFEXTRA=$SELFAUTOPARENT/texmf-extra
% If you set this, add $TEXMFEXTRA in the list below
%
TEXMF =                        {$HOMETEXMF,!!$VARTEXMF,$TEXMFLOCAL,!!$TEXMFMAIN}

% The system trees.  These are the trees that are shared by all the users.
SYSTEXMF =                        $TEXMF

% The temporary area
TEMP =                        /var/tmp

% Where generated fonts may be written.  This tree is used when the sources
% were found in a system tree and either that tree wasn't writable, or the
% varfonts feature was enabled in MT_FEATURES in mktex.cnf.
VARTEXFONTS  =                        $VARTEXMF/fonts

% Where to look for ls-R files.  There need not be an ls-R in the
% directories in this path, but if there is one, Kpathsea will use it.
TEXMFDBS =                        $TEXMF

% It may be convenient to define TEXMF like this:
%   TEXMF =                        {$HOMETEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN,$HOME}
% which allows users to set up entire texmf trees, and tells TeX to
% look in places like ~/tex and ~/bibtex.  If you do this, define TEXMFDBS
% like this:
%   TEXMFDBS =                        $HOMETEXMF;$TEXMFLOCAL;$TEXMFMAIN;$VARTEXFONTS
% or mktexlsr will generate an ls-R file for $HOME when called, which is
% rarely desirable.  If you do this you'll want to define SYSTEXMF like
% this:
%   SYSTEXMF =                        $TEXMFLOCAL;$TEXMFMAIN
% so that fonts from a user's tree won't escape into the global trees.
%
% On some systems, there will be a system tree which contains all the font
% files that may be created as well as the formats.  For example
%   VARTEXMF =                        /var/lib/texmf
% is used on many Linux systems.  In this case, set VARTEXFONTS like this
%   VARTEXFONTS =                        $VARTEXMF/fonts
% and do not mention it in TEXMFDBS (but _do_ mention VARTEXMF).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Usually you will not need to edit any of the other variables in part 1. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% WEB2C is for Web2C specific files.  The current directory may not be
% a good place to look for them.
WEB2C =                        $TEXMF/web2c

% TEXINPUTS is for TeX input files -- i.e., anything to be found by \input
% or \openin, including .sty, .eps, etc.

% LaTeX-specific macros are stored in latex.
TEXINPUTS.latex =                        .;$TEXMF/tex/{latex,generic,}//
TEXINPUTS.hugelatex =                        .;$TEXMF/tex/{latex,generic,}//

% Fontinst needs to read afm files.
TEXINPUTS.fontinst =                        .;$TEXMF/{tex{/fontinst,},fonts/afm}//

% Plain TeX.  Have the command tex check all directories as a last
% resort, we may have plain-compatible stuff anywhere.
TEXINPUTS.tex =                        .;$TEXMF/tex/{plain,generic,}//
% other plain-based formats
TEXINPUTS.amstex =                        .;$TEXMF/tex/{amstex,plain,generic,}//
TEXINPUTS.ftex =                        .;$TEXMF/tex/{formate,plain,generic,}//
TEXINPUTS.texinfo =                        .;$TEXMF/tex/{texinfo,plain,generic,}//
TEXINPUTS.eplain  =                        .;$TEXMF/tex/{eplain,plain,generic,}//

% e-TeX.
TEXINPUTS.elatex   =                        .;$TEXMF/{etex,tex}/{latex,generic,}//
TEXINPUTS.etex     =                        .;$TEXMF/{etex,tex}/{plain,generic,}//

% PDFTeX.  This form of the input paths is borrowed from teTeX.  A certain
% variant of TDS is assumed here, unaffected by the build variables.
TEXINPUTS.pdftexinfo =                        .;$TEXMF/{pdftex,tex}/{texinfo,plain,generic,}//
TEXINPUTS.pdflatex =                        .;$TEXMF/{pdftex,tex}/{latex,generic,}//
TEXINPUTS.pdftex   =                        .;$TEXMF/{pdftex,tex}/{plain,generic,}//
TEXINPUTS.pdfelatex =                        .;$TEXMF/{pdfetex,pdftex,etex,tex}/{latex,generic,}//
TEXINPUTS.pdfetex   =                        .;$TEXMF/{pdfetex,pdftex,etex,tex}/{plain,generic,}//

% Omega.
TEXINPUTS.lambda =                        .;$TEXMF/{omega,tex}/{lambda,latex,generic,}//
TEXINPUTS.omega =                        .;$TEXMF/{omega,tex}/{plain,generic,}//

% Context macros by Hans Hagen:
TEXINPUTS.context   =                        .;$TEXMF/{pdfetex,pdftex,etex,tex}/{context,plain,generic,}//

% cstex, from Petr Olsak
TEXINPUTS.cslatex =                        .;$TEXMF//tex/{cslatex,csplain,latex,generic,}//
TEXINPUTS.csplain =                        .;$TEXMF/tex/{csplain,plain,generic,}//
TEXINPUTS.pdfcslatex =                        .;$TEXMF/{pdftex,tex}/{cslatex,csplain,latex,generic,}//
TEXINPUTS.pdfcsplain =                        .;$TEXMF/{pdftex,cstex,tex}/{csplain,plain,generic,}//

% Polish
TEXINPUTS.platex  =                        .;$TEXMF/{tex}/{platex,latex,generic,}//
TEXINPUTS.pdfplatex =                        .;$TEXMF/{pdftex,tex}/{platex,latex,generic,}//
TEXINPUTS.pdfmex   =                        .;$TEXMF/{pdftex,tex}/{mex,plain,generic,}//
TEXINPUTS.pdfemex   =                        .;$TEXMF/{pdfetex,pdftex,tex}/{mex,plain,generic,}//
TEXINPUTS.mex   =                        .;$TEXMF/tex/{mex,plain,generic,}//

% french
TEXINPUTS.frlatex =                        .;$TEXMF/{mltex,tex}/{french,latex,generic,}//
TEXINPUTS.frtex   =                        .;$TEXMF/{mltex,tex}/{french,plain,generic,}//
TEXINPUTS.frpdflatex =                        .;$TEXMF/{mltex,pdftex,tex}/{french,latex,generic,}//
TEXINPUTS.frpdftex   =                        .;$TEXMF/{mltex,pdftex,tex}/{french,plain,generic,}//

% MLTeX
TEXINPUTS.mltex =                        .;$TEXMF/{mltex,tex}/{plain,generic,}//
TEXINPUTS.mllatex =                        .;$TEXMF/{mltex,tex}/{latex,generic,}//

% odd formats  needing their own paths
TEXINPUTS.lollipop =                        .;$TEXMF/tex/{lollipop,generic,plain,}//
TEXINPUTS.lamstex  =                        .;$TEXMF/tex/{lamstex,generic,plain,}//

% David Carlisle's xmltex
TEXINPUTS.xmltex =                        .;$TEXMF/tex/{xmltex,latex,generic,}//
TEXINPUTS.pdfxmltex =                        .;$TEXMF/{pdftex,tex}/{xmltex,latex,generic,}//

% Sebastian Rahtz' jadetex for DSSSL
TEXINPUTS.pdfjadetex =                        .;$TEXMF/{pdftex,tex}/{jadetex,generic,plain,}//
TEXINPUTS.jadetex =                        .;$TEXMF/tex/{jadetex,generic,plain,}//

% Earlier entries override later ones, so put this last.
TEXINPUTS =                        .;$TEXMF/tex/{generic,}//

% Metafont, MetaPost inputs.
MFINPUTS =                        .;$TEXMF/metafont//;{$TEXMF/fonts,$VARTEXFONTS}/source//
MPINPUTS =                        .;$TEXMF/metapost//

% mft
MFTINPUTS =                        .;$TEXMF/mft//

```

```
% Web and CWeb input paths.
WEBINPUTS =                        .;$TEXMF/web//
CWEBINPUTS =                        .;$TEXMF/cweb//

% Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see web2c/INSTALL),
% and string pools (.pool) for ini{tex,mf,mp}.  It is silly that we have six
% paths and directories here (they all resolve to a single place by default),
% but historically ...
TEXFORMATS =                        .;$TEXMF/web2c
MFBASES    =                        $TEXFORMATS
MPMEMS     =                        $TEXFORMATS
TEXPOOL    =                        $TEXFORMATS
MFPOOL     =                        $TEXFORMATS
MPPOOL     =                        $TEXFORMATS

% Device-independent font metric files.
VFFONTS =                        .;$TEXMF/fonts/vf//
TFMFONTS =                        .;{$TEXMF/fonts,$VARTEXFONTS}/tfm//

% The $MAKETEX_MODE below means the drivers will not use a cx font when
% the mode is ricoh.  If no mode is explicitly specified, kpse_prog_init
% sets MAKETEX_MODE to /, so all subdirectories are searched.  See the manual.
% The modeless part guarantees that bitmaps for PostScript fonts are found.
PKFONTS =                        .;{$TEXMF/fonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modeless}//

% Similarly for the GF format, which only remains in existence because
% Metafont outputs it (and MF isn't going to change).
GFFONTS =                        .;$TEXMF/fonts/gf/$MAKETEX_MODE//

% A backup for PKFONTS and GFFONTS. Not used for anything.
GLYPHFONTS =                        .;$TEXMF/fonts

% For texfonts.map and included map files used by mktexpk.
% See ftp://ftp.tug.org/tex/fontname.tar.gz.
TEXFONTMAPS =                        .;$TEXMF/fontname

% BibTeX bibliographies and style files.
BIBINPUTS =                        .;$TEXMF/bibtex/{bib,}//
BSTINPUTS =                        .;$TEXMF/bibtex/{bst,}//

% PostScript headers, prologues (.pro), encodings (.enc) and fonts;
% this is also where pdftex finds included figures files!

TEXPSHEADERS.pdflatex   =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfelatex  =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdftexinfo =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfcslatex =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfcsplain =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfetex    =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfjadetex =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfplatex  =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfxmltex         =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdfmex     =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdftex     =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.pdftexinfo =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.cont-de    =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.cont-en    =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.cont-nl    =                        .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS.context    =                        .;$TEXMF/{etex,tex,pdftex,dvips,fonts/{type1,pfb}}//
TEXPSHEADERS =                        .;$TEXMF/{dvips,fonts/{type1,pfb},pdftex}//

% PostScript Type 1 outline fonts.
T1FONTS =                        .;$TEXMF/fonts/{type1,pfb}//;$TEXMF/fonts/misc/hbf//

% PostScript AFM metric files.
AFMFONTS =                        .;$TEXMF/fonts/afm//

% TrueType outline fonts.
```

```
TTFONTS =                        .;$TEXMF/fonts/{truetype,ttf}//
TTF2TFMINPUTS =                        .;$TEXMF/ttf2pk//

% Type 42 outline fonts.
T42FONTS =                        .;$TEXMF/fonts/type42//

% A place to puth everything that doesn't fit the other font categories.
MISCFONTS =                        .;$TEXMF/fonts/misc//

% Dvips' config.* files (this name should not start with `TEX'!).
TEXCONFIG =                        .;$TEXMF/dvips//

% Makeindex style (.ist) files.
INDEXSTYLE =                        .;$TEXMF/makeindex//;$TEXMF/tex//

% Used by DMP (ditroff-to-mpx), called by makempx -troff.
TRFONTS =                        /usr/lib/font/devpost
MPSUPPORT =                        .;$TEXMF/metapost/support

% For xdvi to find mime.types and .mailcap, if they do not exist in
% $HOME.  These are single directories, not paths.
% (But the default mime.types, at least, may well suffice.)
MIMELIBDIR =                        $SELFAUTOPARENT/etc
MAILCAPLIBDIR =                        $SELFAUTOPARENT/etc

% TeX documentation and source files, for use with kpsewhich.
TEXDOCS =                        .;$TEXMF/doc//
TEXSOURCES =                        .;$TEXMF/source//

% allow for compressed files, and various extenions
TEXDOCSSUFFIX =                        :.dvi:.pdf:.ps:.html:.txt:.tex
TEXDOCSCOMPRESS =                        :.gz:.bz2:.zip:.Z
TEXDOCEXT =                        {$TEXDOCSSUFFIX}{$TEXDOCSCOMPRESS}

% Omega-related fonts and other files.  The odd construction for OFMFONTS
% makes it behave in the face of a definition of TFMFONTS.  Unfortunately
% no default substitution would take place for TFMFONTS, so an explicit
% path is retained.
OFMFONTS =                        .;{$TEXMF/fonts,$VARTEXFONTS}/{ofm,tfm}//;$TFMFONTS
OPLFONTS =                        .;{$TEXMF/fonts,$VARTEXFONTS}/opl//
OVFFONTS =                        .;{$TEXMF/fonts,$VARTEXFONTS}/ovf//
OVPFONTS =                        .;{$TEXMF/fonts,$VARTEXFONTS}/ovp//
OTPINPUTS =                        .;$TEXMF/omega/otp//
OCPINPUTS =                        .;$TEXMF/omega/ocp//

%dvipdfm
DVIPDFMINPUTS =                        .;$TEXMF/dvipdfm//

%% t4ht utility, sharing files with TeX4ht
TEX4HTFONTSET=alias,iso8859
TEX4HTINPUTS =                        .;$TEXMF/tex4ht/base//;$TEXMF/tex4ht/ht-fonts/{$TEX4HTFONTSET}//
T4HTINPUTS=                        .;$TEXMF/tex4ht/base//
%% The mktex* scripts rely on KPSE_DOT. Do not set it in the environment.

XDVIINPUTS=.;$TEXMF/{xdvi,dvips}//
KPSE_DOT =                        .

% This definition isn't used from this .cnf file itself (that would be
% paradoxical), but the compile-time default in paths.h is built from it.
% The SELFAUTO* variables are set automatically from the location of
% argv[0], in kpse_set_progname.
%
% About the /. construction:
% 1) if the variable is undefined, we'd otherwise have an empty path
%    element in the compile-time path.  This is not meaningful.
% 2) if we used /$VARIABLE, we'd end up with // if VARIABLE is defined,
%    which would search the entire world.
%
```

```
% The TETEXDIR stuff isn't likely to be relevant unless you're using teTeX,
% but it doesn't hurt.
%
TEXMFCNF =                        .;$VARTEXMF/web2c;{$SELFAUTOLOC,$SELFAUTODIR,$SELFAUTOPARENT}\
{,{/share,}/texmf{.local,}/web2c};c:/TeX/texmf/web2c


% Suggestions for editor settings under Windows. Uncomment your
% preferred option. The corresponding MFEDIT can also be set for use with
% Metafont.
%
% Winedt:
%  TEXEDIT=C:\WinEdt\WinEdt.exe "[Open('%s');SelLine(%d,7)]
% Textpad:
%  TEXEDIT =                        c:\Progra~1\TextPad\System\Ddeopn32 TextPad %s(%d)
% UltraEdit (newer Win32 versions):
%  TEXEDIT =                        uedit32 %s/%d/1
% WinTeXShell32:
%  TEXEDIT =                        texshell.exe /l=%d %s
% vi, vim, gvim. here we show Windows gvim.exe:
%  TEXEDIT =                        gvim.exe %s +%d
% PFE:
%  TEXEDIT=pfe32/g%d  %s
% MED:
%  TEXEDIT=med.exe "%s" %d
% TSE:
%  TEXEDIT=e32.exe "%s" -n%d
% Epsilon (Lugaru)            http://www.lugaru.com/
%  TEXEDIT="c:\Program Files\eps90\bin\e32.exe" +%d %s
% WinShell
% TEXEDIT=C:\Progra~1\WinShell\WinShell.exe -c %s -l %d

% For unix
%
% vi, vim, NEdit, (X)Emacs, pico, jed
% TEXEDIT =                        vi     +%d %s
% TEXEDIT =                        vim    +%d %s
% TEXEDIT =                        nedit  +%d %s
% TEXEDIT =                        xemacs +%d %s

%(x)fte:
% TEXEDIT =                        xfte   -l%d %s


%-----------------------------------------------------------------------------
% Write .log/.dvi/etc. files here, if the current directory is unwritable.
% TEXMFOUTPUT =                        /tmp

% If a dynamic file creation fails, log the command to this file, in
% either the current directory or TEXMFOUTPUT.  Set to the
% empty string or  0  to avoid logging.
MISSFONT_LOG =                        missfont.log

% Set to a colon-separated list of words specifying warnings to suppress.
% To suppress everything, use TEX_HUSH =                        all; this is equivalent to
% TEX_HUSH =                        checksum:lostchar:readable:special
TEX_HUSH =                        none

% Enable system commands via \write18{...}?
shell_escape =                        f

% Allow TeX \openout/\openin on filenames starting with `.' (e.g., .rhosts)?
% a (any)        : any file can be opened.
% r (restricted) : disallow opening "dotfiles".
% p (paranoid)   : as 'r' and disallow going to parent directories, and
%                  restrict absolute paths to be under $TEXMFOUTPUT.
openout_any =                        p
openin_any =                        a
```

```
% Allow TeX, MF, and MP to parse the first line of an input file for
% the %&format construct.
parse_first_line =                        t

% Allow TeX, eTeX, Omega to include ‘src:’ specials in the dvi file.
% These specials are used by viewers to jump from the viewer into
% the editor at the right page/lineno.
% Possible values : none auto cr display hbox math par parend vbox
src_specials =                        none

% Disable search on multiple suffixes filenames. In many case, when ‘foo.bar’
% is looked for, you do not want to look for ‘foo.bar.tex’ before. This flag
% disables searching for standard suffixes if the file name has already an
% extension of 3 characters. Default value is true (old behaviour).
% allow_multiple_suffixes =                        f

% Enable the mktex... scripts by default?  These must be set to 0 or 1.
% Particular programs can and do override these settings, for example
% dvips's -M option.  Your first chance to specify whether the scripts
% are invoked by default is at configure time.
%
% These values are ignored if the script names are changed; e.g., if you
% set DVIPSMAKEPK to ‘foo’, what counts is the value of the environment
% variable/config value ‘FOO’, not the ‘MKTEXPK’ value.
%
% MKTEXTEX =                        0
% MKTEXPK =                        0
% MKTEXMF =                        0
% MKTEXTFM =                        0
% MKOCP =                        0
% MKOFM =                        0

% What MetaPost runs to make MPX files.  This is passed an option -troff
% if MP is in troff mode.  Set to ‘0’ to disable this feature.
MPXCOMMAND =                        makempx


%  Part 3: Array and other sizes for TeX (and Metafont and MetaPost).
%
% If you want to change some of these sizes only for a certain TeX
% variant, the usual dot notation works, e.g.,
% main_memory.hugetex =                        20000000
%
% If a change here appears to be ignored, try redumping the format file.
%

%-----------------------------------------------------------------
% Memory. Must be less than 8,000,000 total.
%
% main_memory is relevant only to initex, extra_mem_* only to non-ini.
% Thus, have to redump the .fmt file after changing main_memory; to add
% to existing fmt files, increase extra_mem_*.  (To get an idea of how
% much, try \tracingstats=2 in your TeX source file;
% web2c/tests/memtest.tex might also be interesting.)
%
% To increase space for boxes (as might be needed by, e.g., PiCTeX),
% increase extra_mem_bot.
%
%
main_memory =                        1500000 % words of inimemory available; also applies to inimf&mp
extra_mem_top =                        0    % extra high memory for chars, tokens, etc.
extra_mem_bot =                        0    % extra low memory for boxes, glue, breakpoints, etc.

% Words of font info for TeX (total size of all TFM files, approximately).
font_mem_size =                                200000

% Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
font_max =                        1000
```

```

% Extra space for the hash table of control sequences (which allows 10K
% names as distributed).
hash_extra =                        15000

% Max number of characters in all strings, including all error messages,
% help texts, font names, file names, control sequences.
% These values apply to TeX and MP.

% Minimum pool space after TeX/MP's own strings; must be at least
% 25000 less than pool_size, but doesn't need to be nearly that large.
string_vacancies =                45000
max_strings =                        65000                        % max number of strings
pool_free =                        47500                        % min pool space left after loading .fmt
pool_size  =                        125000
% Hyphenation trie. As distributed, the maximum is 65535; this should
% work unless `unsigned short' is not supported or is smaller than 16
% bits.  This value should suffice for UK English, US English, French,
% and German (for example).  To increase, you must change
% `ssup_trie_opcode' and `ssup_trie_size' in tex.ch (and rebuild TeX);
% the trie will then consume four bytes per entry, instead of two.
%
% US English, German, and Portuguese: 30000.
% German: 14000.
% US English: 10000.
%
trie_size =                        262000

% Buffer size.  TeX uses the buffer to contain input lines, but macro
% expansion works by writing material into the buffer and reparsing the
% line.  As a consequence, certain constructs require the buffer to be
% very large.  As distributed, the size is 50000; most documents can be
% handled within a tenth of this size.
buf_size =                        200000

% Parameter specific to MetaPost.
% Maximum number of knots between breakpoints of a path.
% Set to 2000 by default.
% path_size.mpost =                        10000

% These are pdftex-specific.
obj_tab_size =                        200000 % PDF objects
dest_names_size=300000 % destinations

% These are Omega-specific.
ocp_buf_size =                        500000        % character buffers for ocp filters.
ocp_stack_size =                        10000        % stacks for ocp computations.
ocp_list_size =                        1000        % control for multiple ocps.

% These work best if they are the same as the I/O buffer size, but it
% doesn't matter much.  Must be a multiple of 8.
dvi_buf_size =                        16384    % TeX
gf_buf_size =                        16384     % MF

% It's probably inadvisable to change these. At any rate, we must have:
% 45 < error_line      < 255;
% 30 < half_error_line < error_line - 15;
% 60 <=                        max_print_line;
% These apply to Metafont and MetaPost as well.
error_line =                        79
half_error_line =                50
max_print_line =                79
stack_size =                        1500 % simultaneous input sources
save_size =                        5000 % for saving values outside current group
param_size =                        1500 % simultaneous macro parameters
max_in_open =                        15   % simultaneous input files and error insertions
hyph_size =                        1000 % number of hyphenation exceptions, >610 and <32767
nest_size =                        500  % simultaneous semantic levels (e.g., groups)
```

```


%-----------------------------------------------------------------
%
main_memory.hugetex =                        1500000
param_size.hugetex =                        1500
stack_size.hugetex =                        1500
hash_extra.hugetex =                        15000
string_vacancies.hugetex =                45000
pool_free.hugetex =                        47500
nest_size.hugetex =                        500
save_size.hugetex =                        5000
pool_size.hugetex =                        1250000
max_strings.hugetex =                        65000

main_memory.mf           =                800000
main_memory.mpost =                        1000000
pool_size.mpost =                        500000

buf_size.context         =                200000 % needed for omega bug
extra_mem_bot.context    =                4000000
extra_mem_top.context    =                2000000
font_mem_size.context    =                500000
hash_extra.context       =                50000
main_memory.context      =                1500000
max_strings.context      =                100000
nest_size.context        =                   500
obj_tab_size.context     =                300000
param_size.context       =                  5000
pool_free.context        =                 47500
pool_size.context        =                1250000
save_size.context        =                 50000
stack_size.context       =                  5000
string_vacancies.context =                 90000
% Context's metafun
main_memory.metafun      =                2500000
pool_size.metafun        =                1000000

% cslatex
main_memory.cslatex =                        1500000
param_size.cslatex =                        1500
stack_size.cslatex =                        1500
hash_extra.cslatex =                        15000
string_vacancies.cslatex =                45000
pool_free.cslatex =                        47500
nest_size.cslatex =                        500
save_size.cslatex =                        5000
pool_size.cslatex =                        1250000
max_strings.cslatex =                        55000
font_mem_size.cslatex=                        400000

main_memory.lambda =                        1500000

% redundant. all LaTeX should be huge
main_memory.hugelatex =                        1500000
param_size.hugelatex =                        1500
stack_size.hugelatex =                        1500
hash_extra.hugelatex =                        15000
string_vacancies.hugelatex =                45000
pool_free.hugelatex =                        47500
nest_size.hugelatex =                       500
save_size.hugelatex =                       5000
pool_size.hugelatex =                        1250000
max_strings.hugelatex =                        55000
font_mem_size.hugelatex=                400000

% standard LaTeX is itself huge

```

```
main_memory.latex =                        1500000
param_size.latex =                        1500
stack_size.latex =                        1500
hash_extra.latex =                        15000
string_vacancies.latex =                45000
pool_free.latex =                        47500
nest_size.latex =                        500
save_size.latex =                        5000
pool_size.latex =                        1250000
max_strings.latex =                        55000
font_mem_size.latex=                        400000

main_memory.jadetex =                        1500000
param_size.jadetex =                        1500
stack_size.jadetex =                        1500
hash_extra.jadetex =                        15000
string_vacancies.jadetex =                45000
pool_free.jadetex =                        47500
nest_size.jadetex =                        500
save_size.jadetex =                        5000
pool_size.jadetex =                        1250000
max_strings.jadetex =                        55000
font_mem_size.jadetex=                        400000


main_memory.pdfjadetex =                2500000
param_size.pdfjadetex =                        1500
stack_size.pdfjadetex =                        1500
hash_extra.pdfjadetex =                        50000
string_vacancies.pdfjadetex =                55000
pool_free.pdfjadetex =                        47500
nest_size.pdfjadetex =                        500
save_size.pdfjadetex =                        5000
pool_size.pdfjadetex =                        1250000
max_strings.pdfjadetex =                55000

main_memory.xmltex =                        1500000
param_size.xmltex =                        1500
stack_size.xmltex =                        1500
hash_extra.xmltex =                        50000
string_vacancies.xmltex =                45000
pool_free.xmltex =                        47500
nest_size.xmltex =                        500
save_size.xmltex =                        50000
pool_size.xmltex =                        1250000
max_strings.xmltex =                        55000

main_memory.pdfxmltex =                        2500000
param_size.pdfxmltex =                        1500
stack_size.pdfxmltex =                        1500
hash_extra.pdfxmltex =                        50000
string_vacancies.pdfxmltex =                45000
pool_free.pdfxmltex =                        47500
nest_size.pdfxmltex =                        500
save_size.pdfxmltex =                        50000
pool_size.pdfxmltex =                        1250000
max_strings.pdfxmltex =                        55000

font_mem_size.pdflatex =                210000
main_memory.pdflatex =                        1500000
param_size.pdflatex =                       3000
stack_size.pdflatex =                       3000
hash_extra.pdflatex =                        15000
string_vacancies.pdflatex =                55000
pool_free.pdflatex =                        47500
nest_size.pdflatex =                        500
pool_size.pdflatex =                        1250000
save_size.pdflatex =                        50000
```

```
max_strings.pdflatex =                        55000

main_memory.pdfelatex =                        1500000
param_size.pdfelatex =                        1500
stack_size.pdfelatex =                        1500
hash_extra.pdfelatex =                        15000
string_vacancies.pdfelatex =                45000
pool_free.pdfelatex =                        47500
nest_size.pdfelatex =                        500
pool_size.pdfelatex =                        1250000
save_size.pdfelatex =                        50000
max_strings.pdfelatex =                        55000

main_memory.pdfetex      =                1500000 % 1000000 bot/top
hash_extra.pdfetex       =                 50000
pool_size.pdfetex        =                1250000
string_vacancies.pdfetex =                 90000
max_strings.pdfetex      =                100000
pool_free.pdfetex        =                 47500
nest_size.pdfetex        =                   500
param_size.pdfetex       =                  5000
save_size.pdfetex        =                 50000
stack_size.pdfetex       =                  5000
obj_tab_size.pdfetex     =                256000
```