Editorial

For a long period Baskerville has benefited from the sure hand of Sebastian Rahtz at the editorial helm. That period came to an end with the publication of the last number of Volume 8 and editorial control is now the collective responsibility of the Committee, with a Guest Editor for each number. It is a hard act to follow, but made easier for me as editor of this issue by having Sebastian as its main contributor. I am grateful to him for providing the article written to introduce Version 4 of the \TeX\ Live CD-ROM. This was originally written for TUGboat and is reproduced here essentially unchanged.

We are also having to adjust to being without the services of Robin Fairbairns, on whom we used to rely for the production and distribution of Baskerville. We are grateful to both Robin and Sebastian for the service that they have given over the years and we wish them well for the future. Although neither is now a member of the UK-T\TeX\G Committee, they will continue to serve the \TeX\ community: Robin as maintainer of the UK branch of the CTAN \TeX\ archive and Sebastian as Chairman of the TUG 2000 Conference to be held in Oxford next year.

The production of the \TeX\ Live series of CD-ROMs is arguably the most significant achievement of the worldwide \TeX\ community in recent years. System administrators and individual operators now have an easy means of installing and maintaining \TeX\ and its enormous circle of friends on various platforms, and in a way that minimises platform differences and brings some order to what would otherwise be a chaotic system. If you have not already used the recently distributed CD-ROM to upgrade to Version 4, then I urge you to do so. Sebastian’s article will help you do this, and you won’t regret it.

TUG 2000 Conference

The TUG annual meeting for the year 2000 will be held in Oxford, England, from Saturday, 12 August, to Friday, 18 August. The venue will be Wadham College, with both accommodation and conference facilities being provided by the college.

The conference chairman is Sebastian Rahtz (Oxford University Computing Services) and the head of local organisation is Kim Roberts (Oxford University Press).

Survey of Members

In order to better serve its members, the Committee of UK-T\TeX\G is conducting a survey of the membership. The information gained will be used to help us plan meetings and conferences, to guide us in soliciting articles for Baskerville, and to determine what other services are likely to be of most benefit to our members.

Please take the time to complete and return either the snail-mailed paper version, or the e-mailed electronic version of the questionnaire, but not both. The reward for doing this is a \TeX\ Lion lapel pin.
An introduction to \TeX{} Live 4

Introduction

This article\footnote{The guide to \texttt{kpathsea} and \texttt{web2c} was written by Michel Goossens, and Fabrice Popineau wrote the section on Windows installation and use. The \TeX{} Live CD-ROM distribution is a joint effort by the \TeX{} Users Group, and the UK, French, German, Czech/Slovak, Dutch, Indian and Polish user groups. For the 1999 edition, we are particularly grateful to:} is rewritten from that published in \textit{TUGboat} for \TeX{} Live 3 in 1998 and describes the main features of the \TeX{} Live 4 CD-ROM—a \TeX{} Live/WEB2C distribution for Unix, Linux, Windows32 (and other) systems, and a wide-ranging set of macros, fonts and documentation conforming to the \texttt{TeX} Directory Standard (TDS)—which can be used with nearly every \TeX{} setup.

The CD-ROM is provided as a benefit to members of user groups such as TUG and UK-T\textsc{e}G, and has already been distributed to members of the latter. To keep up-to-date on the \TeX{} Live project, please visit its Web page.\footnote{\url{http://www.tug.org/texlive.html}} The \TeX{} Users Group recognizes the importance of the \TeX{} Live CD-ROM and supports its development and production. Volunteers to assist with this work are encouraged to contact \texttt{tex-live@tug.org}.

A fuller version of this document (in English, French, German and Slovak) can be found on the CD-ROM in \texttt{tldoc}.

Changes since \TeX{} Live 3

Although there have been no structural changes, a very great many changes have been made, some more visible than others. Changes that users should know about include:

1. The main \TeX{} programs are based on Web2c version 7.3;
2. Both Unix and Win32 versions are identical to \texttt{t\TeX{} 0.9} (as of the end of March 1999), and simply add more programs, and a much larger support tree;
3. New programs include \texttt{dvipdfm} (DVI to PDF driver) and \texttt{tth} (\TeX{} to HTML converter), as well as new versions of \texttt{pdft\TeX{}}, \texttt{\Omega}, and \texttt{\epsilon\TeX{}};
4. The ‘December 1998’ (actually March 1999) \texttt{DST\TeX{}} is included;
5. A brand new Windows install program is provided;
6. A great many font and macro packages have been updated;

7. Packages are now starting to be classified as ‘free’ or ‘non-free’ (according to the Debian Free Software Guidelines)\footnote{\url{http://www.debian.org/intro/free}} and we expect during the coming year to complete this work, and be able to offer a genuinely ‘free’ \TeX{} CD-ROM at the start of 2000.

Naturally, much effort has been expended on testing the structural integrity of the \texttt{texmf} tree (in particular, checking it against what \texttt{texmf} does, and checking the licensing conditions of packages).

We very strongly urge any package authors reading this to consider looking at how their work is arranged on the CD-ROM, and contacting us with any problem. But most importantly, look at the \texttt{Catalogue} maintained by Graham Williams\footnote{\url{http://www.cmsis.csiro.au/Graham.Williams/TeX/catalogue.html}} and check your details in there, especially the licensing! Future versions of \TeX{} Live will rely more and more on the \texttt{Catalogue}.

Structure and contents of the CD-ROM

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

\begin{itemize}
\item \texttt{bin} The \TeX{} family of programs, arranged in separate platform directories;
\item \texttt{tldoc} Documentation for \TeX{} Live;
\item \texttt{FAQ} Frequently Asked Questions, in English, French, and German;
\item \texttt{info} Documentation in GNU ‘info’ format for the \TeX{} system;
\item \texttt{man} Documentation in the form of Unix man pages for the \TeX{} system;
\item \texttt{source} The source of all programs, including the main \texttt{web2c}, \tex{}, and METAFONT distributions—stored in a compressed tar archive;
\item \texttt{support} Various bits of \TeX{}-related software which are not installed by default, such as \texttt{music\TeX{}}, \texttt{support} programs, and a complete distribution of Ghostscript version 5.50;
\end{itemize}
You can use the \TeX Live CD-ROM in three ways:

1. You can mount the CD-ROM on your file system, adjust your \texttt{PATH}, 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.

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 10 megabytes, or 100 megabytes for a recommended good-sized system).

3. You can install selected packages to work either with your existing \TeX Live 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 \texttt{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 \texttt{does} support Rock Ridge, and use this with the others.

### Running from the CD-ROM under Unix

The organisation of Web2c means that you can run programs simply by adding the appropriate directory under \texttt{bin} on the CD-ROM to your \texttt{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.

<table>
<thead>
<tr>
<th>System</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC Alpha OSF/1</td>
<td>alpha-osf4.0</td>
</tr>
<tr>
<td>HP9000 HPUX</td>
<td>hppa11-hpux10.10</td>
</tr>
<tr>
<td>Intel Linux</td>
<td>1386-linux</td>
</tr>
<tr>
<td></td>
<td>1386-linux-libc5</td>
</tr>
<tr>
<td>SGI IRIX</td>
<td>mips-irix6.2</td>
</tr>
<tr>
<td>IBM RS 6000 AIX</td>
<td>rs6000-aix4.1.4</td>
</tr>
<tr>
<td>Sun Sparc Solaris</td>
<td>sparc-solaris2.5.1</td>
</tr>
<tr>
<td>Windows 95 or NT (Intel)</td>
<td>win32</td>
</tr>
</tbody>
</table>

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 \texttt{texmf-localconfig} on the CD-ROM (which does not exist!), so you must override this by setting the \texttt{VARTEXMF} environment variable.

Thus \texttt{sh} or \texttt{bash} users on an Intel PC running Linux can mount the \TeX Live CD-ROM on \texttt{/cdrom} by issuing the command:

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

Then they should include the directory containing the binaries for the given architecture into the search path by updating the \texttt{PATH} variable.

```bash
PATH=/cdrom/bin/i386-linux:$PATH
export PATH
export VARTEXMF
```

For convenience, these statements can also be entered into the \texttt{.profile} script.

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. It is a good idea to immediately run the \texttt{texconfig} script to initialise things, and check it all works.
Installing 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 four things:

1. the type of system you are on, or want to install for;
2. the collections you want to install, at the basic, recommended or other level;
3. the location on your hard disk to put the files;
4. some runtime behaviour features.

You choose options by typing a letter or number and pressing ‘return’. In the example, a Linux ELF system has been detected, the default of all collections to recommended level has been chosen, 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 100 megabytes of disk free; however, the basic setup will only take about 10 megabytes, and you can enhance it with selected packages as you need them.

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`.

The `options` item lets you decide whether to make new fonts be created in another location (if you want the main fonts be created in another location (if you want the main), and you can enhance it with selected packages as you need them.

When you choose `<>` for ‘collections’, you will see the display of available collections, the level of installation selected, and the disk space required (Figure 2). You can set alternative levels of installation for each collection, ranging from `none` to `all`. You can either set this for all collections at once, or choose a particular collection and set its level (Figure 3).

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 7) 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 `TEXMFCON` in your environment to `$TEXMFMAIN/texmf/web2c`.

Installing individual packages 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., `ams2`), 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=package
--collection=collection
--cddir=cddir
--listdir=listdir
```

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=archive
```

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
```

Other examples of usage are:

- Install the `I\TeX` package `natbib`:

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

- Install the `I\TeX` package `alg` with no source files and no documentation:

```bash
>> sh install-pkg.sh --package=alg
```
An introduction to \TeX\ Live 4

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
name & selection & size \\
\hline
1 & bibtex & 7597 kB \\
2 & doc & 21152 kB \\
3 & dvips & 430 kB \\
4 & etex & 102 kB \\
5 & fonts & 51447 kB \\
6 & formats & 14651 kB \\
7 & generic & 459 kB \\
8 & graphics & 9674 kB \\
9 & lang & 19618 kB \\
10 & latex & 23429 kB \\
11 & metapost & 1443 kB \\
12 & omega & 4986 kB \\
13 & pdftex & 471 kB \\
14 & plain & 1113 kB \\
15 & texlive & 10155 kB \\
\hline
SUM: & & 166829 kB \\
\hline
\end{tabular}
\end{center}

\begin{flushleft}
\textbf{global commands:} select \texttt{<N>one} / \texttt{<B>asic} / \texttt{R<E>commended} / \texttt{<A>ll} for all collections
\texttt{<R>} return to platform menu
\texttt{<Q>} quit
\end{flushleft}
Collection: Fonts
================================================================================

Fonts, including metrics, virtual fonts and sources
================================================================================

<N> No packages
<B> Basic packages [1023 kB]
<E> Basic + Recommended packages [51447 kB]
<A> All packages [127417 kB]
================================================================================

<R> return to collection menu
<Q> quit

Enter command:

Figure 3. Customizing a collection

• Install all the packages available in the other Plain TeX collection:

  $ sh install-pkg.sh --collection=plain3$

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

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

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.

Installation and use under Windows

This section only applies to systems running Windows 9x or NT. If you run Windows 3.1, you will have to install emtex from the top level systems directory by hand.

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 Windows 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.

What is fppTeX?

The system on the CD-ROM for Windows is Fabrice Popineau's fppTeX. This is a port to Windows 9x and Windows NT—referred to as Win32—of the well known distribution teTeX for Unix. More precisely, given obvious differences between Unix and Win32, some things behave differently under fppTeX, some are still missing, some are just different, but the large majority behave just the same as under Unix.

What is in this port

This version includes all the same programs as the Unix side of TeX Live, with some additions which can be installed.

Among the programs, there are a few DLLs:
- the ones that begin with msvc are the Microsoft C library targeted for multi-threaded applications,
- the Kpathsea dynamic-linked library,
- zlib.dll (compress library) and libpng.dll (Portable Network Graphics) for pdfTeX and a few other programs,
- tex.dll, pdftex.dll and a few others refer to TeX engines—see the explanation below.

All the various TeX engines are distributed in the form of a .dll file for the core, and a .exe file for the front-end. This is the answer to the problem of link files that do not exist on Win32 platforms. For example, the TeX engine is made up of:

```
11/19/98 11:07a 217,088 tex.dll
11/19/98 11:07a 16,384 tex.exe
```

and the latex.exe file is nothing but a rough copy of tex.exe using the same core tex.dll. The same approach is used for the mktex*.exe family of programs which are linked to the mktex.dll library.

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 do this, you must add the bin/win32 directory of the CD-ROM to your PATH, using the Windows configuration software. Now you can run the programs at a command prompt, or use the shareware WinEdt editor, which runs the programs from convenient menus.

How to install it

When you put TeX Live in your computer, the setup should run automatically; if it does not, run the program autorun.exe. Now follow the instructions—here are some hints:
- Choose a root for your installation, c:\TeX is proposed by default, but you can change it because you will need lots
of disk space: more than 300Mb for a full installation, and beware that the cluster size on FAT partitions will make the package appear even bigger;  
• Do not use any path with embedded ‘space’ character: TeX won’t like that. The setup.exe checks for that anyway;  
• Your main texmf tree will be root/texmf and designated by the variable $TEXMFPARENT;  
• You have the opportunity to add some more texmf trees:  
  – one local texmf tree, which is designated by the variable $TEXMFLocal and is by default root/texmf/ local. It is intended to store your site local macros and style files, and also any locally generated font files. If you do not specify a local texmf tree, you will be asked to set the variable VARTEXFONTs to point to some directory where those fonts will be stored. If you specify a local texmf tree, it will be used for those fonts.  
  – one home texmf tree, which is designated by the variable $HOMEtexmf and is by default $HOME/texmf. This is meaningful only under Windows NT, where users have a "$HOME". Usually, Windows 9x users do not have a "$HOME", so they should leave this place empty;  
These locations can be edited manually by looking for their variables names in the file texmf/web2c/texmf.cnf.  
• You will be asked if you want to install:  
  – PK fonts,  
  – only free packages, in which case some packages will even disappear from the list for a custom installation,  
  – if you want the documentation accompanying each of the packages being installed; guides and general documentation will always be installed, but not the material that is provided with some specific package;  
  – idem for source files.  
• Choose your setup from basic, recommended and full. If you already know what you want to install, you can also choose to do a custom setup. In this case, you will be presented with a list of collections of packages. Each collection is (de)selectable by itself, and so is each package individually. A short description is provided for many packages, thanks to the Web catalogue of Graham Williams.  
• Ignore the ‘setup’ collection in the list of collections. It refers to the files in the \setupv32 on the CD-ROM and should have been made invisible;  
• You will have the opportunity to add some more special packages, namely Ghostscript and Ghostview, the PostScript interpreter and previewer, ImageMagick which allows for image manipulations and conversions, WinEdt which is a good shareware environment for editing and typesetting with TeX, and textshell which does quite the same as WinEdt (free but less fancy).  
• You can review your installation settings, and if everything is okay, the file transfer will begin;  
• Once it is done, the installers for Ghostview and WinEdt will be called if selected. The installation and configuration of everything else is handled by setup.exe. It is completely automatic. The last step is the build of ‘ls-R’ files;  
• Eventually, the setup will be over and the documentation displayed using your default Web browser. Windows 9x users will need to reboot before being able to run anything.  
A number of items will appear under the Start-> Programs->TeXLive menu.  
What does the setup hide from me ?  
If you want to hack the configuration by yourself, here is a more detailed description of what the setup does — and what it does not. Your PATH is modified to make the programs installed accessible. It is checked for any older version of fpTeX or TeX-Live, and if one is found, its entry in PATH is removed. This is done by looking for the file kpathsea.d11 along your PATH.  
If Ghostscript installation has been requested, the location of gs5.50 is added to your PATH because the files gswin32c.exe (command line interface to Ghostscript) and gsdll32.d11 (dll embedded Ghostscript) are accessed by several programs of the distribution. Ghostscript uses the registry now (version 5.50) and so does not need any other setting to find the relevant files. Previous or customized installations might require you to set the GS_LIB variable. See the appropriate documentation.  
ImageMagick is also added to your PATH if it has been selected. Moreover, the right delegates.mk file is copied according to your platform (Windows 9x or Windows NT). See ImageMagick documentation for more details.  
Your main <root>/texmf/web2c/texmf.cnf file is edited to reflect the texmf trees you have specified, and the location of locally generated fonts. The variables $TEXMF, $TEXMFLocal, $HOMETEXMF and $VARTEXFONTS are modified.  
The file <root>/texmf/web2c/mktext.cnf is edited to add the feature ‘varfonts’, which will force any locally generated font to be stored in $VARTEXFONTS.  
The configuration for tex4ht is undertaken by editing <root>/texmf/tex4ht/base. The only relevant part is that it needs to run the convert.exe tool of ImageMagick, so the full path to access it is provided.  
Testing the installation  
A valuable tool to test the installation now is the program kpsewhich.  
As a first step, you should check if Web2C correctly identifies the location of your texmf tree. Open a command prompt window and type  

```
kpsewhich -expand-path=$TEXMF
```

The answer should be the location of your texmf tree (e.g., c:/TeX/texmf if you unpacked the archive files as in the example above — note that the answer is a Unix style path, i.e., the MS-Dos style \ is substituted by /; you don’t have to worry about this). If you do not get the right answer you have probably changed the default directory structure. In this case you have to set the variable TEXMF manually to the root directory of your texmf tree.  
If you want to be on the safe side, you may type in mktextslr to update the ls-R database, even though a proper ls-R file should be provided after installation.
Network installation and filesystem considerations

All the support files, everything except the files in bin/win32, are shareable with a Unix installation. This means you can use Samba to either mount from a Windows 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 operating system and architecture you want to use in the bin directory. That means for example bin/win32 and bin/1386-linux.
- Install a local copy of the binaries and format files. In this case, assign $TEXMFMAIN to the main texmf tree that will lie on the network.

These schemes should have been handled by the InstallShield installer. But so many problems arose with this installer that these features have been delayed to the next version of the setup program.

Win32 supports multiple filesystems:

- MS-Dos FAT, 8.3 and uppercase filenames
- Protected mode FAT, long filenames, but case-insensitive
- NTFS, long filenames and case-sensitive
- ISO9660 CD-ROM, 8.3 and uppercase filenames

Moreover, Win32 calls which refer to filenames are case-insensitive, and there are several other features in NTFS that Win32 can’t use for the moment. Another dimension is the use of different directory separators: / or \, but Win32 calls accept both.

So what difficulties may arise?

Most likely, you will have some style files with long filenames. If you are running on a filesystem which supports them,3 there is no problem and you have nothing to do. Otherwise, you will need to use the alias feature of Kpathsea. Suppose, for instance, you are trying to install texmf on a FAT partition and you have the style file named longtabl.sty in your tree. The filename will be truncated to its 8.3 form: longtabl.sty. In this case, you will need to create a file named aliases in the same place as the ls-R file in your texmf tree. This file should contain the following line:

```
longtabl.sty longtabl.sty
```

All references to longtable.sty will be redirected to longtabl.sty if the long filename is not found.

Otherwise, if you think you have trouble with filenames, consider doing the following:

- paths in config files and environment variables should be written using / rather than \;
- ls-R databases should be in lower case, even if you are running on FAT or CD-ROM;
- use the debug feature of Kpathsea and kpsewhich to demonstrate your problem and email us the results of your investigations.

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).

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.

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., `/var/TeX` or `/usr/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/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
- `$TEXDIR/info` GNU ‘info’ manuals
- `$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 (that is, put binaries directly into $TEXDIR/bin), you can use the configure option --disable-multiplatform.

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

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" &
```

\footnote{For example, NTFS but not FAT!}
Before you think that everything is okay, 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:

```bash
>> cd /usr/TeX/bin/i586-pc-linux-gnu
>> ls | wc
```

The result should be 204. `make world` is equivalent to make all install strip.

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

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

### Final configuration steps

Set up your `PATH` to include the directory containing the just-installed binaries (e.g., `/usr/TeX/bin/mips-sgi-irix6.3`); 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

```bash
>> texconfig help
```

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

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

### A user’s guide to the Web2c system

Web2c contains a set of \TeX related programs, i.e., \TeX itself, METAFONT, METAPOST, Bn\TeX, etc; it works on Unix, Windows 3.1, 9x/NT, 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);
- `dvistuff` Produces modified copy of DVI file;
- `dvitomp` DVI to MPX (METAPOST pictures);
- `dvitox` DVI to human-readable text;
- `gftpdf` Generic font proofoheets;
- `gftox` Generic to packed fonts;
- `gftyphe` GF to human-readable text;
- `makempx` METAPOST label typesetting;
- `mf` Creating typeface families;
- `mft` Prettyprinting METAFONT source;
- `mpost` Creating technical diagrams;
- `mptox` METAPOST label extraction;
- `newer` Compare modification times;
- `patgen` Creating hyphenation patterns;
- `pktogf` Packed to generic fonts;
- `pktyp` PK to human-readable text;

- `ptotf` Property list to TFM;
- `pooltype` Display web pool files;
- `tangle` web to Pascal;
- `tex` Typesetting;
- `tftool` TFM to property list;
- `vftovf` Virtual font to virtual property list;
- `vptovf` Virtual property list to virtual font;
- `weave` web to \TeX.

### Kpathsea path searching

Let us first describe the generic path searching mechanism of the Kpathsea library. We call a `path element` any directory name. 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 11) 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.

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 13).

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 TEXMF.CNF (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 the section starting on page 16, 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.[programe] [=] value
```

where the “=” and surrounding whitespace are optional.
- The “variable” name may contain any character other than whitespace, “=”, or “.”, but sticking to “A-Za-z_” is safest.
- If “.programe” is present, the definition only applies if the program that is running is named programe or programe.exe. This allows different flavors of \TEX to have different search paths, for example.

- “value” may contain any characters except “%” and “\”. The \var.programe feature is not available on the right-hand side; instead, you must use an additional variable. A “;” in “value” is translated to “:” if running under Unix; this is useful to be able to have a single \texmf.cnf for Unix, MS-Dos and Windows systems.
- All definitions are read before anything is expanded, so variables can be referenced before they are defined.

A configuration file fragment illustrating most of these points is shown in Figure 4.

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.

Default expansion

If the highest-priority search path (see “Path sources” on page 10) contains an extra colon (i.e., leading, trailing, or double), 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 “;”.

Brace expansion

A useful feature is brace expansion, which means that, for instance, \{a,b\} 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 (on one line!):

```
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 \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.
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 ($\TeX{}M$ 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 \texttt{mktexlsr} script included with the distribution. It is invoked by the various \texttt{mktex}... scripts. In principle, this script just runs the command

\begin{verbatim}
cd /your/texmf/root \&& ls -LAR ./ \> ls-R
\end{verbatim}

presuming your system’s \texttt{ls} produces the right output format (GNU’s \texttt{ls} is all right). To ensure that the database is always up-to-date, it is easiest to rebuild it regularly via \texttt{cron}, so that for changes in the installed files—perhaps after installing or updating a \TeX{} package—the file \texttt{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.

\texttt{kpselwhich}: Standalone path searching

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

\begin{verbatim}
>>> kpselwhich option... filename...
\end{verbatim}

The options specified in “\texttt{option}” can start with either “-” or “\texttt{--}”, and any unambiguous abbreviation is accepted.

\texttt{kpselwhich} 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 “\texttt{find}” utility for that).

The more important options are described next.

-\texttt{dpi=num} Set the resolution to “\texttt{num}”; this only affects “gf” and “pk” lookups. “-D” is a synonym, for compatibility with \texttt{dvips}. Default is 600.

-\texttt{format=name} Set the format for lookup to “\texttt{name}”. By default, the format is guessed from the filename. For formats which do not have an associated unambiguous suffix, such as \texttt{METAPOST} support files and \texttt{dvips} configuration files, you have to specify the name as found in the first column of Table 1 on p. 15, which lists currently recognized names, a description, associated environment variables,\footnote{You can find definitions for these environment variables in the file \texttt{texmf.cnf} (page 16)}, and possible file extensions.

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 \texttt{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 file-name. “/” and all the usual expansions are supported. The options “-path” and “-format” are mutually exclusive.

-programname=name
  Set the program name to “name”. This can affect the search paths via the “.program” 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”.

Examples of use

Let us now have a look at kpsewhich in action.

```
>> kpsewhich article.cls
 /usr/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 “TEXMEROOT” root directory. Similarly, all of the following are found without problems thanks to their unambiguous suffix.

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

The last item 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/texmf/fonts/pk/ljfour/jknappen/...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 `mktecpk`.

Next we turn our attention to `dvips`’s header and configuration files. We first look at one of the commonly used files, the general prolog `tex.pro` for `T`eX support, before turning our attention to the generic configuration file `config.ps` and the PostScript font map `psfonts.map`. As the “.ps” suffix is ambiguous we have to specify explicitly which type we are considering (“dvips config”) for the file `config.ps`.

```
>> kpsewhich --format="dvips config" config.ps
 /usr/texmf/dvips/config/config.ps
```

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" utm-config.
 /usr/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/texmf/dvips/psnfss/config.utm
```

This map file defines the file names of the Type1 PostScript fonts in the URW collection. Its contents look like (we only show some of the lines):

```
utm8r NimbusRomNo9L-Medi ... <utm8a.pfb
utm8r NimbusRomNo9L-MedIt... <utm8a.pfb
utmr8r NimbusRomNo9L-Regu ... <utmr8a.pfb
utmr8r NimbusRomNo9L-Regult... <utmr8a.pfb
utmr8r NimbusRomNo9L-RegulMed... <utmr8a.pfb
```

Let us, for instance, take the Times Regular instance `utm8a.pfb` and find its position in the `texmf` directory tree by using a search for Type1 font files:

```
>> kpsewhich utm8a.pfb
 /usr/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.
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. \texttt{stat} calls (file tests). When running with an up-to-date \texttt{ls-R} database this should almost give no output.
2. References to hash tables (like \texttt{ls-R} database, map files, configuration files).
3. File open and close operations.
4. General path information for file types searched by Kpathsea. This is useful to find out where a particular path for the file was defined.
5. Directory list for each path element (only relevant for searches on disk).
6. A value of \texttt{-1} will set all the above options; in practice you will probably always use these levels if you need any debugging.

Similarly, with the \texttt{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 \texttt{KPATHSEA_DEBUG} environment variable, and setting it to (a combination of) values as described in the above list.

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

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

This little file uses only the font \texttt{cmr10}, so let us look how \texttt{dvips} prepares the PostScript file (we want to use the Type1 version of the Computer Modern fonts, hence the option \texttt{-Pcms}).

\begin{verbatim}
>> dvips -d4100 hello-world -Pcms -o
\end{verbatim}

In this case we have combined \texttt{dvips}'s debug class 4 (font paths) with Kpathsea's path element expansion (see \texttt{dvips} Reference Manual, \texttt{texmf/doc/html/dvips/dvips_toc.html}). The output (slightly rearranged) appears in Figure 5. \texttt{dvips} starts by locating its working files. First, \texttt{texmf.cnf} is found, which gives the definition of the search paths for the other files, then the file database \texttt{ls=R} (to optimize file searching) and the file \texttt{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 \texttt{dvips} goes on to find the generic configuration file \texttt{config.ps}, before looking for the customization file \texttt{dvipsrc} (which, in this case is \texttt{not found}). Finally, \texttt{dvips} locates the config file for the Computer Modern PostScript fonts \texttt{config.cms} (this was initiated with the \texttt{-Pcms} option on the \texttt{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.

\begin{verbatim}
>> more /usr/texmf/dvips/cms/config.cms
p +ams.map
p +cmbkm.map
p +ansbkm.map
\end{verbatim}

\texttt{dvips} thus goes on to find all these files, plus the generic map file \texttt{psfonts.map}, which is always loaded (it contains declarations for commonly used PostScript fonts; see the last part of the previous sub-section for more details about PostScript map file handling).

At this point \texttt{dvips} identifies itself to the user...

\begin{verbatim}
This is dvips 5.78 Copyright 1998 Radical...n
\end{verbatim}

then goes on to look for the prolog file \texttt{texc.pro}.

After having found the file in question, \texttt{dvips} outputs date and time, and informs us that it will generate the file \texttt{hello-world.ps}, then that it needs the font file \texttt{cmr10}, and that the latter is declared as “resident”:

\begin{verbatim}
TeX output 1998.02.26:1204' -> hello-world.ps
Defining font () cmr10 at 10.0pt
Font cmr10 <CMR10> is resident.
\end{verbatim}

Now the search (Figure 6) starts for \texttt{cmr10.tfm}, which is found, then a few more prolog files (not shown) are referenced, and finally the Type1 instance \texttt{cmr10.pfb} of the font is located and included in the output file (see last line).

Runtime options
Another of the nice features of Web2c 7.3 is its possibility to control a number of memory parameters (in particular, array sizes) via the runtime file \texttt{texmf.cnf} read by Kpathsea. The listing of \texttt{texmf.cnf} is shown in the section starting on page 16; the settings of all parameters can be found in Part 3 of that file. The more important control variables are:

\texttt{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 \texttt{hugetex.fmt}. Using the standard way of specifying the program name used by Kpathsea, the particular value of the \texttt{main_memory} variable will then be read from \texttt{texmf.cnf} (compare the generic value and the “huge” one instantiated by \texttt{hugetex}, etc.).

\texttt{extra_mem_bot} Extra space for “large” \TeX data structures: boxes, glue, breakpoints, etc. Especially useful if you use \texttt{PCTeX}.

\texttt{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.

\texttt{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 \texttt{hugetex} and \texttt{pdflatex} program invocations ask for an extra 15,000 control sequences (the default value of \texttt{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.
Figure 5. Finding configuration files

Figure 6. Finding the Type 1 font file
## Table 1: Kpathsea file types

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variables</th>
<th>Suffixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>afm</td>
<td>Adobe font metrics</td>
<td>AFMFONTS</td>
<td>.afm</td>
</tr>
<tr>
<td>base</td>
<td>METAFONT memory dump</td>
<td>MFBASES, TEXMFINI</td>
<td>.base</td>
</tr>
<tr>
<td>bib</td>
<td>Bib\TeX\ bibliography source</td>
<td>BIBINPUTS, TEXBIB</td>
<td>.bib</td>
</tr>
<tr>
<td>bst</td>
<td>Bib\TeX style files</td>
<td>ESTINPUTS</td>
<td>.bst</td>
</tr>
<tr>
<td>cnf</td>
<td>Runtime configuration files</td>
<td>TEXMCNF</td>
<td>.cnf</td>
</tr>
<tr>
<td>dvips</td>
<td>dvips configuration files, e.g., config.ps and psfonts.map</td>
<td>TEXCONFIG</td>
<td>.map</td>
</tr>
<tr>
<td>fmt</td>
<td>\TeX memory dump</td>
<td>TEXFORMATS, TEXMFINI</td>
<td>.fmt, .efmt,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>.efm</td>
</tr>
<tr>
<td>gf</td>
<td>generic font bitmap</td>
<td>GFFONTS</td>
<td>.gf</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>graphic/figure</td>
<td>Encapsulated PostScript figures</td>
<td>TEXPICTS, TEXINPUTS</td>
<td>.eps, .epsi</td>
</tr>
<tr>
<td>ist</td>
<td>makeindex style files</td>
<td>TEXINDEXSTYLE, INDEXSTYLE</td>
<td>.ist</td>
</tr>
<tr>
<td>ls-R</td>
<td>Filename databases</td>
<td>TEXMDFS</td>
<td>.map</td>
</tr>
<tr>
<td>mem</td>
<td>METAPOST memory dump</td>
<td>MPMEMS, TEXMFINI</td>
<td>.mem</td>
</tr>
<tr>
<td>mf</td>
<td>METAFONT source</td>
<td>MFINPUTS</td>
<td>.mf</td>
</tr>
<tr>
<td>mpool</td>
<td>METAFONT program strings</td>
<td>MPOOL, TEXMFINI</td>
<td>.pool</td>
</tr>
<tr>
<td>mft</td>
<td>MFT style file</td>
<td>MFTINPUTS</td>
<td>.mft</td>
</tr>
<tr>
<td></td>
<td>miscellaneous fonts</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MISCFONTS</td>
<td></td>
</tr>
<tr>
<td>mp</td>
<td>METAPOST source</td>
<td>MPINPUTS</td>
<td>.mp</td>
</tr>
<tr>
<td>mpool</td>
<td>METAPOST program strings</td>
<td>MPOOL, TEXMFINI</td>
<td>.pool</td>
</tr>
<tr>
<td>MetaPost support</td>
<td>METAPOST support files, used by DMP</td>
<td>MFSUPPORT</td>
<td></td>
</tr>
<tr>
<td>ocp</td>
<td>\ Omega compiled process files</td>
<td>OCPINPUTS</td>
<td>.ocp</td>
</tr>
<tr>
<td>ofm</td>
<td>\ Omega font metrics</td>
<td>OFMFONTS, TEXFONTS</td>
<td>.ofm, .tfm</td>
</tr>
<tr>
<td>opl</td>
<td>\ Omega property lists</td>
<td>OPLFONTS, TEXFONTS</td>
<td>.opl</td>
</tr>
<tr>
<td>otp</td>
<td>\ Omega translation process files</td>
<td>OTPINPUTS</td>
<td>.otp</td>
</tr>
<tr>
<td>ovf</td>
<td>\ Omega virtual fonts</td>
<td>OVFONTS, TEXFONTS</td>
<td>.ovf</td>
</tr>
<tr>
<td>ovp</td>
<td>\ Omega virtual property lists</td>
<td>OVPFONTS, TEXFONTS</td>
<td>.ovp</td>
</tr>
<tr>
<td>pk</td>
<td>packed bitmap fonts</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PostScript header</td>
<td>downloadable PostScript</td>
<td>TEXPSHEADERS, PSHEADERS</td>
<td>.pro, .enc</td>
</tr>
<tr>
<td>tex</td>
<td>\TeX source</td>
<td>TEXINPUTS</td>
<td>.tex, .cls,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>.sty, .clo,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>.def</td>
</tr>
<tr>
<td>TeX system documentation</td>
<td>Documentation files for the \TeX system</td>
<td>TEXDOCS</td>
<td></td>
</tr>
<tr>
<td>TeX system sources</td>
<td>Source files for the \TeX system</td>
<td>TEXSOURCES</td>
<td></td>
</tr>
<tr>
<td>texpool</td>
<td>\TeX program strings</td>
<td>TEXPOOL, TEXMFINI</td>
<td>.pool</td>
</tr>
<tr>
<td>tfa</td>
<td>\TeX font metrics</td>
<td>TPFONTS, TEXFONTS</td>
<td>.tfa, .ttc</td>
</tr>
<tr>
<td>Troff fonts</td>
<td>Troff fonts, used by DMP</td>
<td>TPFONTS</td>
<td></td>
</tr>
<tr>
<td>true type fonts</td>
<td>True Type outline fonts</td>
<td>T1PFONTS</td>
<td></td>
</tr>
<tr>
<td>type1 fonts</td>
<td>Type 1 PostScript outline fonts</td>
<td>T1INPUTS, T1PFONTS, TEXPSHEADERS, DVIPSHEADERS</td>
<td>.pfa, .pfb</td>
</tr>
<tr>
<td>type42 fonts</td>
<td>Type 42 PostScript outline fonts</td>
<td>T42PFONTS</td>
<td></td>
</tr>
<tr>
<td>vf</td>
<td>virtual fonts</td>
<td>VFFONTS, TEXFONTS</td>
<td>.vf</td>
</tr>
<tr>
<td>web2c files</td>
<td>Web2c support files</td>
<td>WEB2C</td>
<td></td>
</tr>
<tr>
<td>other text files</td>
<td>text files used by ‘foo’</td>
<td>FOGINPUTS</td>
<td></td>
</tr>
<tr>
<td>other binary files</td>
<td>binary files used by ‘foo’</td>
<td>FOGINPUTS</td>
<td></td>
</tr>
</tbody>
</table>
The `texmf.cnf` file

% TeX Live `texmf.cnf`
% 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 `$SELFAUTO...` variables here so TeX will
% find where to look dynamically. See the manual and the definition
% below of `TEXMF.cnf`.
%
% The main tree, which must be mentioned in `$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.
TEXMF = {"HOMETEXMF,"!!"TEXMFLocal,"!!"VARTEXMF,"!!"TEXMFMAIN"}

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

% 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 MTFEATURES in `mktex.cnf`.
VARTEXFONTS = /var/tmp/texfonts

% 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,"!!"VARTEXFONTS"

% 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
VARTEXM = /var/lib/texmf
% is used on many Linux systems. In this case, set `VARTEXFONTS` like this
VARTEXFONTS = "$VARTEXM/fonts"
% and do not mention it in `TEXMFDBS` (but _do_ mention `VARTEXMF`).
% Usually you will not need to edit any of the other variables in part 1. %
% `WEB2C` is for `Web2C` specific files. The current directory may not be
% a good place to look for them.
WEB2C = "$TEXMF/web2c"
An introduction to \TeX\ Live

% 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.texinfo = .;TEXMF/tex/{texinfo,plain,generic,};//
TEXINPUTS.tex = .;TEXMF/tex/{plain,generic,};//
TEXINPUTS.pdfjade = .;TEXMF/pdftex/pdftex/etex.tex/{jade,plain,generic,};//

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

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

% Omega.
TEXINPUTS.lambda = .;TEXMF/omega/omega.tex/{lambda,latex,generic,};//
TEXINPUTS.omega = .;TEXMF/omega/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/cslatex/cslatex,csplain,latex,generic,};//
TEXINPUTS.csplain = .;TEXMF/csplain,csplain,plain,generic,};//
TEXINPUTS.pdfcslatex = .;TEXMF/pdftex/pdftex/cslatex,csplain,latex,generic,};//
TEXINPUTS.pdfcsplain = .;TEXMF/pdftex/pdftex/csplain,csplain,plain,generic,};//

% Polish
TEXINPUTS.platex = .;TEXMF/platex/platex,latex,generic,};//
TEXINPUTS.pdfex = .;TEXMF/pdfex/pdfex/mex,plain,generic,};//
TEXINPUTS.mex = .;TEXMF/mex/mex,plain,generic,};//

% french
TEXINPUTS.fr.tex = .;TEXMF/frtex/frtex,frtex,latex,generic,};//
TEXINPUTS.frtex = .;TEXMF/frtex/frtex,frtex,latex,generic,};//

% ML\TeX
TEXINPUTS.ml = .;TEXMF/mltex/mltex,plain,generic,};//
TEXINPUTS.ml = .;TEXMF/mltex/mltex,latex,generic,};//

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

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

% Metafont, MetaPost inputs.
MFINPUTS = .;TEXMF/metafont/;{$TEXMF/fonts,$VARTEXFONTS}/source/
% 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 = .;$TEXMF/web2c
MBASES = .;$TEXMF/web2c
MPMEMS = .;$TEXMF/web2c
TEXPOOL = .;$TEXMF/web2c
MFPOOL = .;$TEXMF/web2c
MPPOOL = .;$TEXMF/web2c

% 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.
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 = .;$TEXMF/{tex,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfelatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.pdfxelatex = .;$TEXMF/{text,pdftex,dvips,fonts/type1};//
TEXPSHEADERS.context = .;$TEXMF/{etex,tex,pdftex,dvips,fonts/type1};//

% PostScript Type 1 outline fonts.
T1FONTS = .;$TEXMF/fonts/type1//

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

% TrueType outline fonts.
TTFONTS = .;$TEXMF/fonts/truetype//

% Type 42 outline fonts.
T42FONTS = .;$TEXMF/fonts/type42//
An introduction to TeX Live 4

% A place to put 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/

% 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.

MIMELIBDIR = c:/TeX/etc

MAILCAPLIBDIR = c:/TeX/etc

% TeX documentation and source files, for use with kpsewhich.

TEXDOCS = .;/TEXMF/doc/

TEXSOURCES = .;/TEXMF/source/

% 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 returned.

OFMFONTS = .;/$TEXMF/fonts,$VARTEXFONTS}/ofm,tfm//;$TFMFONTS

OPLFONTS = .;/$TEXMF/fonts,$VARTEXFONTS}/opl/

OVPFONTS = .;/$TEXMF/fonts,$VARTEXFONTS}/ovp/

OVFFONTS = .;/$TEXMF/fonts,$VARTEXFONTS}/ovf/

OTPINPUTS = .;/TEXMF/omega/otp/

OCPINPUTS = .;/TEXMF/omega/ocp/

%% TeX4ht utility, sharing files with TeX4ht

T4HTINPUTS = .;/TEXMF/tex4ht/

%% The mktex* scripts rely on KPSE_DOT. Do not set it in the environment.

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
% 2) if we used /$VARNAME, we’d end up with /// if VARNAME is defined,
% which would search the entire world.
% The TETEXDIR stuff isn’t likely to be relevant unless you’re using TeX,
% but it doesn’t hurt.

TEXMFCNF = .;/$SELFAUTOLOC,$SELFAUTODIR,$SELFAUTOPARENT}\n{},/texmf/.local,)/web2c;c:/TeX/texmf/web2c

% Part 2: Non-path options.

% 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
%
% 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
% MKTEXMF = 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.
%
% For some xy-pic samples, you may need as much as 700000 words of memory.
% For the vast majority of documents, 60000 or less will do.
%
% main_memory = 263000 % words of inimemory available; also applies to inifm&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 = 0

% 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.

pool_size = 125000

% 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 = 25000

max_strings = 15000  % max number of strings
pool_free = 5000    % min pool space left after loading .fmt

% 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.

trie_size = 64000

% 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 = 50000

% These are Omega-specific.
ocp_buf_size = 20000  % character buffers for ocp filters.
ocp_stack_size = 10000  % stacks for ocp computations.
ocp_list_size = 1000  % control for multiple ocps.

dvi_buf_size = 16384  % TeX

gf_buf_size = 16384  % MF

% 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.

save_size = 4000  % for saving values outside current group
param_size = 500   % simultaneous macro parameters
max_in_open = 15  % simultaneous input files and error insertions

% 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;

error_line = 79
half_error_line = 50
max_print_line = 79

stack_size = 300  % simultaneous input sources

% simultaneous semantic levels (e.g., groups)

main_memory.context = 1100000
hash_extra.context = 25000
pool_size.context = 750000
string_vacancies.context = 45000
max_strings.context = 55000
pool_free.context = 475000
nest_size.context = 500
param_size.context = 1500
save_size.context = 5000
stack_size.context = 1500

main_memory.hugetex = 1100000
param_size.hugetex = 1500
stack_size.hugetex = 1500
hash_extra.hugetex = 15000
string_vacancies.hugetex = 45000
pool_free.hugetex = 475000
nest_size.hugetex = 500
save_size.hugetex = 5000
pool_size.hugetex = 500000
max_strings.hugetex = 55000

main_memory.hugelatex = 1100000
param_size.hugelatex = 1500
stack_size.hugelatex = 1500
hash_extra.hugelatex = 15000
string_vacancies.hugelatex = 45000
pool_free.hugelatex = 475000
nest_size.hugelatex = 500
save_size.hugelatex = 5000
pool_size.hugelatex = 500000
max_strings.hugelatex = 55000

main_memory.jadetex = 1500000
param_size.jadetex = 1500
stack_size.jadetex = 1500
hash_extra.jadetex = 50000
string_vacancies.jadetex = 45000
pool_free.jadetex = 475000
nest_size.jadetex = 500
save_size.jadetex = 5000
pool_size.jadetex = 500000
max_strings.jadetex = 55000

main_memory.pdfjadetex = 2500000
param_size.pdfjadetex = 1500
stack_size.pdfjadetex = 1500
hash_extra.pdfjadetex = 50000
string_vacancies.pdfjadetex = 45000
pool_free.pdfjadetex = 475000
nest_size.pdfjadetex = 500
save_size.pdfjadetex = 5000
pool_size.pdfjadetex = 500000
max_strings.pdfjadetex = 55000

main_memory.pdflatex = 1500000
param_size.pdflatex = 1500
stack_size.pdflatex = 1500
hash_extra.pdflatex = 15000
string_vacancies.pdflatex = 45000
pool_free.pdflatex = 475000
nest_size.pdflatex = 500
pool_size.pdflatex = 500000
save_size.pdflatex = 5000
max_strings.pdflatex = 55000
An introduction to \TeX\ Live 4

LAT\TeX 

Line by Line
2nd Edition
Tips and Techniques for Document Processing

ANTONI DILLER

"\LaTeX\_ is a document preparation system which can be used to type set a wide variety of documents, but you do not need to know very much about typesetting in order to use it."


Written for \LaTeX\ novices, this easy-to-follow tutorial provides clear explanations and realistic examples. It is designed to help users find solutions to specific tasks, such as indexing, setting fundamental and complicated mathematical formulae and producing simple line diagrams. It also aims to guide readers through the process of creating documents as simple as letters and articles or as complex as books - all to a professional presentation quality. A glossary provides guidance on standard commands and an appendix explains the differences between \LaTeX\_ and the original version. Many of the typesetting examples from the book are coded as templates and are available on the accompanying Website.

0471 97918X 320pp £27.50 December 1998

Visit the accompanying Website at: www.cs.bham.ac.uk/~ard
UK-T\TeX~Information

The 1998–99 UK-T\TeX~committee

Chairman
Philip Taylor: P.Taylor@mail.rhbnc.ac.uk

Treasurer
Peter Abbott: peter.abbott@tex.ac.uk

Secretary
John Palmer: johnp@bcs.org.uk

Membership Secretary
Phil Molyneux: molyneux@kingston.ac.uk
Kaveh Bazargan: kaveh@focal.demon.co.uk
Malcolm Clark: malcolm.clark@kcl.ac.uk
Roy Everett: rpe@anglianet.co.uk
James Foster: j.foster@sussex.ac.uk
David Hardy: dhardy@sunrise-setting.co.uk
Hong Ji: hong@focal.demon.co.uk
Kim Roberts: robertsk@oup.co.uk
Dominik Wujastyk: ucgadkw@ucl.ac.uk

Contacting UK-T\TeX

Orders for books or software should be addressed to the Treasurer: Peter Abbott, 1 Eyemore Close, Selly Oak, Birmingham B29 4LB. Fax/telephone: 0121-476 2159.

General enquiries should be sent to the Secretary: John Palmer, 69 St Cross Road, Winchester SO23 9RE. Telephone: 01962 865261.

Please get in touch with the Membership Secretary [Phil Molyneux, Kingston Business School, Kingston University, Kingston Hill, Kingston-upon-Thames KT2 7LB; e-mail: molyneux@kingston.ac.uk; tel (work) 0181-547 2000 ext 5233, (home) 0181-549 0045] regarding payment of UK-T\TeX~subscriptions, or changes to personal details held on file.

UK-T\TeX~maintains pages on the World Wide Web at http://uk.tug.org/ E-mail enquiries about UK-T\TeX~may be sent to uktug-enquiries@tex.ac.uk.

Baskerville

Articles should be submitted using electronic mail to baskerville@tex.ac.uk; contributors who cannot submit using e-mail may seek advice from the Secretary. Any correspondence concerning Baskerville may also be addressed in this way. Earlier issues of Baskerville can be found on CTAN in usergrps/uktug/baskervi.

Book Discounts for UK-T\TeX~members

We have arrangements with Addison-Wesley for their well-known \TeX-related publications, with International Thomson Publishing for the O'Reilly & Associates Inc. series, and with Oxford University Press, to supply books to members at discounted prices.

Please send details (including ISBN if possible) of books required to Peter Abbott. Peter will supply a quotation.

We are allowed to offer this service only to current members of UK-T\TeX~and/or members of TUG. Please send your order and cheque (in UK £) to Peter Abbott. Cheques should be made payable to ‘UKTUG’. All books will be routed through UK-T\TeX. In all cases, please notify Peter Abbott by e-mail, 'phone, fax or letter when books are delivered. Provided that the book(s) are in stock, it will normally take about a week from receipt of order to delivery of the book(s).

Obtaining \TeX~from CTAN

The UK \TeX~Archive at ftp://ftp.tex.ac.uk/ is part of the CTAN (Comprehensive \TeX~Archive Network) collaborating network of archives on the Internet organised under the ægis of the \TeX~Users Group.

The CTAN archives run an enhanced \ftp~server which supports dynamic compression, uncompression, and archive-creation options. Fetch the top-level file README. archive-features for information. The server also supports site-defined commands to assist you. Please read README. site-commands for a brief overview.

For interfaces to, and catalogues of, CTAN, you are referred to Graham Williams' \TeX~and \ET\TeX~Catalogue which is available from CTAN as help/Catalogue/catalogue.html; other interfaces are listed in http://www.tug.org/interest.html.

Please report any problems with the CTAN archives to ctan@ctan.org.

Obtaining \TeX~on CD-ROM

UK-T\TeX, in collaboration with GUTenberg, TUG, NTG and DANTe e.V., have produced \TeX~Live, a “plug-and-play” CD-ROM for Linux, Unix, and Windows 32, based on the Web2c \TeX~setup. As it is formatted according to ISO 9660, with both Joliet and Rock Ridge extensions, the platform-independent files can, in principle, be read on all operating systems which can deal with that format. The CD is supplied free of charge to all members of UK-T\TeX.

Complete DOS, Windows 32 and Macintosh implementations are available as packages on the CD; floppy disk sets will now be supplied only on special request.

Future meetings of UK-T\TeX

It is planned to hold two meetings in 1999, one on May 28th and one, including the AGM, before the end of September. The spring meeting will be on “\TeX~for Windows 32”, and the autumn one on the relationships between \TeX~and L\TeX~on the one hand, and SGML, HTML, XML and related notations on the other. We plan to hold the AGM in London, as this seems likely to maximise attendance.

For further information please contact one of the Committee, or visit the UK-T\TeX~Web site at http://uk.tug.org/.

The Committee of the UK \TeX~Users’ Group