

Anleitung zur T_EX Live CD-ROM, Version 4

Sebastian Rahtz
sebastian.rahtz@oucs.ox.ac.uk
Michel Goossens
m.goossens@cern.ch

März 2000

Inhaltsverzeichnis

1	Einleitung	2
1.1	T _E X-Erweiterungen	2
2	Aufbau und Inhalt der CD	3
2.1	Der TDS Verzeichnisbaum	3
3	Installation und Anwendung der CD unter Unix	4
3.1	Der Einsatz von TeX Live direkt von der CD	5
3.2	Installation des TeX Live-Systems auf die Festplatte	5
3.3	Installation einzelner Pakete von der CD auf eine Festplatte	7
3.4	texconfig	9
4	Installation und Einsatz mit Windows	9
4.1	Direkter Start von der CD-ROM	9
4.2	Installation auf die Festplatte	10
5	Installation von T_EX auf einer nicht unterstützten Rechnerplattform	10
5.1	Voraussetzungen	10
5.2	Konfiguration	10
5.3	Der Aufruf von make	11
5.4	Letzte Schritte zur Konfiguration	11
6	Anleitung zum Web2c-System	12
6.1	Dateisuche mit der Kpathsea-Bibliothek	13
6.2	Dateinamen-Datenbanken	17
6.3	Einstellungen zur Laufzeit	24
7	Andere T_EX-Systeme auf der CD-ROM	25
7.1	emTeX/TDS	25
8	Geschichte und Danksagungen	25
9	Zukünftige Versionen	26

Tabellenverzeichnis

1	Kpathsea Dateitypen	18
---	---------------------	----

1 Einleitung

Diese Anleitung beschreibt die **T_EX Live 5** CD-ROM, die ein T_EX/L^AT_EX-System für Unix und 32-Bit Windows enthält. Dazu zählen T_EX, L^AT_EX 2_ε, METAFONT, MetaPost, Makeindex und BIBT_EX, sowie eine umfassende Auswahl an Makros, Zeichensätzen und Beschreibungen, die gemäß der Standard-T_EX-Verzeichnisstruktur (TDS) auf der CD-ROM abgelegt sind. **T_EX Live** eignet sich somit zur Nutzung mit nahezu jeder T_EX-Installation.

Die T_EX-Programme basieren auf der Web2c-Implementierung (Version 7.3), die sich zum Ziel gesetzt hat, den Gebrauch von T_EX so einfach wie möglich zu gestalten. Web2c benutzt die Kpathsea-Bibliothek von Karl Berry und Olaf Weber für Dateizugriffe, die sich durch hohe Effizienz und besondere Anpaßbarkeit in bestehende Dateistrukturen auszeichnet. Das **T_EX Live-T_EX** kann direkt von der CD-ROM gestartet oder auf die Festplatte installiert werden.

Die meisten T_EX-Komplettsysteme auf der CD-ROM enthalten viele Treiber und unterstützende Programme wie `dvips` (PostScript-Druckertreiber), `xdvi` (X-Windows Bildschirmpreview), `dvilj` (HP LaserJet-Druckertreiber), `lacheck` (L^AT_EX Syntax-Überprüfung), `tex4ht` (T_EX nach HTML-Konvertierung), `dviconcat` und `dviselect`, `dv2dt` und `dt2dv` (`dvi` nach ASCII und umgekehrt), sowie die PostScript-Utilities von Angus Duggan.

1.1 T_EX-Erweiterungen

Unter den T_EX-Systemen der **T_EX Live-CD-ROM** befinden sich drei T_EX-Erweiterungen im Experimentalstadium:

1. ϵ -T_EX, das bei 100 %-iger Kompatibilität zum normalen T_EX einen kleinen, aber mächtigen Satz neuer Befehle bereitstellt, und die Erweiterungen T_EX- \mathbb{X} E_T für Textsatz von rechts nach links, wie er im Arabischen gebraucht wird. Die Dokumentation zu ϵ -T_EX finden Sie in der Datei [texmf/doc/etex/base/etex_man.pdf](#) auf der CD-ROM.
2. pdfT_EX, das optional Acrobat-PDF-Dateien anstelle von `.dvi`-Dateien erzeugt. Die Benutzeranleitung für dieses System finden Sie in der Datei [texmf/doc/pdftex/pdftex-1.pdf](#). In der Datei [texmf/doc/pdftex/samplepdf/samplepdf.tex](#) finden Sie ein Anwendungsbeispiel. Das L^AT_EX-Paket `hyperref` bietet eine Option „`pdftex`“, mit der die Möglichkeiten der aktiven Querverweise von Acrobat PDF zur Verfügung gestellt werden.
3. Ω (Omega), ein T_EX-System, das intern mit 16-Bit-Unicode-Zeichen arbeitet und damit gleichzeitiges Arbeiten mit nahezu allen auf der Welt eingesetzten Zeichenkodierungen erlaubt. Außerdem werden über dynamisch geladene, sogenannte „ Ω Translation Processes“ (OTPs), Mechanismen zur Verfügung gestellt, die beliebige Eingaben vor der Bearbeitung durch T_EX nach bestimmten Regeln umformen. Die Anleitung zu Ω finden Sie auf der CD-ROM in der Datei [texmf/doc/omega/base/doc-1.8.tex](#).

Bei ϵ -T_EX (Version 2.1) handelt es sich um ein stabiles T_EX für den täglichen Einsatz, das allerdings in zukünftigen Versionen neue Funktionalitäten enthalten wird. pdfT_EX (Version 0.14f) und Ω (Version 1.11) befinden sich momentan noch in Entwicklung; die Versionen auf der CD-ROM datieren auf Mitte März 2000.

2 Aufbau und Inhalt der CD

Das Wurzelverzeichnis der CD-ROM enthält folgende wichtige Verzeichnisse:

bin Die ausführbaren Programme des \TeX -Systems, jeweils für die verschiedenen Rechnerplattformen in Unterverzeichnissen zusammengefaßt

tldoc Die Dokumentation zu **\TeX Live**

FAQ Die „Frequently Asked Questions“ in englischer, französischer und deutscher Sprache

info Die Anleitungen zum \TeX -System im GNU-info-Format

man Die Anleitungen zum \TeX -System im Unix-man-Format

source Die Quelldateien aller Programme inklusive der Web2c-Quellen für die \TeX - und METAFONT-Pakete als komprimierte tar-Archive

support Verschiedene Programme, die *nicht* automatisch mitinstalliert werden wie z. B. Musix \TeX , Hilfsprogramme und der GhostScript-Interpreter 5.50 mit Zeichensätzen

texmf Der Verzeichnisbaum für Makros, Zeichensätze und Anleitungen

usergrps Informationen zu \TeX -Anwendervereinigungen

Für Unix-Systeme gibt es zwei verschiedene Installations-Skripten, nämlich `install-cd.sh` und `install-pkg.sh`; diese werden in Abschnitt 3 auf Seite 4 erklärt.

2.1 Der TDS Verzeichnisbaum

Der Verzeichnisbaum `share/texmf` der **\TeX Live**-CD-ROM besteht aus verschiedenen „Sammlungen“ (collections), die wiederum verschiedene „Pakete“ enthalten, von denen die CD-ROM über 400 enthält. Bei einem normalen Installationsvorgang wird der gesamte Inhalt einer Sammlung auf die Festplatte kopiert. Es ist aber auch möglich, einzelne Pakete gezielt auszuwählen und zu installieren. Es gibt folgende Sammlungen:

ams Die Makropakete und Zeichensätze der American Mathematical Society

bibtex Die Stildateien und Verwaltungsprogramme für Bib \TeX

doc Grundlegende Anleitungen und Hilfstexte in verschiedenen Formaten, unter anderem auch HTML und PDF

dvips Hilfsprogramme und Zusätze für den PostScript-Treiber dvips von Tom Rokicki

etex ϵ - \TeX -Dateien

fonts Quelldateien, Metriken, PostScript-Type 1-Zeichensätze und Bitmaps von Zeichensätzen

formats Verschiedene Formate, unter anderem Eplain, Rev \TeX , physxx, texpss, alateX, text1, lollipop, usw.

generic Formatunabhängige Makros

graphics Makropakete zur Ausgabe von Grafik

lang Sprachunterstützung für \TeX -Anwender, die keine englischsprachigen Texte erstellen

latex L^AT_EX mit allen Hilfsprogrammen und den Zusatz-Makropaketen

metapost Extras zu MetaPost

omega Extras zu Ω

pdftex Extras zu pdfT_EX

plain Makropakete für Plain T_EX

systems Ausführbare Programme für Unix- und 32-Bit Windows-Systeme

texlive Grundlegendes Material für die T_EX Live-CD-ROM

Jede Sammlung besteht aus den grundlegenden Paketen (*basic*, 1), den empfohlenen Paketen (*recommended*, 2) und den zusätzlichen Paketen (*other*, 3). Die Sammlung *latex1* ist zum Einsatz von L^AT_EX notwendig, *latex2* sind Pakete, die fast alle Anwender gut gebrauchen können und *latex3* enthält zusätzliche Pakete. Das Verzeichnis *texmf/tpm* enthält eine Liste der Dateinamen (zur Verwendung durch die Installationsprogramme), die die Pakete jeweils enthalten.

3 Installation und Anwendung der CD unter Unix

Sie können die T_EX Live-CD-ROM auf drei Arten benutzen:

1. Sie können die CD-ROM als eigenes Dateisystem mounten, Ihre PATH-Variable anpassen und T_EX direkt von der CD-ROM aus starten. Dadurch brauchen Sie nur wenig Speicher auf der Festplatte und können sofort auf alle Programme zugreifen. Obwohl dieses Verfahren nicht gerade für hohe Arbeitsgeschwindigkeit sorgt, kann man z. B. auf einem PC unter Linux gut damit leben.
2. Sie können das Gesamtsystem oder Teile davon auf Ihrer Festplatte installieren. Dieses Verfahren eignet sich am Besten für Personen, die mindestens 10 MByte, besser jedoch 100 MByte Speicher auf der Festplatte für T_EX freimachen können.
3. Sie können einzelne Pakete in Ihr bestehendes T_EX-System integrieren, egal, ob es ein T_EX Live-System oder ein anderes ist.

Die folgenden Abschnitte beschreiben jedes dieser drei Verfahren ausführlich zum Nachvollziehen.

Achtung: Diese CD-ROM wurde im ISO 9660 (High Sierra) Format mit Rock Ridge und Joliet Erweiterungen produziert. Damit Ihr Unix-Rechner die CD-ROM optimal nutzen kann, muß Ihr System diese Erweiterungen unterstützen. Schlagen Sie in der Systemdokumentation des *mount*-Befehls nach, um zu erfahren, ob Ihr System diese Erweiterungen unterstützt. Falls Sie die CD-ROM in einem lokalen Netzwerk einsetzen, suchen Sie sich einen Rechner aus, der die Erweiterungen unterstützt und mounten Sie die anderen Rechner auf diesen.

Linux, FreeBSD, Sun, SGI, AIX 4 und DEC Alpha-Systeme haben normalerweise keine Schwierigkeiten, die CD-ROM zu benutzen. Wenn Sie ein anderes System einsetzen und die CD-ROM erfolgreich einbinden konnten, erbitten wir von Ihnen detaillierte Anweisungen, um diese Anleitung weiter verbessern zu können.

Der Rest der Anleitung geht davon aus, daß Sie die CD-ROM erfolgreich unter Verwendung der Rock Ridge und Joliet Erweiterungen gemountet haben.

3.1 Der Einsatz von TeX Live direkt von der CD

Das Web2c-System ist so aufgebaut, daß Sie zum ordnungsgemäßen Funktionieren lediglich das korrekte Unterverzeichnis unter bin im Wurzelverzeichnis der CD-ROM an den Systempfad PATH anhängen müssen. Alle Konfigurationsdateien und Daten werden automatisch gefunden. Folgende Tabelle gibt die Zuordnung zwischen dem Unterverzeichnis in bin und dem entsprechenden Zielsystem an.

DEC Alphaev5 OSF 4.0d	alphaev5-osf4.0d
HP9000 HPUX 10.10	hppa2.0-hpux10.20
Intel x86 with GNU/Linux	i386-linux
Intel x86 with FreeBSD ELF 3.4	i386-freebsd
SGI IRIX 6.5	mips-irix6.5
IBM RS 6000 AIX 4.2.*	rs6000-aix4.2.1.0
Sun Sparc Solaris 2.7	sparc-solaris2.7
Windows 9X/2000/NT	win32

Vielleicht fragen Sie sich schon, was passiert, wenn Sie Konfigurationsdateien ändern oder neue Zeichensätze erzeugen. Da die CD-ROM nicht beschreibbar ist, könnte das ja schiefgehen. Um dieses Problem zu umgehen, können Sie auf einem parallel angelegten, beschreibbaren Dateisystem diese geänderten und neuen Dateien ablegen. Dieser zweite TeX-Dateibaum wird *vor* dem Dateibaum der CD-ROM durchsucht, so daß in jedem Fall die geänderten Daten benutzt werden. Für diesen zweiten Dateibaum ist das (nicht existierende) Startverzeichnis `texmf-localconfig` auf der CD-ROM vorgesehen, so daß Sie mit der Umgebungsvariablen `VARTEXMF` unbedingt ein beliebiges anderes Verzeichnis einstellen *müssen*.

Benutzer, die eine Bourne- oder Korn-Shell kompatible Shell wie *sh* oder *bash* einsetzen und die TeX Live-CD-ROM auf einem Linux-Rechner über den Befehl

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

auf das Verzeichnis `/cdrom` mounten, könnten in der Shell-Konfigurationsdatei `.profile` über folgende Befehle das TeX Live-System aktivieren:

```
PATH=/cdrom/bin/i386-linux:$PATH
export PATH
VARTEXMF=/usr/TeX.local
export VARTEXMF
```

Wenn Sie sich bezüglich des `mount`-Befehls oder des Verzeichnisses nicht ganz sicher sind, bitten Sie Ihren Systemverwalter um Unterstützung.

Notwendige Konfigurations- und Steuerdateien werden auf Ihre Festplatte kopiert, sobald Sie sie das erste Mal benötigen. Sie sollten allerdings lieber gleich das Skript `texconfig` aufrufen, die notwendigen Einstellungen vornehmen und das TeX-System testen.

3.2 Installation des TeX Live-Systems auf die Festplatte

Zum Installieren auf eine Festplatte unter Unix sollten Sie zunächst die CD-ROM mounten (siehe vorheriger Abschnitt). *Wechseln Sie dann ins Wurzelverzeichnis der CD-ROM* und starten das Installations-Skript:

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

(Auf manchen Unix-Systemen müssen Sie anstelle von `sh` eine der Shells `sh5` oder `bash` verwenden.) Das Installations-Skript liest daraufhin die Beschreibung der Sammlungen und Pakete von der CD-ROM, versucht zu ermitteln, welches Rechnersystem Sie einsetzen und gibt folgenden Text aus:

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

Danach wird die Installations-Befehlsmaske (Abbildung 1) eingeblendet, über die Sie vier Parameter einstellen können:

1. Zielsystem für die Installation
2. Auswahl der zu installierenden Sammlung (collection) bei gegebener Stufe (*basic*, *recommended* oder *other*)
3. Das Zielverzeichnis auf Ihrer Festplatte
4. Einige Laufzeit-Einstellungen

Die Auswahl der Parameter erfolgt durch Eingabe eines Buchstabens oder einer Ziffer, gefolgt von der Eingabetaste. In der abgebildeten Beispielmaste wurde ein Linux ELF-System entdeckt, alle Sammlungen wurden auf der Stufe *recommended* (empfohlen) ausgewählt (Voreinstellung) und als Zielverzeichnis wurde `/usr/local` angenommen (Voreinstellung). Der benötigte Festplattenspeicher für die ausgewählten Parameter wird ebenfalls angezeigt. Wenn Sie die Installation im vorgeschlagenen Umfang vornehmen, benötigen Sie ungefähr 100 MByte Festplattenkapazität. Die Minimalinstallation benötigt lediglich ungefähr 10 MByte. Diese Minimalinstallation können Sie durch Hinzunahme der von Ihnen eingesetzten Pakete für eine optimale Arbeitsgeschwindigkeit verbessern.

Im Zielverzeichnis werden die ausführbaren Programme im Unterverzeichnis `bin` abgelegt, die benötigten Daten im Unterverzeichnis `texmf`.

Zurück zur Hauptmaske. Die Laufzeiteinstellungen (Befehl 0) ermöglichen Ihnen, das schon erwähnte, beschreibbare, parallele \TeX -Verzeichnis zu erzeugen (*alternate directory*) bzw. eine Installation auf Festplatte vorzunehmen, die für die Verzeichnisse *man* und *info* anstelle von Kopien der Dateien Verweise auf die CD-ROM erzeugt. In diesem Fall muß natürlich wie bei einem \TeX Live-System die CD-ROM ständig eingelegt und gemountet sein. Außerdem benötigen Sie hierfür natürlich Systemverwalterrechte („*root*“).

Wenn Sie die Installationsstufen der Sammlungen mit Befehl `<C>` ändern, wird Abbildung 2 eingeblendet. Sie sehen alle angebotenen Sammlungen, die gewählten Installationsstufen und die jeweils benötigte Festplattenkapazität. Sie können die Installationsstufen jeweils einzeln einstellen, oder alle Sammlungen mit dem gleichen Wert belegen. Ein Beispiel sehen Sie in Abbildung 3.

Wenn Sie alle Einstellungen vorgenommen haben, kehren Sie zur Hauptmaske zurück und starten die Installation über Befehl `I`. Mittels Ihrer Angaben der zu installierenden Sammlungen und der Indexdateien auf der CD-ROM wird eine Liste der zu kopierenden Dateien erzeugt. Nachdem diese Dateien auf Ihrer Festplatte kopiert wurden, wird bei einem Gesamtsystem dessen Initialisierungssequenz gestartet (Format-Dateien erzeugen usw). Danach müssen Sie lediglich das richtige `bin`-Verzeichnis der \TeX -Installation in Ihren Pfad aufnehmen, um mit \TeX arbeiten zu können. Wenn Sie die ausführbaren Dateien noch eine Ebene in der Verzeichnishierarchie nach oben kopieren wollen, z. B. von `/usr/local/bin/alpha-osf3.2` nach `/usr/local/bin`, müssen Sie anschließend die Datei `texmf/web2c/texmf.cnf` editieren und die Variable (siehe Anhang 10) von

```
TEXMFMAIN = $SELFAUTOPARENT
```

zu

```
TEXMFMAIN = $SELFAUTODIR
```

abändern.

Wenn die Binärdateien in ein vollkommen anderes Verzeichnis installiert werden sollen, müssen Sie die Environment-Variable `TEXMFMAIN` auf dieses Verzeichnis setzen und die Variable `TEXMFCNF` auf `$TEXMFMAIN/texmf/web2c`.

```

===== TeX Live installation procedure =====
=> 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
<C> collections:    24 out of 35, disk space required: 193176 kB
<S> systems:       1 out of 8, disk space required:   8355 kB
                    total disk space required: 201531 kB
<L> install level (1: basic, 2: recommended, 3: all): 2
<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
<I> start installation, <H> help, <Q> quit

Enter command:

```

Abbildung 1: Installationsmaske, Hauptbildschirm

	name	selection	size
<1>	bibtex	[recommended]	7597 kB
<2>	doc	[recommended]	21152 kB
<3>	dvips	[recommended]	430 kB
<4>	etex	[recommended]	102 kB
<5>	fonts	[recommended]	51447 kB
<6>	formats	[recommended]	14651 kB
<7>	generic	[recommended]	459 kB
<8>	graphics	[recommended]	9674 kB
<9>	lang	[recommended]	19618 kB
<U>	latex	[recommended]	23429 kB
<V>	metapost	[recommended]	1443 kB
<W>	omega	[recommended]	4986 kB
<X>	pdftex	[recommended]	471 kB
<Y>	plain	[recommended]	1113 kB
<Z>	texlive	[recommended]	10155 kB
		SUM:	166829 kB

```

=====
global commands: select <N>one / <B>asic / R<E>commended / <A>ll
                  for all collections
<R> return to platform menu
<Q> quit

```

Abbildung 2: Installationsmaske, Auswahl der Sammlungen

3.3 Installation einzelner Pakete von der CD auf eine Festplatte

Sie können die **TeX Live**-CD-ROM auch dazu benutzen, ein bestehendes **TeX**-System auf den neuesten Stand zu bringen bzw. die Daten einer bestehenden Installation zu ergänzen. Das im vorigen Abschnitt vorgestellte Installations-Skript ist nur für eine Erstinstallation gedacht. Für den gerade angesprochenen Zweck dient ein anderes Installations-Skript, nämlich `install-pkg.sh`. Zunächst sollten Sie wiederum die CD-ROM mounten, *in das Wurzelverzeichnis wechseln* und das Skript durch

```
>> sh install-pkg.sh Optionen
```

```

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:

```

Abbildung 3: Auswahl der Installationsstufe (hier: Fonts)

starten. Das Skript kann über neun Kommandozeilenparameter gesteuert werden. Die ersten vier dienen zur Auswahl des zu installierenden Pakets, zur Auswahl der Sammlung (z. B. `ams2`), zur Angabe des Verzeichnisses, auf das die CD-ROM gemountet ist und zur Angabe des Verzeichnisses, das die Dateilisten der Pakete enthält. (Normalerweise werden die letzten beiden automatisch gesetzt.)

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

Die eigentliche Aktion wird durch vier weitere Parameter gesteuert. Die ersten beiden verhindern die Installation der Dokumentation bzw. der Quelldateien eines Pakets. Die dritte verhindert den Aufruf von `mktexlsr` bei Abschluß der Installation, das die Datenbank der Verzeichnisstruktur neu aufbauen würde. Der vierte Parameter dient zur Anzeige der zu installierenden Dateien, ohne daß die Installation tatsächlich durchgeführt wird.

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

Außerdem können Sie statt einer Installation der Dateien ein tar-Archiv im angegebenen Verzeichnis anlegen lassen.

- `--archive=name`

Wenn wir beispielsweise wissen wollen, welche Dateien mit dem `fancyhdr`-Paket installiert werden würden, müßten wir folgenden Befehl eingeben, der die entsprechende Ausgabe zur Folge hat:

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

```

texmf/doc/latex/fancyhdr/fancyhdr.dvi
texmf/doc/latex/fancyhdr/fancyhdr.tex
texmf/lists/latex3/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

```

Weitere Beispiele:

- Installation des L^AT_EX-Pakets natbib:

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

- Installation des L^AT_EX-Pakets alg ohne Quelldateien und ohne Dokumentation:

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

- Installation aller Pakete aus der Sammlung Plain T_EX „other“:

```
>> sh install-pkg.sh --collection=plain3
```

- Erzeugung eines tar-Archivs der Dateien des PStricks-Pakets im Verzeichnis /tmp:

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

3.4 texconfig

Nachdem das Installationsprogramm die Dateien kopiert hat, können Sie das texconfig-Programm aufrufen, um das T_EX-System an Ihre lokalen Gegebenheiten anzupassen. Sie können das Programm auch zu jedem beliebigen anderen Zeitpunkt aufrufen. Falls das dialog-Programm verfügbar ist, startet texconfig mit einem maskenbasierten Kommandobildschirm, sonst wird es über Kommandozeilenparameter gesteuert. Sie sollten für sämtliche Wartungsarbeiten wie Installation neuer Drucker oder Neuaufbau der Dateinamen-Datenbank texconfig verwenden. Es gibt für beide Betriebsmodi Hilfstexte, die Ihnen die Möglichkeiten von texconfig erklären.

4 Installation und Einsatz mit Windows

Dieser Abschnitt gilt nur für 32-Bit Windows-Systeme.

Neben der 32-Bit-Unterstützung muß Ihr Windows die Microsoft Joliet-Erweiterungen für CD-ROMs unterstützen. Sie können das ganz einfach testen, indem Sie sich den Inhalt der CD-ROM im Explorer anzeigen lassen. Sehen Sie lange Dateinamen in Klein-/Großschreibung, ist Ihr System für **T_EX Live** geeignet, ansonsten nicht.

Die 32-Bit Windows-T_EX-Systeme enthalten einen Previewer, der sich in der Bedienung am etablierten xdv anlehnt, nämlich Windvi. Die Anleitung finden Sie unter texmf/doc/html/windvi/windvi.html.

4.1 Direkter Start von der CD-ROM

Sie können – auf Kosten der Ausführungsgeschwindigkeit – alle zu T_EX gehörigen Programme direkt von der CD-ROM starten und auf alle Makros und Zeichensätze direkt über die CD-ROM zugreifen. Dazu müssen Sie nur das CD-ROM-Verzeichnis bin/win32 mit in den Pfad aufnehmen, am einfachsten durch Einfügen entsprechender Anweisungen in AUTOEXEC.BAT oder das Kontrollfeld System in den Einstellungen. Jetzt können Sie die T_EX-Programme durch Aufruf in einer DOS-Box starten oder für bequemes, menü- und icongesteuertes Arbeiten, den Shareware-Editor WinEdt heranziehen.

4.2 Installation auf die Festplatte

Zur Installation legen Sie die CD ein und warten auf den automatischen Start oder – falls Sie diese Eigenschaft nicht aktiviert haben – Sie starten das Programm `setup.exe` im Verzeichnis `setupw32/setup`, das ebenfalls mit den Dateilisten auf der CD arbeitet. Sie können die Installationsstufen für die Sammlungen (für eine Beschreibung lesen Sie bitte Abschnitt 2.1) einstellen und bei mangelnder Festplattenkapazität die Anleitungen bzw. Quelldateien von der Installation ausschließen. Außerdem müssen Sie die beiden Verzeichnisse für die Hauptinstallation und für die lokal angepaßten Teile des $\text{T}_{\text{E}}\text{X}$ -Systems angeben. Ferner haben Sie die Möglichkeit, die Shareware- $\text{T}_{\text{E}}\text{X}$ Shell WinEdt und die PostScript-Emulation GhostScript zu installieren.

Beachten Sie bitte, daß es sich bei WinEdt um Shareware handelt und registrieren Sie Ihre Kopie beim Autor, wenn Sie sie nach der Evaluierungsphase einsetzen.

Unter Windows kann die Clustergröße von DOS-Partitionen die benötigte Festplattenkapazität radikal beeinflussen. Es gibt Hunderte von einzelnen Dateien im $\text{T}_{\text{E}}\text{X}$ -System, dadurch wird bei großen Clustern sehr viel Plattenplatz unnötig belegt. Bei einer Partitionsgröße von über 1 Gbyte beträgt die Clustergröße 32 Kbyte, somit auch die Dateimindestgröße. Speziell bei `.tfm`-Dateien mit Größen von wenigen KByte werden dadurch enorme Speichermengen verschenkt (Der sogenannte Slack).

Nach erfolgter Installation müssen Sie Windows neu starten. Danach rufen Sie die Programme in einer DOS-Box oder über WinEdt-Icons und -Menüs auf.

5 Installation von $\text{T}_{\text{E}}\text{X}$ auf einer nicht unterstützten Rechnerplattform

Wenn Sie eine Rechnerplattform einsetzen, für die wir keine Binärdateien zur Verfügung gestellt haben, müssen Sie $\text{T}_{\text{E}}\text{X}$ und die zugehörigen Programme mit Hilfe der Quellcodes selbst übersetzen. Das ist gar nicht so schwer, wie es klingt. Alles, was Sie brauchen, finden Sie auf der CD-ROM im Verzeichnis `source`.

Zunächst sollten Sie jedoch die von $\text{T}_{\text{E}}\text{X}$ benötigten Daten durch eine Installation ohne Binärdateien auf die Festplatte kopieren bzw. durch Mounten der CD-ROM verfügbar machen.

5.1 Voraussetzungen

Zum Übersetzen von $\text{T}_{\text{E}}\text{X}$ und allen weiteren benötigten Programmen benötigen Sie ca. 100 Mbyte Festplattenkapazität. Dazu kommen ein ANSI-C-Compiler, ein `make` sowie ein Scanner- und Parsergenerator. Die GNU-Programme (`gcc`, GNU `make`, `m4`, `flex` und `bison`) sind praxiserprobt und auf den meisten Plattformen verfügbar. `gcc-2.7.*`, `flex-2.4.7` und GNU `make-3.72.1` oder neuere Versionen sollten anstandslos funktionieren. Eventuell können Sie auch andere C-Compiler oder `make`-Programme einsetzen, doch dann sollten Sie sich wirklich gut mit Ihrem System auskennen, wenn es Schwierigkeiten während des Übersetzungsprozesses gibt. Der Befehl `uname` muß einen gültigen Namen liefern.

5.2 Konfiguration

Zunächst entpacken Sie das komprimierte Archiv in `source` auf Ihre Festplatte und wechseln in dieses Verzeichnis. Überlegen Sie sich jetzt einen Pfad, unter dem das $\text{T}_{\text{E}}\text{X}$ -System installiert werden soll. Natürlich sollte das derselbe sein, den Sie schon zum Kopieren der Daten verwendet haben (z.B. `/usr/local` oder `/usr/local/TeX`).

Die automatische Systemerkennung und -anpassung wird durch `configure` mit einem Aufruf wie diesem gestartet:

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

Bei „prefix“ handelt es sich um das Verzeichnis, in dem sich die bereits kopierten **T_EX Live**-Dateien befinden; dieses Verzeichnis wird weiter strukturiert: (\$TEXDIR ist das Installationsverzeichnis):

\$TEXDIR/man	Anleitungen im Unix-man-Format
\$TEXDIR/share/texmf	T _E X-Daten mit Zeichensätzen, Makros, usw.
\$TEXDIR/info	Anleitungen im GNU-info-Format
\$TEXDIR/bin/\$PLATFORM	ausführbare Programme

Wenn Sie wollen, können Sie ‘share/’ als Namensbestandteil von texmf weglassen, da sowohl \$TEXDIR/share/texmf als auch \$TEXDIR/texmf automatisch von configure erkannt werden. Sollten Sie ein anderes Verzeichnis vorziehen, müssen Sie dieses configure mit Kommandozeilenparameter --datadir mitteilen.

Wenn Sie die erkannte Plattform nicht als \$PLATFORM im Pfad der ausführbaren Programme haben wollen, sondern z. B. direkt in \$TEXDIR/bin installieren wollen, verwenden Sie die Option --disable-multiplatform.

Zum Weglassen spezieller T_EX-Systeme wie Ω oder ε-T_EX gibt es eine Reihe von Optionen, die Sie durch Eingabe von ./configure -help zur Anzeige bringen.

5.3 Der Aufruf von make

Vergewissern Sie sich, daß die Shell-Variable noclobber nicht gesetzt ist und geben

```
>> make world
```

ein. Jetzt ist Zeit für eine Tasse Tee oder Kaffee. . .

Eine gute Idee ist auch das Speichern der Ausgaben beim Übersetzen durch Anlegen eines Protokolls:

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

Nehmen Sie nicht fälschlicherweise nach dem Durchlauf von make an, daß alles glattgegangen ist, sondern vergewissern sich durch Kontrolle des Protokolls. (GNU make zeigt Fehler z. B. immer mit der Zeichenkette „Error:“ an.) Außerdem sollten Sie sich vergewissern, daß alle ausführbaren Programme auch erzeugt wurden:

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

Wenn Sie als Ergebnis 214 erhalten, ist alles vorhanden. Der Aufruf von make world ist übriggend äquivalent zu make all install strip.

Wenn für den Aufruf von make install besondere Systemverwalterrechte notwendig sein sollten, können Sie das Kommando auch in zwei Durchgängen absetzen:

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

5.4 Letzte Schritte zur Konfiguration

Nehmen Sie das Verzeichnis der gerade erzeugten Programme (z. B. /usr/local/TeX/bin/mips-sgi-irix6.5) in Ihre PATH-Variable auf; genauso verfahren Sie mit den Variablen MANPATH und INFOPATH für die neuen Anleitungen, z. B. \$TEXDIR/man und \$TEXDIR/info.

Mit texconfig können Sie Voreinstellungen für Trennmuster, Papiergröße, Befehle zum Drucken, bevorzugten METAFONT-Modus usw. vornehmen. Das Programm arbeitet interaktiv oder Sie erhalten eine Anleitung durch Eingabe von

>> texconfig help

Falls Sie z. B. nicht die deutsche Papiergröße „A4“ verwenden, sondern stattdessen „lettersize“ können Sie dies durch

>> texconfig dvips paper letter

>> texconfig xdvi paper us

einstellen.

6 Anleitung zum Web2c-System

Web2c besteht aus einer Reihe von Programmen, die zusammen ein komplettes T_EX-System darstellen. Dazu gehören z. B. T_EX selbst, METAFONT, MetaPost, BIBT_EX, usw. Die erste Implementierung eines T_EX-Systems in der Programmiersprache C stammt von Tomas Rokicki und datiert zurück in das Jahr 1987. Rokicki benutzte als Basis sog. Change-Files unter Unix, die von Howard Trickey und Pavel Curtis entwickelt wurden. Tim Morgan hat dieses System, für das der Name Web-to-C eingeführt wurde, gepflegt. 1990 hat Karl Berry mit Unterstützung Dutzender Helfer die Weiterentwicklung übernommen und 1997 an Olaf Weber weitergegeben. Die neueste Fassung von Web2c ist Version 7.3, die im März 1999 freigegeben wurde. Auf diesem T_EX-System basiert die **T_EX Live** CD-ROM.

Web2c 7.2 läuft unter Unix, Windows 3.1, 9x/NT, DOS und auf weiteren Betriebssystemen. Es benutzt die Original-Quelldateien von Donald E. Knuth und weitere in der Sprache web entwickelte Programme als Basis und übersetzt diese in C-Quellcode. Darüberhinaus bietet das System viele Makros und Funktionen zur Nutzung der originalen T_EX-Software. Hier eine Liste der grundlegenden Programme eines T_EX-Systems:

bibtex Verwaltung von Bibliographien

dmp troff nach MPX Konverter (MetaPost-Bilder)

dvicopy Umwandlung von virtuellen Zeichensätzen in DVI-Dateien

dvitomp DVI nach MPX Konverter (MetaPost-Bilder)

dvitype Textanzeige aus DVI-Dateien

gftodvi Erzeugung von Prüfdrucken für Zeichensätze

gftopk Packen von Zeichensätzen

gftype Textanzeige von Zeichensätzen

makempx Anzeige von MetaPost-Beschriftungen

mf Zeichensatzerzeugung

mft Formatierte Ausgabe von METAFONT-Quellen

mpost METAFONT-ähnliches Grafikprogramm

mpto Extrahierung von MetaPost-Beschriftungen

newer Vergleich von Dateierstellungsdaten

patgen Erzeugung von Trennmustern

`pktofg` Entpacken von Zeichensätzen
`pktype` Anzeige gepackter Zeichensätze
`pltotf` Umwandlung von Property-Listen in TFM-Dateien
`pooltype` Anzeige der Bildschirmtexte eines WEB-Programms
`tangle` Konverter web nach Pascal
`tex` T_EX-Programm
`tftopl` Umwandlung TFM-Datei in Property-Liste
`vftovp` Virtueller Zeichensatz in Property-Liste
`vptovf` Property-Liste in virtuellen Zeichensatz
`weave` web-Code als T_EX-Dokumentation

Die genaue Funktionsweise und die möglichen Parameter sind der Beschreibung der jeweiligen Pakete bzw. der Web2c-Dokumentation zu entnehmen. Trotzdem wird Ihnen ein Überblick über Zusammenspiel und Funktionsweise der Web2c-Programme sicher helfen, besser mit dem System zurechtzukommen.

Zunächst verstehen alle Programme die grundlegenden Parameter der GNU-Software:

`-help` kurzer Hilfstext
`-verbose` ausführliche Ausgaben zum Programmablauf
`-version` Ausgabe der Versionsnummer

Die Programme des Web2c-Systems benutzen zum Lokalisieren der benötigten Dateien im Dateisystem die Kpathsea-Bibliothek. Diese Bibliothek optimiert und beschleunigt den Suchprozeß im Dateisystem. Ihre Arbeitsweise wird durch einige Umgebungsvariablen und eine Konfigurationsdatei gesteuert. Web2c kann mehr als einen Dateibaum gleichzeitig verwalten und ermöglicht somit die schon beschriebene T_EX Live-Installation unter Verwendung der CD-ROM mit Ablage modifizierter Konfigurationsdateien und zusätzlicher Zeichensätze in einem zweiten Dateibaum. Die Suche nach Dateien wird durch Analyse der Datei `ls-R` beschleunigt, die in jedem Wurzelverzeichnis eines T_EX-Dateibaums vorhanden ist. Sie enthält für jede Datei die genaue Position im Dateibaum relativ zum Wurzelverzeichnis.

6.1 Dateisuche mit der Kpathsea-Bibliothek

Wir beschreiben zunächst den grundlegenden Suchmechanismus der Kpathsea-Bibliothek.

Ein *Suchpfad* ist eine durch Kommata oder Semicola getrennte Liste von *Pfadkomponenten*, die üblicherweise Verzeichnisnamen darstellen. Ein Suchpfad kann sich aus vielen Komponenten zusammensetzen. Die Suche nach einer Datei `my-file` über den Suchpfad `./dir` bewirkt, daß Kpathsea jede Komponente nacheinander überprüft, also zunächst `./my-file` und dann `/dir/my-file`. Als Ergebnis wird entweder die erste gefundene Datei oder eine Liste aller passenden Dateien geliefert.

Um auf allen Dateisystemen effizient zu arbeiten, verwendet Kpathsea ggf. andere Datei/Verzeichnis-Separatoren als `:"` und `/"`.

Beim Überprüfen einer Pfadkomponente *p* überprüft Kpathsea zunächst, ob eine Dateinamen-Datenbank (siehe auch Dateinamen-Datenbank auf Seite 17) für die Pfadkomponente zuständig ist, d. h. beispielsweise steht die Datenbank in einem Verzeichnis, das im Pfad vor der zu überprüfenden Komponente *p* steht. In diesem Fall wird zur Bestimmung der Position der gesuchten Datei die Datenbank herangezogen.

Nur, wenn keine passende Datenbank existiert, bzw., wenn die Datei nicht in der Datenbank gefunden wird, durchsucht Kpathsea das Dateisystem. Diese zeitaufwendige Suche kann über Spezifikation der Pfadkomponente p mit Präfix “!” unterbunden werden. Zur Suche generiert Kpathsea eine Liste der Verzeichnisse, die im Pfadelement enthalten sind, und durchsucht jedes dieser Verzeichnisse nach der gesuchten Datei.

Für Dateien kann auch ein Schalter „file must exist“ gesetzt werden (Datei muß vorhanden sein). Wenn dieser Schalter nicht gesetzt ist und beispielsweise über das $\text{T}_{\text{E}}\text{X}$ -Kommando `openin` eine VF-Datei wie `cmr10.vf` gelesen werden soll, wäre es falsch, nach dieser Datei zu suchen, weil es sie gar nicht gibt. Speziell für neu installierte VF-Dateien sollten Sie also unbedingt die `ls-R`-Datenbank aktualisieren, weil die Dateien sonst nicht gelesen werden und kein Fehler erzeugt wird. Dieser Vorgang wiederholt sich für jede Komponente eines Suchpfades: zunächst wird die Datenbank überprüft, danach ggf. das Dateisystem. Wird die Datei gefunden, stoppt die Suche (normalerweise) und der komplette Pfad zur gesuchten Datei wird als Ergebnis zurückgegeben.

Außer Verzeichnisnamen dürfen Pfadkomponenten für Kpathsea folgende Elemente enthalten: (verschachtelte) Vorgaben, Umgebungsvariablen, Werte aus der Konfigurationsdatei, Home-Verzeichnisse von Benutzern und Startverzeichnisse für eine rekursive Suche. Diese Elemente werden vor einer Dateisuche von Kpathsea in gewöhnliche Verzeichnis- oder Dateinamen expandiert. Diese Expansion wird in den folgenden Abschnitten erklärt, und zwar genau in der Reihenfolge, wie die Elemente auch von Kpathsea bearbeitet werden.

Beachten Sie, daß Kpathsea bei absoluten und explizit relativen Komponenten, d. h., die Komponente beginnt mit den Zeichen “/”, “. /” oder “. . /”, nur überprüft, ob die Datei existiert.

6.1.1 Bestandteile von Pfadkomponenten

Ein Suchpfad kann aus vielen verschiedenen Bestandteilen aufgebaut werden. In der Reihenfolge, wie Kpathsea sie auswertet, sind das:

1. Eine benutzerdefinierte Umgebungsvariable, z.B. `TEXINPUTS`. Wird der Variablen ein Punkt und ein Programmname nachgestellt, wie beispielsweise bei `TEXINPUTS.latex`, hat diese Form Vorrang vor den „gewöhnlichen“ Variablen.
2. Einträge aus programmspezifischen Konfigurationsdateien, beispielsweise eine Zeile “`S /a:/b`” in der Konfigurationsdatei `conf ig.ps` zum Programm `dvips`.
3. Die Einträge aus der Kpathsea-Konfigurationsdatei `texmf.cnf`, z. B. “`TEXINPUTS=/c:/d`” (siehe folgenden Text).
4. Die Einstellung beim Übersetzen der Programme.

Unter Verwendung der Parameter zur Fehlersuche können Sie diese Werte für einen Suchpfad auch anzeigen lassen. (Siehe dazu den Abschnitt Fehlersuche auf Seite 22.)

6.1.2 Konfigurationsdateien

Die Kpathsea-Bibliothek liest zur Laufzeit die Konfigurationsdateien namens `texmf.cnf`. Der zugehörige Suchpfad zum Auffinden dieser Konfigurationsdateien steht in der Umgebungsvariablen `TEXMFCNF` (die Voreinstellung ist `texmf/web2c`). Kpathsea liest *alle* Dateien namens `texmf.cnf`, die es in diesem Suchpfad findet. Die zuerst gelesenen Definitionen haben dabei Vorrang vor nachher gelesenen Werten. Wenn der Suchpfad auf `.$TEXMF` steht, überschreiben die Einstellungen in `./texmf.cnf` diejenigen in `.$TEXMF/texmf.cnf`.

Im folgenden wird die Syntax der `texmf.cnf`-Datei angegeben. Konsultieren Sie zum besseren Verständnis beim Lesen die auf der CD-ROM enthaltene Konfigurationsdatei, die ab Seite 27 im Anhang der Anleitung abgedruckt ist.

- Kommentare beginnen mit einem „%“ Kommentare beginnen mit einem “%” und erstrecken sich bis zum Zeilenende.
- Leerzeilen werden überlesen.
- Ein \-Zeichen am Zeilenende faßt die aktuelle mit der folgenden Zeile zusammen. Leerraum in der Folgezeile wird *nicht* überlesen.
- Jede andere Zeile muß folgenden Aufbau haben:

Variable [.Programname] [=] Wert

wobei das „=“-Zeichen und umgebender Leerraum weggelassen werden dürfen.

- Der Name von *Variable* darf alle Zeichen enthalten außer Leerraum, “=“ und “.“. Am besten verwenden Sie ausschließlich die Zeichen “A-Za-z_”.
- Wenn das Suffix *.Programname* angegeben wird, gilt die Variable nur für das entsprechende Programm *Programname* oder *Programname.exe*. Auf diese Weise können beispielsweise verschiedene T_EX-Formate mit unterschiedlichen Suchpfaden arbeiten.
- *Wert* darf alle Zeichen außer % und @ enthalten. Die Einschränkung der Werte auf bestimmte Programme über ein Suffix ist nicht zulässig. Ein “;“-Zeichen in *Wert* wird unter Unix in ein “:“-Zeichen umgewandelt. Dadurch ist die Verwendung der gleichen Konfigurationsdateien für Unix und DOS/Windows-Systeme möglich.
- Die Definitionen werden komplett eingelesen, bevor eine Expansion stattfindet. Dadurch können Sie Variablen vor Ihrer Definition schon benutzen.

Der Ausschnitt einer Konfigurationsdatei demonstriert diese Möglichkeiten.

```

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;}//

```

6.1.3 Expansion von Pfadkomponenten

Kpathsea verwendet in Suchpfaden ähnliche Zeichen und Konstrukte wie Unix-Shells. Als Beispiel wird die Definition `~$USER/{foo,bar}//baz` in alle Pfade im Home-Verzeichnis expandiert, die unter den Verzeichnissen `foo` oder `bar` liegen und ein Unterverzeichnis oder eine Datei mit dem Namen `baz` enthalten. Der Expansionsmechanismus wird im folgenden erklärt.

6.1.4 Expansion der Voreinstellungen

Wenn der höchstpriorisierte Suchpfad (siehe „Expansion von Pfadkomponenten“ auf Seite 14) einen zusätzlichen (vorangestellten, nachgestellten oder verdoppelten) Doppelpunkt enthält, wird an dieser Stelle der Suchpfad eingefügt, der als nächstes in der Hierarchie folgt. Auch bei diesem gilt dieselbe Regel. Wenn beispielsweise die Umgebungsvariable

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

gesetzt wird (hier: C-Shell) und in `texmf.cnf` die Variable `TEXINPUTS` folgenden Wert erhält

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

dann lautet der Suchpfad schließlich:

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

Da es sinnlos wäre, denselben Pfad mehrfach einzufügen, wird die Ersetzung nur einmal vorgenommen, und zwar in der Reihenfolge vorne, hinten und Mitte. Mehrfach verdoppelte Doppelpunkte bleiben unverändert.

6.1.5 Expansion geschweifter Klammern

Die Expansion geschweifter Klammern ist zur Definition mehrerer \TeX -Hierarchien sehr nützlich. Beispielsweise wird $v\{a,b\}w$ zu $vaw:vbw$. Verschachtelungen sind dabei erlaubt. Diese Technik kann dazu benutzt werden, durch Zuweisung an $\$TEXMF$ verschiedene \TeX -Hierarchien einzuführen. Als Beispiel finden Sie in `texmf.cnf` folgende Definition (Zeile 29):

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

Eine Anwendung wie

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

führt dann dazu, daß erst im aktuellen Verzeichnis gesucht wird, dann der gesamte Dateibaum $\$HOME-TEXMF/tex$ und $\$TEXMFLOCAL/tex$ (auf der Festplatte) und schließlich der gesamte Dateibaum $\$VAR-TEXMF/tex$ sowie $\$TEXMFMAIN/tex$ (nur in der Datenbank `ls-R`) durchsucht wird. Dadurch kann man bequem zwei parallel installierte \TeX -Hierarchien durchsuchen, beispielsweise eine unveränderliche auf CD-ROM und eine dynamisch angepaßte auf Festplatte, in der neue Programmversionen und zusätzliche Zeichensätze installiert werden. Durch Verwendung der Variablen $\$TEXMF$ in allen Definitionen wird grundsätzlich zuerst der neuere Dateibaum durchsucht.

6.1.6 Expansion von Unterverzeichnissen

Zwei oder mehrere aufeinanderfolgende Vorwärtsschrägstriche in einer Pfadkomponente, die auf einen Verzeichnisnamen d folgen, werden expandiert zu allen Unterverzeichnissen von d . Dieser Vorgang findet rekursiv statt, wobei erst alle Verzeichnisse auf einer Ebene bearbeitet werden, dann deren Unterverzeichnisse, usw. Auf den jeweiligen Ebenen ist es nicht beeinflußbar, in welcher Reihenfolge die Unterverzeichnisse bearbeitet werden.

Wenn nach den Schrägstrichen Namen angegeben werden, dann werden nur Unterverzeichnisse mit passenden Namen in die Suche einbezogen. Beispielsweise wird `"/a//b"` in die Pfade `/a/1/b`, `/a/2/b`, `/a/1/1/b` usw. expandiert, aber nicht zu `/a/b/c` oder `/a/1`. (Jeweils vorausgesetzt, daß die Verzeichnisse existieren.)

Mehrere `"/"`-Konstruktionen innerhalb einer Pfadkomponente sind zulässig, allerdings nicht am Pfadanfang.

6.1.7 Zusammenfassung der Sonderzeichen

Folgende Zusammenfassung wiederholt alle Sonderzeichen, die in Kpathsea-Konfigurationsdateien auftreten können:

- : Trennzeichen für Pfadkomponenten; als erstes oder letztes Zeichen im Pfad bewirkt es die Expansion der Voreinstellungen.

; Trennzeichen für Pfadkomponenten für andere Rechnerplattformen als Unix (Verwendung wie :)
 \$ Expansion von Variableninhalten
 ~ Home-Verzeichnis eines Benutzers
 { . . . } Expansion geschweifeter Klammern, beispielsweise wird $a\{1,2\}b$ zu $a1b : a2b$
 // Expansion von Unterverzeichnissen. Tritt niemals am Anfang einer Pfadkomponente auf.
 % Kommentar
 \ Zeilenkonkatenation
 !! Einschränkung der Suche auf *ausschließlich* die Dateinamen-Datenbank. Das Dateisystem wird *nicht* durchsucht!

6.2 Dateinamen-Datenbanken

Kpathsea unternimmt etliche Anstrengungen, um den Zugriff auf Festplatte und CD-ROM zur Suche nach Dateien einzuschränken. Auf T_EX-Systemen mit vielen Unterverzeichnissen kann die Suche in jedem möglichen Verzeichnis nach einer bestimmten Datei eine lange Zeit in Anspruch nehmen, besonders wenn einige Hundert Zeichensatzverzeichnisse durchforstet werden müssen. Um dieses Problem abzumildern, benutzt Kpathsea eine Art Datenbankdatei namens `ls-R`, die die Zuordnung von Dateinamen auf Verzeichnisse enthält. Dadurch muß nicht jedesmal die Festplatte durchsucht werden.

Eine zweite Datenbank in der Datei `aliases` kann eine Zuordnung zwischen den Namen in `ls-R` und weiteren Namen vornehmen und so beispielsweise hilfreich bei der Umsetzung von „8.3“-Dateinamen (wie sie auf der CD-ROM obligatorisch sind) auf die „echten“, aussagekräftigen Dateinamen zur Seite stehen.

6.2.1 Die `ls-R`-Datenbank

Wie schon öfters erwähnt, muß die Datenbank der Dateinamen in der Datei `ls-R` gespeichert sein. Sie sollten eine solche Datenbank für jede T_EX-Hierarchie (normalerweise in `$TEXMF`) Ihres Systems anlegen. Die meisten T_EX-Systeme arbeiten nur mit einer Hierarchie. Kpathsea sucht die Datenbanken `ls-R` über den Pfad `TEXMFDBS`.

Es wird empfohlen, die Pflege der „`ls-R`“-Dateien dem mitgelieferten Skript `mktexlsr` zu überlassen. Dieses Skript wird automatisch von den verschiedenen „`mktex . . .`“-Skripten aufgerufen. Das Skript ruft grob gesagt den Befehl

```
cd /your/teXmf/root && ls -LAR ./ >ls-R
```

auf, falls das `ls`-Kommando Ihres Rechners eine Ausgabe im richtigen Format liefert. (So wie das GNU `ls`.) Wenn Sie ganz sichergehen wollen, daß die Datenbank immer auf dem neuesten Stand ist, sollten Sie sie in regelmäßigen Abständen mit Hilfe eines `crontab`-Eintrags aktualisieren lassen. Dadurch wird nach einer Installation eines Pakets von Hand trotzdem sichergestellt, daß die Datenbank aktuell ist.

Wenn eine Datei nicht über die Datenbank gefunden wird, sucht Kpathsea normalerweise auf der Festplatte weiter. Wenn eine Pfadkomponente mit „!!“ beginnt, wird dagegen niemals die Festplatte durchsucht.

6.2.2 kpsewhich: Dateisuche

Mit dem `kpsewhich`-Programm können Sie unabhängig vom Aufruf irgendeines \TeX -Programms nach Dateien in der \TeX -Hierarchie suchen (als schnellere Alternative zu einem `find`-Befehl). Dies wird von den verschiedenen “`mktex...`”-Skripten bis zum Exzess vorexerziert.

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

Die Optionen werden entweder mit “-” oder mit “--” eingeleitet. Jede eindeutige Abkürzung ist zulässig.

Argumente der Kommandozeile, die keine Optionen darstellen, werden als Dateinamen interpretiert. Für jeden Dateinamen wird der erste passende Pfad gemeldet. Um eine Liste aller passenden Pfade zu erhalten, müssen Sie ein “`find`”-Kommando absetzen.

Im folgenden werden die wichtigen Optionen beschrieben.

`-dpi=num` Stellt die Auflösung für die Suche nach Zeichensätzen auf *num* dpi. Alternativ kann die Option `-D` (kommt von `dvips`) benutzt werden. Voreinstellung ist 600.

`-format=name`

Setzt das Format zur Suche auf *Name*. Per Voreinstellung versucht `kpsewhich` das Format aus dem Dateinamen zu raten. Bei Formaten ohne zugeordneten Suffix wie den zu MetaPost gehörenden Dateien und den Konfigurationsdateien zu `dvips` müssen Sie den entsprechenden Namen eingeben, den Sie in der ersten Spalte von Tabelle 1 finden. Diese Tabelle enthält die momentan bekannten Namen, eine Beschreibung und die zugehörigen Environmentvariablen. ¹

Tabelle 1: Kpathsea Dateitypen

<i>Name</i>	<i>Beschreibung</i>	<i>Environment</i>	<i>Endung</i>
<code>afm</code>	Adobe PostScript Metrikdateien	AFMFONTS	<code>.afm</code>
<code>base</code>	Metafont-Basis	MFBASES, TEXMFINI	<code>.base</code>
<code>bib</code>	BIB \TeX -Literaturdatenbank	BIBINPUTS, TEXBIB	<code>.bib</code>
	Pixelbilder von Zeichensätzen	GLYPHFONTS, TEXFONTS	
<code>bst</code>	BIB \TeX -Stildateien	BSTINPUTS	<code>.bst</code>
<code>cnf</code>	Konfigurationsdateien	TEXMFCNF	<code>.cnf</code>
<code>dvips config</code>	<code>dvips</code> -Konfigurationsdateien, z. B. <code>config.ps</code> und <code>psfonts.map</code>	TEXCONFIG	<code>.map</code>
<code>fmt</code>	\TeX -Format	TEXFORMATS, TEXMFINI	<code>.fmt</code> , <code>.efmt</code> , <code>.efm</code>
<code>gf</code>	METAFONT-Zeichensätze	GFFONTS	<code>.gf</code>
<code>graphic/figure</code>	Encapsulated-PostScript-Dateien	TEXPICTS, TEXINPUTS	<code>.eps</code> , <code>.epsi</code>
<code>ist</code>	makeindex Stil-Dateien	TEXINDEXSTYLE, INDEXSTYLE	<code>.ist</code>
<code>ls-R</code>	Dateinamen-Datenbanken	TEXMFDBS	
<code>map</code>	Zeichensatzzuordnungstabellen	TEXFONTMAPS	<code>.map</code>
<code>mem</code>	MetaPost-Format	MPMEMS, TEXMFINI	<code>.mem</code>
<code>mf</code>	Metafont-Quelldatei	MFINPUTS	<code>.mf</code>
<code>mfpool</code>	Metafont-Bildschirmmeldungen	MFPOOL, TEXMFINI	<code>.pool</code>
<code>mft</code>	MFT-Stildateien	MFTINPUTS	<code>.mft</code>
	verschiedene Zeichensätze	MISCFONTS	
<code>mp</code>	MetaPost-Quelldateien	MPINPUTS	<code>.mp</code>
<code>mppool</code>	MetaPost-Bildschirmmeldungen	MPPPOOL, TEXMFINI	<code>.pool</code>

¹Die normalerweise mit den Environmentvariablen verwendeten Werte und die möglichen Endungen sehen Sie in der Datei `texmf.cnf` (Seite 27).

Kpathsea Dateitypen *Forts.*

<i>Name</i>	<i>Beschreibung</i>	<i>Environment</i>	<i>Endung</i>
MetaPost support	MetaPost-Hilfsdateien für DMP	MPSUPPORT	
ocp	Ω-Prozeß	OCPINPUTS	.ocp
ofm	Ω-Metrik	OFM FONTS, TEX FONTS	.ofm, .tfm
opl	Ω-Property-Liste	OPL FONTS, TEX FONTS	.opl
otp	Ω-Prozeß	OTPINPUTS	.otp
ovf	Ω virtuelle Zeichensätze	OVP FONTS, TEX FONTS	.ovf
ovp	Ω virtuelle Property-Liste	OVP FONTS, TEX FONTS	.ovp
pk	gepackte Zeichensätze	program FONTS wobei (programm z. B. XDVl, usw., PK FONTS, TEXPKS, GLYPH FONTS, TEX FONTS	.pk
PostScript header	PostScript-Macros	TEXPSHEADERS, PSHEADERS	.pro, .enc
tex	TeX-Quelldatei	TEXINPUTS	.tex, .cls, .sty, .clo, .def
TeX system documentation	Dokumentation zum TeX-System	TEXDOCS	
TeX system sources	TeX-Quellen	TEXSOURCES	
texpool	TeX-Bildschirmmeldungen	TEXPOOL, TEXMFINI	.pool
tfm	TeX-Metrikdateien	TFM FONTS, TEX FONTS	.tfm
Troff fonts	Troff-Zeichensätze für DMP	TRFONTS	
truetype fonts	TrueType-Zeichensätze	TTFONTS	.ttf, .ttc
type1 fonts	Type 1 PostScript-Zeichensätze	T1FONTS, T1INPUTS, TEXPSHEADERS, DVIPSHEADERS	.pfa, .pfb
type42 fonts	Type 42 PostScript-Zeichensätze	T42FONTS	
vf	virtuelle Zeichensätze	VFFONTS, TEX FONTS	.vf
web2c files	Web2c-Hilfsdateien	WEB2C	
other text files	sonstige Textdateien	FOOINPUTS	
other binary files	sonstige Binärdateien	FOOINPUTS	

Die beiden letzten Einträge in Tabelle 1 sind Spezialfälle, in denen die Pfade bzw. Umgebungsvariablen nach den zugehörigen Programmen benannt sind. Hier wird der Programmname in Großbuchstaben umgewandelt und anschließend die Zeichenkette INPUTS angehängt.

Die Umgebungsvariablen werden in der Konfigurationsdatei `texmf.cnf` definiert. Nur, wenn Sie explizit eine der Einstellungen ändern wollen, definieren Sie die entsprechenden Variablen über Ihre Shell.

Übrigens können Sie nur eine der beiden Optionen “`-format`” und “`-path`” gleichzeitig angeben.

-mode=string

Setzt für das Suchen von Zeichensätzen den Generierungsmodus. Normalerweise werden alle Zeichensätze gemeldet.

-must-exist

Versucht in jedem Fall, die Dateien zu finden, auch durch Suchen auf der Festplatte. Normalerweise wird nur die `ls-R`-Datenbank konsultiert.

- path=string
Sucht entlang des angegebenen Pfades statt des Standard-Pfades, der auf Grund der Extension gewählt wird. Alle Expansionen sind zulässig. Bei Verwendung der “-path”-Option darf keine “-format”-Option angegeben werden.
- programe=name
Setzt den Programmnamen für die genauere Variablenspezifikation über “.Programname”. Voreinstellung ist “kpsewhich”.
- show-path=name
Zeigt den Suchpfad für die angegebene Extension. Diese kann entweder als Name (“.pk”, “.vf”, etc.) oder als Zahl (wie bei der “-format”-Option) spezifiziert werden.
- debug=num
Legt die Stufe für die Fehlersuche fest.

6.2.3 Beispiele

Wir schauen uns nun die Funktionsweise von Kpathsea anhand einiger Beispiele an.

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

Wir suchen unter den \TeX -Quelldateien nach der Datei `article.cls`. Nachdem der suffix “.cls” eindeutig ist, müssen wir den Typ „tex“ nicht angeben. Die “TEXMF”-Hierarchie enthält die Datei im Unterverzeichnis `tex/latex/base`. Ähnlich bereiten die folgenden Beispiele auf Grund eindeutiger Suffixe keine Probleme.

```
>> 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
```

Beim letzten Beispiel handelt es sich um eine $\text{BIB}\TeX$ -Literaturdatenbank für *TUGBoat*-Artikel.

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

Zeichensatzdateien mit Endung `.pk` werden von Anzeige- oder Druckaufbereitungsprogrammen wie `dvips` und `xdvi` verwendet. Nachdem wir keine “.pk”-Dateien verwenden, sondern die Type 1-Zeichensätze, die auf der CD-ROM enthalten sind, wird auch keine “.pk”-Datei angezeigt.

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

Für die neuen EC-Zeichensätze liegen noch keine Type1-Umsetzungen vor. Da unser voreingestellter METAFONT-Modus `ljfour` mit einer Auflösung von 600 dpi ist, finden wir (nachdem er schon einmal gebraucht und automatisch erzeugt wurde) eine entsprechende Instanz dieses Zeichensatzes.

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

In diesem Fall interessieren wir uns durch Angabe von `--dpi=300` nur für entsprechende Zeichensätze in der Auflösung 300 dpi, wobei keiner gefunden wird. Ein Programm wie `dvips` oder `xdvi` würde einen solchen fehlenden Zeichensatz durch Aufruf des Skripts `mktexpk` mit entsprechenden Parametern automatisch erzeugen.

Als nächstes wenden wir uns den Header- und Konfigurations-Dateien von `dvips` zu. Zunächst suchen wir nach der Konfiguration für $\text{T}_{\text{E}}\text{X}$ -Unterstützung, dem Prolog `tex.pro`. Danach suchen wir die allgemeine Konfigurationsdatei (`config.ps`) und schließlich die PostScript-Zeichensatzzuordnungsdatei `psfonts.map`. Da der Suffix `“.ps”` nicht eindeutig ist, müssen wir den gewünschten Typ (`„dvips config“`) für die Datei `config.ps` spezifizieren.

```
>> 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
```

Jetzt suchen wir nach den Dateien für den PostScript-Zeichensatz URW Times. Nach dem Namensschema von Karl Berry beginnen die Namen mit `„utm“`. Zunächst suchen wir die Konfigurationsdatei, die den Namen der Zeichensatzzuordnungsdatei enthält.

```
>> kpsewhich --format="dvips config" config.utm  
/usr/local/texmf/dvips/psnfss/config.utm
```

Diese Datei enthält folgende Anweisung:

```
p +utm.map
```

Die angegebene Datei `utm.map` wollen wir als nächstes suchen:

```
>> kpsewhich --format="dvips config" utm.map  
/usr/local/texmf/dvips/psnfss/utm.map
```

Diese Zuordnungsdatei wird im Unterverzeichnis `urw` bei den Hilfsdateien für `dvips` gefunden. Sie enthält die Dateinamen der Type1-PostScript-Zeichensätze, die für URW Times benutzt werden. Ein kleiner Auszug aus dieser Datei:

```
utmb8r NimbusRomNo9L-Medi ... <utmb8a.pfb  
utmbi8r NimbusRomNo9L-MediItal... <utmbi8a.pfb  
utmr8r NimbusRomNo9L-Regu ... <utmr8a.pfb  
utmri8r NimbusRomNo9L-ReguItal... <utmri8a.pfb  
utmb08r NimbusRomNo9L-Medi ... <utmb8a.pfb  
utmro8r NimbusRomNo9L-Regu ... <utmr8a.pfb
```

Wenn wir jetzt beispielsweise nach dem Zeichensatz Times Regular (`utmr8a.pfb`) suchen, finden wir ihn im `texmf`-Verzeichnis unter den Type1-Zeichensätzen:

```
>> kpsewhich utmr8a.pfb  
/usr/local/texmf/fonts/type1/urw/utm/utmr8a.pfb
```

Diese Beispiele sollten deutlich gemacht haben, wie leicht bestimmte Dateien im $\text{T}_{\text{E}}\text{X}$ -Dateibaum gefunden werden können. Dies ist sehr wichtig, wenn Sie den Verdacht haben, daß eine falsche Version einer Datei verwendet wird. Sie lassen sich einfach die verwendete Datei von `kpsewhich` anzeigen.

6.2.4 Fehlersuche

Manchmal ist wichtig, bis ins Detail nachzuvollziehen, wie ein Programm eine bestimmte Datei findet. Zu diesem Zweck bietet die Kpathsea-Bibliothek verschiedene Stufen für Ausgaben zur Fehlersuche an.

- 1 stat-Aufrufe (Überprüfung, ob Datei existiert); mit einer aktuellen `ls-R`-Datenbank sollten Sie fast keine Ausgaben erhalten.
- 2 Zugriffe auf Suchlisten (wie `ls-R`-Datenbank, Zuordnungsdateien (`.map`), Konfigurationsdateien)
- 4 Öffnen und Schließen von Dateien
- 8 Ausgabe der voreingestellten Pfade für Extensionen
- 16 Verzeichnisliste für jede Pfadkomponente (nur bei Festplattenzugriff)
- 32 Suchaktionen nach Dateien

Durch Angabe von `-1` setzen Sie alle Stufen gleichzeitig. Für eine effiziente Fehlersuche sollten Sie sich auf die wichtigsten Ausgaben beschränken.

Für `dvips` gibt es einen ähnlichen Mechanismus zur Erzeugung von Analysemeldungen um herauszufinden, warum bestimmte Dateien geöffnet wurden bzw. wo vielleicht das Problem liegt, wenn Dateien nicht gefunden werden.

Da fast alle Programme die Kpathsea-Bibliothek benutzen, können Sie die gewünschte Stufe auch über die Umgebungsvariable `KPATHSEA_DEBUG` einstellen, indem Sie einen der Werte oder eine additive Kombination spezifizieren.

Wir betrachten als Beispiel eine kleine \LaTeX -Quelldatei namens `hello-world.tex` mit folgendem Inhalt:

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

Diese Datei verwendet nur einen Zeichensatz, `cmr10`. Wir sehen uns jetzt einmal genau an, wie `dvips` die PostScript-Datei erzeugt. (Da wir die `Type1`-Variante der Computer Modern Roman-Zeichensätze verwenden wollen, haben wir die Option `-Pcms` verwendet).

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

Hier haben wir als Stufe zur Fehlersuche eine Kombination der Stufe 4 von `dvips` (siehe `dvips-Handbuch`, texmf/doc/html/dvips/dvips_toc.html). Die Ausgabe sieht in etwa wie [Abbildung 4](#) aus (die Ausgabe ist für besseren Überblick etwas umgestaltet). Zunächst sucht `dvips` (bzw. `Kpathsea`) seine Konfigurationsdateien, nämlich `texmf.cnf` (das die Pfade der anderen Dateien enthält), dann die Dateinamen-Datenbank `ls-R` (zur Optimierung der Suche) und die Datei `aliases`, mit deren Hilfe für eine Datei mehrere Namen vereinbart werden können, z. B. um die kurzen "8.3" DOS-Namen mit aussagefähigen, langen Namen zu assoziieren. Danach wird die allgemeine `dvips`-Konfigurationsdatei `config.ps`, anschließend die benutzerspezifische Konfigurationsdatei `.dvipsrc` (wird hier *nicht* gefunden) gesucht. Als letztes sucht `dvips` die Zuordnungsdatei für Computer Modern PostScript-Zeichensätze `config.cms` (bedingt durch die Option `-Pcms` beim Aufruf von `dvips`). Diese Datei enthält die Dateinamen der Listen, die die die Zuordnung zwischen Dateinamen und Zeichensatznamen herstellen.

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

```

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

```

Abbildung 4: 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

```

Abbildung 5: 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]

```

Abbildung 6: Finding the font file

```

p +cms.map
p +cmbkm.map
p +amsbkm.map

```

dvips versucht, all diese Dateien und zusätzlich die allgemeine Zeichensatzzuordnungstabelle `ps-fonts.map` zu laden, die immer konsultiert wird; der letzte Teil von Abschnitt 6.2.3 erklärt diese Tabellen genauer.

Jetzt erfolgt die normale Startmeldung von dvips:

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

Danach wird nach `texc.pro` gesucht:

```
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
```

Danach gibt `dvips` Datum und Uhrzeit aus und meldet den Dateinamen der generierten PostScript-Datei `hello-world.ps`. Jetzt wird die Zeichensatzdatei `cmr10` benötigt, die `dvips` als „resident“ meldet.

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

Es geht weiter mit `cmr10.tfm` und einigen weiteren Prologdateien, deren Ausgaben wir hier weglassen. Letztlich wird die Type1-Zeichensatzdatei `cmr10.pfb` gesucht (und gefunden) und in die Ausgabedatei integriert (siehe letzte Zeile).

```
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]
```

6.3 Einstellungen zur Laufzeit

Zu den willkommenen Erweiterungen von `Web2c 7.3` zählt die Möglichkeit, zur Laufzeit einige Speichergrößen über die Datei `texmf.cnf` anpassen zu können (insbesondere die Größe einiger Stacks). Eine ausführliche Liste der veränderbaren Parameter finden Sie in der Datei, die ab Seite 27 in dieser Anleitung abgedruckt ist. Die interessanten Parameter beginnen in Teil 3. Die wichtigsten Werte sind:

`main_memory` Arbeitsspeicher für `TEX`, `METAFONT` und `MetaPost` in Worten. Für jede Einstellung muß eine eigene Format-Datei erstellt werden. Allerdings können Sie mehrere Versionen von `TEX` unter verschiedenen Namen erzeugen und in der Konfigurationsdatei jeweils eigene Einträge vorsehen. Hier gibt es ein Monster-`TEX` namens „`hugetex`“. Das normale `TEX` erhält den Wert in Zeile 357, für `hugetex` gilt Zeile 432.

`extra_mem_bot` Extraspeicher für „große“ `TEX`-Datenstrukturen wie Boxen, Glue, Breakpoints, usw. Besonders bei Anwendung von `PTCTEX` sollte dieser Wert erhöht werden.

`font_mem_size` Anzahl Worte für Speicherung von Zeichensatzinformationen. Entspricht ungefähr dem Speicherbedarf der gelesenen TFM-Dateien.

`hash_extra` Zusätzlicher Platz für Suchlisten. In der Hauptliste können ca. 10000 Einträge verwaltet werden. Bei einem Buch mit vielen Querverweisen reicht dieser Platz unter Umständen nicht aus. In der Datei `texmf.cnf` (ab Seite 27) sehen Sie für die Programme `hugetex` und `pdflatex`, daß jeweils 15000 Worte zusätzlich angefordert werden. (Die Voreinstellung ist 0, wie Sie in Zeile 369 sehen.)

Natürlich sind diese Parameter kein Ersatz für eine wirklich dynamische Speicherverwaltung. Mit der gegenwärtigen Version von \TeX ist dieses Konzept aber nur extrem schwer zu implementieren, darum stellt dieses Verfahren eine praktikable Lösung dar.

7 Andere \TeX -Systeme auf der CD-ROM

7.1 emTeX/TDS

Die DOS und OS/2 \TeX -Implementierung emTeX von Eberhard Mattes (mattes@azu.informatik.uni-stuttgart.de) enthält das Textsatzprogramm \TeX , den Zeichensatzgenerator METAFONT, Drucker- und Bildschirmtreiber und alle Hilfsprogramme wie BibTeX und MakeIndex. Ebenfalls enthalten sind die Makropakete \LaTeX 2.09 und \LaTeX 2 ϵ . Die enthaltenen Zeichensätze liegen als METAFONT-Quelldateien und Zeichensatzbibliotheken (mit vollautomatischer Verwaltung und Generierung zur Laufzeit) vor.

Die Installationsanleitung finden Sie auf der CD-ROM in der Datei [systems/msdos/emtex/README.ENG](#).

emTeX/TDS ist ein auf emTeX basierendes, voll TDS-konformes \TeX -System für OS/2 Warp. Es werden viele zusätzliche Pakete und Zeichensätze geliefert, vollständige PostScript-Unterstützung ist gewährleistet, ebenso wie Unterstützung vieler Sprachen. emTeX/TDS wurde von Walter Schmidt (wschmi@ibm.net) zusammengestellt.

Die Installationsanleitung finden Sie auf der CD-ROM in der Datei [systems/os2/emtexTDS/install.eng](#).

8 Geschichte und Danksagungen

Diese CD-ROM ist in Zusammenarbeit der \TeX Users Group, der UK \TeX Users Group, der französischen \TeX -Vereinigung GUTenberg und der deutschen \TeX -Anwendervereinigung (DANTE e. V.) unter Mithilfe der niederländischen, tschechischen/slowakischen, indischen und polnischen \TeX -Benutzergruppen entstanden. Die niederländische \TeX -Benutzergruppe hatte Ende 1993 mit der Produktion der 4AllTeX-CD-ROM für MS-DOS die Diskussion angeregt, eine einzige CD-ROM für alle Rechnersysteme zu entwickeln. Zum damaligen Zeitpunkt war dieses Ziel zu hoch gesteckt, doch immerhin entstand aus dieser Diskussion nicht nur die sehr erfolgreiche 4AllTeX-CD-ROM, sondern auch die TUG-Arbeitsgruppe zur Definition der \TeX Directory Structure TDS, die die zur Arbeit mit \TeX notwendigen und hilfreichen Dateien in eine konsistente und handhabbare Verzeichnisstruktur einbettet. Das „Final Draft“-Dokument, das diese Verzeichnisstruktur festlegt, wurde in der Dezemberausgabe 1995 der TUGBoat veröffentlicht. Schon frühzeitig war den Beteiligten klar, daß eine CD-ROM auf der Basis der TDS sehr zu begrüßen wäre. Die **TeX Live**-CD-ROM ist das direkte Resultat der Beratungen der TDS-Arbeitsgruppe. Außerdem hat der Erfolg der 4AllTeX-CD-ROM klargemacht, daß ein ähnliches System auch für Unix-Benutzer eine Erleichterung darstellen würde. Dies ist der zweite Beweggrund für die **TeX Live**-CD-ROM.

Im Herbst 1995 wurde das Projekt, eine TDS-basierte Unix-CD-ROM zu entwickeln, in Angriff genommen. Sehr schnell stießen die Verantwortlichen auf das teTeX-System von Thomas Esser als idealen Ausgangspunkt für diese Arbeit, weil es verschiedene Rechnerplattformen unterstützte und für die Arbeit mit verschiedenen Dateisystemen vorgesehen war. Anfang 1996 wurde in Zusammenarbeit mit Thomas Esser ernsthaft mit der Arbeit begonnen und im Mai 1996 die erste Ausgabe der CD-ROM veröffentlicht. Anfang 1997 stellte Karl Berry eine neue Version seines Web2c- \TeX -Systems vor, das schon nahezu alle Ausstattungsmerkmale aufwies, die Thomas Esser mit teTeX verwirklicht hatte. Die TUG entschied sich daraufhin, die zweite Version der CD-ROM auf der Basis von Web2c unter Verwendung des Installations-Skriptes texconfig aus dem teTeX-Paket zu entwickeln. Die vorliegende dritte Ausgabe basiert auf der inzwischen von Olaf Weber gepflegten/weiterentwickelten Web2c Version

7.2; **T_EX Live** unterstützt fast alle Eigenschaften der zur selben Zeit entstandenen neuen Version von te_EX. Die vierte Ausgabe folgt demselben Schema, indem ihre neuen Versionen von te_EX und Web2c (7.3) zugrunde liegen. Fast die gesamte CD-ROM wurde einer kritischen Überprüfung unterzogen, wobei besonders darauf geachtet wurde, daß doppelte Dateien entfernt werden und die Einordnung der Pakete konsistent erfolgt.

Unser besonderer Dank gilt:

- Karl Berry, von dem die erste Web2c-Version stammt, für kontinuierlichen Rat, Ermutigung und Hilfe;
- Mimi Burbank, die durch Zurverfügungstellung einer Vielzahl verschiedener Rechnersysteme am Florida State University Supercomputer Research Institute die Erstellung der verschiedenen T_EX-Binärprogramme ermöglichte und jederzeit Systeme testete, wann immer man sie fragte;
- Kaja Christiansen, die wichtige Rückmeldungen brachte und half, die Doku vorzubereiten und zu verbessern;
- Thomas Esser, der mit dem exzellenten te_EX die Basis für diese CD schuf und dessen ständige Hilfe das Resultat deutlich verbessert hat;
- Eitan Gurari, mit dessen TeX4ht die HTML-Version dieser Anleitung erstellt wurde und der unermüdlich daran gearbeitet hat, es auf Zuruf zu verbessern;
- Art Ogawa und Pat Monohon, die diese Ausgabe für die TUG koordinierten;
- Petr Olsak, der das tschechische und slowakische Material sehr sorgfältig er- und überarbeitet hat;
- Fabrice Popineau, der unermüdlich an der 32-Bit-Windows-Umsetzung (besonders dem Installationsprogramm) gearbeitet hat und auf vielfältige Weise Ideen, Ratschläge und Programmcode eingebracht hat;
- Staszek Wawrykiewicz, der ein geduldiger Tester war und die polnischsprachigen Beiträge koordiniert hat;
- Olaf Weber für die Geduld beim Zusammenpuzzeln und Pflegen von Web2c 7.3;
- Graham Williams, dessen Arbeit das Makro- und Paketverzeichnis möglich gemacht hat.

Gerhard Wilhelms, Fabrice Popineau, Janka Chlebíková, Staszek Wawrykiewicz, Erik Frambach und Ulrik Vieth haben freundlicherweise diese Anleitung in ihre jeweiligen Landessprachen übersetzt, Teile der Dokumentation geprüft und wertvolle Rückmeldungen gegeben.

9 Zukünftige Versionen

Diese CD-ROM ist nicht perfekt! Wir planen im Abstand von jeweils einem Jahr Neuauflagen zu produzieren, die umfangreichere Hilfstexte, mehr Hilfsprogramme, mehr Installationshilfen und (natürlich) den ständig verbesserten und überprüften Satz Makros und Zeichensätze enthalten. Diese Arbeit wird von Freiwilligen in ihrer knappen Freizeit geleistet. Viel Arbeit steht noch an. Wenn Sie uns helfen können, zögern Sie keine Sekunde, sich bei uns zu melden.

Verbesserungen, Anregungen und Erweiterungen für künftige Ausgaben der **T_EX Live**-CD-ROM senden Sie bitte an:

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

Neue Versionen, Anmerkungen und zusätzliches Material werden über die CTAN-Server im Verzeichnis `info/texlive` zur Verfügung gestellt. Im WWW können Sie sich über die Adresse <http://www.tug.org/tex-live.html> informieren und die Bestellformalitäten einsehen.

10 Die Datei `texmf.cnf`

```
1 % TeX Live texmf.cnf
2 % What follows is a super-summary of what this .cnf file can
3 % contain. Please read the Kpathsea manual for more information.
4 %
5 % texmf.cnf is generated from texmf.in, by replacing @var@ with the
6 % value of the Make variable 'var', via a sed file texmf.sed, generated
7 % (once) by kpathsea/Makefile (itself generated from kpathsea/Makefile.in
8 % by configure).
9 %
10 % Any identifier (sticking to A-Za-z_ for names is safest) can be assigned.
11 % The '=' (and surrounding spaces) is optional.
12 % No % or @ in texmf.in, for the sake of autogeneration.
13 % (However, %'s and @'s can be edited into texmf.cnf or put in envvar values.)
14 % $foo (or ${foo}) in a value expands to the envvar or cnf value of foo.
15 %
16 % Earlier entries (in the same or another file) override later ones, and
17 % an environment variable foo overrides any texmf.cnf definition of foo.
18 %
19 % All definitions are read before anything is expanded, so you can use
20 % variables before they are defined.
21 %
22 % If a variable assignment is qualified with '.PROGRAM', it is ignored
23 % unless the current executable (last filename component of argv[0]) is
24 % named PROGRAM. This foo.PROGRAM construct is not recognized on the
25 % right-hand side. For environment variables, use FOO_PROGRAM.
26 %
27 % Which file formats use which paths for searches is described in the
28 % various programs' and the kpathsea documentation.
29 %
30 % // means to search subdirectories (recursively).
31 % A leading !! means to look only in the ls-R db, never on the disk.
32 % A leading/trailing/doubled ; in the paths will be expanded into the
33 % compile-time default. Probably not what you want.
34 %
35 % You can use brace notation, for example: /usr/local/{mytex:othertex}
36 % expands to /usr/local/mytex:/usr/local/othertex. Instead of the path
37 % separator you can use a comma: /usr/local/{mytex,othertex} also expands
38 % to /usr/local/mytex:/usr/local/othertex. However, the use of the comma
39 % instead of the path separator is deprecated.
40 %
41 % The text above assumes that path separator is a colon (:). Non-UNIX
42 % systems use different path separators, like the semicolon (;).
43 %
44 % Part 1: Search paths and directories.
45 %
46 % You can set an environment variable to override TEXMF if you're testing
47 % a new TeX tree, without changing anything else.
48 %
49 % You may wish to use one of the $SELFAUTO... variables here so TeX will
50 % find where to look dynamically. See the manual and the definition
51 % below of TEXMFCNF.
52 %
53 % The main tree, which must be mentioned in $TEXMF, below:
54 TEXMFMAIN = $SELFAUTOPARENT/texmf
55 % A place for local additions to a "standard" texmf tree.
56 TEXMFLocal = $SELFAUTOPARENT/texmf-local
57 %
58 % User texmf trees can be catered for like this...
```

```

59 HOMETEXMF=$HOME/texmf
60
61 % A place where texconfig stores modifications (instead of the TEXMFMAIN
62 % tree). texconfig relies on the name, so don't change it.
63 VARTEXMF = $SELFAUTOPARENT/texmf-var
64
65 % Now, list all the texmf trees. If you have multiple trees,
66 % use shell brace notation, like this:
67 %   TEXMF = {$HOMETEXMF,!!$VARTEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN}
68 % The braces are necessary.
69 %
70 % A place where to store other TeX support files. It can be a remote
71 % texmf tree, or a tree to store non-free stuff, or ...
72 %   TEXMFEXTRA=$SELFAUTOPARENT/texmf-extra
73 % If you set this, add $TEXMFEXTRA in the list below
74 %
75 TEXMF = {$HOMETEXMF,!!$VARTEXMF,$TEXMFLOCAL,!!$TEXMFMAIN}
76
77 % The system trees. These are the trees that are shared by all the users.
78 SYSTEXMF = $TEXMF
79
80 % The temporary area
81 TEMP = /var/tmp
82
83 % Where generated fonts may be written. This tree is used when the sources
84 % were found in a system tree and either that tree wasn't writable, or the
85 % varfonts feature was enabled in MT_FEATURES in mktex.cnf.
86 VARTEFONTS = $VARTEXMF/fonts
87
88 % Where to look for ls-R files. There need not be an ls-R in the
89 % directories in this path, but if there is one, Kpathsea will use it.
90 TEXMFDDBS = $TEXMF
91
92 % It may be convenient to define TEXMF like this:
93 %   TEXMF = {$HOMETEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN,$HOME}
94 % which allows users to set up entire texmf trees, and tells TeX to
95 % look in places like ~/tex and ~/bibtex. If you do this, define TEXMFDDBS
96 % like this:
97 %   TEXMFDDBS = $HOMETEXMF;$TEXMFLOCAL;$TEXMFMAIN;$VARTEFONTS
98 % or mktexlsr will generate an ls-R file for $HOME when called, which is
99 % rarely desirable. If you do this you'll want to define SYSTEXMF like
100 % this:
101 %   SYSTEXMF = $TEXMFLOCAL;$TEXMFMAIN
102 % so that fonts from a user's tree won't escape into the global trees.
103 %
104 % On some systems, there will be a system tree which contains all the font
105 % files that may be created as well as the formats. For example
106 %   VARTEXMF = /var/lib/texmf
107 % is used on many Linux systems. In this case, set VARTEFONTS like this
108 %   VARTEFONTS = $VARTEXMF/fonts
109 % and do not mention it in TEXMFDDBS (but _do_ mention VARTEXMF).
110
111
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 % Usually you will not need to edit any of the other variables in part 1. %
114 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
115
116 % WEB2C is for Web2C specific files. The current directory may not be
117 % a good place to look for them.
118 WEB2C = $TEXMF/web2c
119
120 % TEXINPUTS is for TeX input files -- i.e., anything to be found by \input
121 % or \openin, including .sty, .eps, etc.
122
123 % LaTeX-specific macros are stored in latex.
124 TEXINPUTS.latex = .;$TEXMF/tex/{latex,generic,}//
125 TEXINPUTS.hugelatex = .;$TEXMF/tex/{latex,generic,}//
126
127 % Fontinst needs to read afm files.
128 TEXINPUTS.fontinst = .;$TEXMF/{tex/{fontinst,},fonts/afm}//
129
130 % Plain TeX. Have the command tex check all directories as a last
131 % resort, we may have plain-compatible stuff anywhere.
132 TEXINPUTS.tex = .;$TEXMF/tex/{plain,generic,}//

```

```

133 % other plain-based formats
134 TEXINPUTS.amstex = .;$TEXMF/tex/{amstex,plain,generic,}//
135 TEXINPUTS.ftex = .;$TEXMF/tex/{formate,plain,generic,}//
136 TEXINPUTS.texinfo = .;$TEXMF/tex/{texinfo,plain,generic,}//
137 TEXINPUTS.eplain = .;$TEXMF/tex/{eplain,plain,generic,}//
138
139 % e-TeX.
140 TEXINPUTS.elatex = .;$TEXMF/{etex,tex}/{latex,generic,}//
141 TEXINPUTS.etex = .;$TEXMF/{etex,tex}/{plain,generic,}//
142
143 % PDFTeX. This form of the input paths is borrowed from TeTeX. A certain
144 % variant of TDS is assumed here, unaffected by the build variables.
145 TEXINPUTS.pdfetexinfo = .;$TEXMF/{pdfetex,tex}/{texinfo,plain,generic,}//
146 TEXINPUTS.pdflatex = .;$TEXMF/{pdfetex,tex}/{latex,generic,}//
147 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,tex}/{plain,generic,}//
148 TEXINPUTS.pdfelatex = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{latex,generic,}//
149 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{plain,generic,}//
150
151 % Omega.
152 TEXINPUTS.lambada = .;$TEXMF/{omega,tex}/{lambda,latex,generic,}//
153 TEXINPUTS.omega = .;$TEXMF/{omega,tex}/{plain,generic,}//
154
155 % Context macros by Hans Hagen:
156 TEXINPUTS.context = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{context,plain,generic,}//
157
158 % cstex, from Petr Olsak
159 TEXINPUTS.cslatex = .;$TEXMF/tex/{cslatex,csplain,latex,generic,}//
160 TEXINPUTS.csplain = .;$TEXMF/tex/{csplain,plain,generic,}//
161 TEXINPUTS.pdfcselatex = .;$TEXMF/{pdfetex,tex}/{cslatex,csplain,latex,generic,}//
162 TEXINPUTS.pdfcsplain = .;$TEXMF/{pdfetex,tex}/{csplain,plain,generic,}//
163
164 % Polish
165 TEXINPUTS.platex = .;$TEXMF/tex/{platex,latex,generic,}//
166 TEXINPUTS.pdfmex = .;$TEXMF/{pdfetex,tex}/{mex,plain,generic,}//
167 TEXINPUTS.mex = .;$TEXMF/tex/{mex,plain,generic,}//
168 TEXINPUTS.pdfplatex = .;$TEXMF/{pdfetex,tex}/{platex,latex,generic,}//
169
170 % french
171 TEXINPUTS.frtex = .;$TEXMF/{mltex,tex}/{plain,generic,}//
172 TEXINPUTS.frlatex = .;$TEXMF/{mltex,tex}/{frlatex,latex,generic,}//
173
174 % MLTeX
175 TEXINPUTS.mltex = .;$TEXMF/{mltex,tex}/{plain,generic,}//
176 TEXINPUTS.mllatex = .;$TEXMF/{mltex,tex}/{latex,generic,}//
177
178 % odd formats needing their own paths
179 TEXINPUTS.lollipop = .;$TEXMF/tex/{lollipop,generic,plain,}//
180 TEXINPUTS.lamstex = .;$TEXMF/tex/{lamstex,generic,plain,}//
181
182 % David Carlisle's xmltex
183 TEXINPUTS.xmltex = .;$TEXMF/tex/{xmltex,latex,generic,}//
184 TEXINPUTS.pdfxmltex = .;$TEXMF/{pdfetex,tex}/{xmltex,latex,generic,}//
185
186 % Sebastian Rahtz' jadetex for DSSSL
187 TEXINPUTS.pdfjadetex = .;$TEXMF/{pdfetex,tex}/{jadetex,generic,plain,}//
188 TEXINPUTS.jadetex = .;$TEXMF/tex/{jadetex,generic,plain,}//
189
190 % Earlier entries override later ones, so put this last.
191 TEXINPUTS = .;$TEXMF/tex/{generic,}//
192
193 % Metafont, MetaPost inputs.
194 MFINPUTS = .;$TEXMF/metafont//;$TEXMF/fonts,$VARTEXFONTS/source//
195 MPINPUTS = .;$TEXMF/metapost//
196
197 % Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see web2c/INSTALL),
198 % and string pools (.pool) for ini{tex,mf,mp}. It is silly that we have six
199 % paths and directories here (they all resolve to a single place by default),
200 % but historically ...
201 TEXFORMATS = .;$TEXMF/web2c
202 MFBASES = .;$TEXMF/web2c
203 MPMEMS = .;$TEXMF/web2c
204 TEXPOOL = .;$TEXMF/web2c
205 MFPOOL = .;$TEXMF/web2c
206 MPPPOOL = .;$TEXMF/web2c

```

```

207 % Device-independent font metric files.
208 VFFONTS = .;$TEXMF/fonts/vf//
209 TFMFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/tfm//
210
211
212 % The $MAKETEX_MODE below means the drivers will not use a cx font when
213 % the mode is ricoh. If no mode is explicitly specified, kpse_prog_init
214 % sets MAKETEX_MODE to /, so all subdirectories are searched. See the manual.
215 % The modeless part guarantees that bitmaps for PostScript fonts are found.
216 PKFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modeless}//
217
218 % Similarly for the GF format, which only remains in existence because
219 % Metafont outputs it (and MF isn't going to change).
220 GFFONTS = .;$TEXMF/fonts/gf/$MAKETEX_MODE//
221
222 % A backup for PKFONTS and GFFONTS. Not used for anything.
223 GLYPHFONTS = .;$TEXMF/fonts
224
225 % For texfonts.map and included map files used by mktexpk.
226 % See ftp://ftp.tug.org/tex/fontname.tar.gz.
227 TEXTONTMAPS = .;$TEXMF/fontname
228
229 % BibTeX bibliographies and style files.
230 BIBINPUTS = .;$TEXMF/bibtex/{bib,}//
231 BSTINPUTS = .;$TEXMF/bibtex/{bst,}//
232
233 % PostScript headers, prologues (.pro), encodings (.enc) and fonts;
234 % this is also where pdftex finds included figures files!
235
236 TEXPSHEADERS.pdflatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
237 TEXPSHEADERS.pdfelatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
238 TEXPSHEADERS.pdfetexinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
239 TEXPSHEADERS.pdfcselatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
240 TEXPSHEADERS.pdfcsplain = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
241 TEXPSHEADERS.pdfetex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
242 TEXPSHEADERS.pdfjadetex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
243 TEXPSHEADERS.pdfplatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
244 TEXPSHEADERS.pdfxmltex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
245 TEXPSHEADERS.pdfmex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
246 TEXPSHEADERS.pdfptex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
247 TEXPSHEADERS.pdfetexinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
248 TEXPSHEADERS.cont-de = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
249 TEXPSHEADERS.cont-en = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
250 TEXPSHEADERS.cont-nl = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
251 TEXPSHEADERS.context = .;$TEXMF/{etex,tex,pdftex,dvips,fonts/type1}//
252 TEXPSHEADERS = .;$TEXMF/{dvips,fonts/type1,pdftex}//
253
254 % PostScript Type 1 outline fonts.
255 T1FONTS = .;$TEXMF/fonts/type1//;$TEXMF/fonts/misc/hbf//
256
257 % PostScript AFM metric files.
258 AFMFONTS = .;$TEXMF/fonts/afm//
259
260 % TrueType outline fonts.
261 TTFONTS = .;$TEXMF/fonts/truetype//
262 TTF2TFMINPUTS = .;$TEXMF/ttf2pk//
263
264 % Type 42 outline fonts.
265 T42FONTS = .;$TEXMF/fonts/type42//
266
267 % A place to puth everything that doesn't fit the other font categories.
268 MISCFONTS = .;$TEXMF/fonts/misc//
269
270 % Dvips' config.* files (this name should not start with 'TEX!').
271 TEXCONFIG = .;$TEXMF/dvips//
272
273 % Makeindex style (.ist) files.
274 INDEXSTYLE = .;$TEXMF/makeindex//
275
276 % Used by DMP (ditroff-to-mpx), called by makempx -troff.
277 TRFONTS = /usr/lib/font/devpost
278 MPSUPPORT = .;$TEXMF/metapost/support
279
280 % For xdvi to find mime.types and .mailcap, if they do not exist in

```

```

281 % $HOME. These are single directories, not paths.
282 % (But the default mime.types, at least, may well suffice.)
283 MIMELIBDIR = $SELFAUTOPARENT/etc
284 MAILCAPLIBDIR = $SELFAUTOPARENT/etc
285
286 % TeX documentation and source files, for use with kpsewhich.
287 TEXDOCS = .;$TEXMF/doc//
288 TEXSOURCES = .;$TEXMF/source//
289
290 % also for compressed files, and various extensions
291 TEXDOCSSUFFIX = .dvi:.ps:.html:.txt
292 TEXDOCSCOMPRESS = .gz:.bz2:.zip:.Z
293 TEXDOCEXT = {$TEXDOCSSUFFIX}{$TEXDOCSCOMPRESS}
294
295 % Omega-related fonts and other files. The odd construction for OFMFONTS
296 % makes it behave in the face of a definition of TFMFONTS. Unfortunately
297 % no default substitution would take place for TFMFONTS, so an explicit
298 % path is retained.
299 OFMFONTS = .;$TEXMF/fonts,$VARTEXFONTS/{ofm,tfm};//;$TFMFONTS
300 OPLFONTS = .;$TEXMF/fonts,$VARTEXFONTS}/opl//
301 OVFFONTS = .;$TEXMF/fonts,$VARTEXFONTS}/ovf//
302 OVPFONTS = .;$TEXMF/fonts,$VARTEXFONTS}/ovp//
303 OTPINPUTS = .;$TEXMF/omega/otp//
304 OCPINPUTS = .;$TEXMF/omega/ocp//
305
306 %% t4ht utility, sharing files with TeX4ht
307 TEX4HTFONTSET=alias,iso8859
308 TEX4HTINPUTS = .;$TEXMF/tex4ht/base//;$TEXMF/tex4ht/ht-fonts/{$TEX4HTFONTSET}//
309 T4HTINPUTS= .;$TEXMF/tex4ht/base//
310 %% The mktex* scripts rely on KPSE_DOT. Do not set it in the environment.
311 KPSE_DOT = .
312
313 % This definition isn't used from this .cnf file itself (that would be
314 % paradoxical), but the compile-time default in paths.h is built from it.
315 % The SELFAUTO* variables are set automatically from the location of
316 % argv[0], in kpse_set_programe.
317 %
318 % About the /. construction:
319 % 1) if the variable is undefined, we'd otherwise have an empty path
320 % element in the compile-time path. This is not meaningful.
321 % 2) if we used /$VARIABLE, we'd end up with // if VARIABLE is defined,
322 % which would search the entire world.
323 %
324 % The TETEXDIR stuff isn't likely to be relevant unless you're using teTeX,
325 % but it doesn't hurt.
326 %
327 TEXMFCNF = .;{$SELFAUTOLOC,$SELFAUTODIR,$SELFAUTOPARENT}\
328 {,/{share,}/texmf{.local,}/web2c};c:/TeX/texmf/web2c
329
330
331 % Suggestions for editor settings under Windows. Uncomment your
332 % preferred option. The corresponding MFEDIT can also be set for use with
333 % Metafont.
334 %
335 % Winedt:
336 % TEXEDIT=C:\WinEdt\WinEdt.exe "[Open('%s');SelLine(%d,7)]
337 % Textpad:
338 % TEXEDIT = c:\Progra~1\TextPad\System\Ddeopn32 TextPad %s(%d)
339 % UltraEdit (newer Win32 versions):
340 % TEXEDIT = uedit32 %s/%d/1
341 % WinTeXShell32:
342 % TEXEDIT = texshell.exe /l=%d %s
343 % vi, vim, gvim. here we show Windows gvim.exe:
344 % TEXEDIT = gvim.exe %s +%d
345 % PFE:
346 % TEXEDIT=pfe32/g%d %s
347 % MED:
348 % TEXEDIT=med.exe "%s" %d
349 % TSE:
350 % TEXEDIT=e32.exe "%s" -n%d
351 % Epsilon (Lugaru) http://www.lugaru.com/
352 % TEXEDIT="c:\Program Files\eps90\bin\e32.exe" +%d %s
353
354 % For unix

```

```

355 %
356 % vi, vim, NEdit, (X)Emacs, pico, jed
357 % TEXEDIT = vi      +%d %s
358 % TEXEDIT = vim     +%d %s
359 % TEXEDIT = nedit   +%d %s
360 % TEXEDIT = xemacs +%d %s
361
362 %(x)fte:
363 % TEXEDIT = xfte    -1%d %s
364
365
366 %-----
367 % Write .log/.dvi/etc. files here, if the current directory is unwritable.
368 % TEXMFOUTPUT = /tmp
369
370 % If a dynamic file creation fails, log the command to this file, in
371 % either the current directory or TEXMFOUTPUT. Set to the
372 % empty string or 0 to avoid logging.
373 MISSFONT_LOG = missfont.log
374
375 % Set to a colon-separated list of words specifying warnings to suppress.
376 % To suppress everything, use TEX_HUSH = all; this is equivalent to
377 % TEX_HUSH = checksum:lostchar:readable:special
378 TEX_HUSH = none
379
380 % Enable system commands via \write18{...}?
381 shell_escape = f
382
383 % Allow TeX \openout/\openin on filenames starting with '.' (e.g., .rhosts)?
384 % a (any)      : any file can be opened.
385 % r (restricted) : disallow opening "dotfiles".
386 % p (paranoid)  : as 'r' and disallow going to parent directories, and
387 %               restrict absolute paths to be under $TEXMFOUTPUT.
388 openout_any = p
389 openin_any = a
390 % Allow TeX, MF, and MP to parse the first line of an input file for
391 % the %&format construct.
392 parse_first_line = t
393
394 % Allow TeX, eTeX, Omega to include 'src:' specials in the dvi file.
395 % These specials are used by viewers to jump from the viewer into
396 % the editor at the right page/lineno.
397 % Possible values : none auto cr display hbox math par parent vbox
398 src_specials = none
399
400 % Disable search on multiple suffixes filenames. In many case, when 'foo.bar'
401 % is looked for, you do not want to look for 'foo.bar.tex' before. This flag
402 % disables searching for standard suffixes if the file name has already an
403 % extension of 3 characters. Default value is true (old behaviour).
404 % allow_multiple_suffixes = f
405
406 % Enable the mktex... scripts by default? These must be set to 0 or 1.
407 % Particular programs can and do override these settings, for example
408 % dvips's -M option. Your first chance to specify whether the scripts
409 % are invoked by default is at configure time.
410 %
411 % These values are ignored if the script names are changed; e.g., if you
412 % set DVIPSMMAKEPK to 'foo', what counts is the value of the environment
413 % variable/config value 'FOO', not the 'MKTEXPK' value.
414 %
415 % MKTEXTEX = 0
416 % MKTEXPK = 0
417 % MKTEXMF = 0
418 % MKTEXTFM = 0
419 % MKOCP = 0
420 % MKOFM = 0
421
422 % What MetaPost runs to make MPX files. This is passed an option -troff
423 % if MP is in troff mode. Set to '0' to disable this feature.
424 MPXCOMMAND = makempx
425
426
427 % Part 3: Array and other sizes for TeX (and Metafont and MetaPost).
428 %

```

```

429 % If you want to change some of these sizes only for a certain TeX
430 % variant, the usual dot notation works, e.g.,
431 % main_memory.hugetex = 20000000
432 %
433 % If a change here appears to be ignored, try redumping the format file.
434
435 % Memory. Must be less than 8,000,000 total.
436 %
437 % main_memory is relevant only to initex, extra_mem_* only to non-ini.
438 % Thus, have to redump the .fmt file after changing main_memory; to add
439 % to existing fmt files, increase extra_mem_*. (To get an idea of how
440 % much, try \tracingstats=2 in your TeX source file;
441 % web2c/tests/memtest.tex might also be interesting.)
442 %
443 % To increase space for boxes (as might be needed by, e.g., PiCTeX),
444 % increase extra_mem_bot.
445 %
446 % For some xy-pic samples, you may need as much as 700000 words of memory.
447 % For the vast majority of documents, 60000 or less will do.
448 %
449 main_memory = 263000 % words of inmemory available; also applies to inmf&mp
450 extra_mem_top = 0 % extra high memory for chars, tokens, etc.
451 extra_mem_bot = 0 % extra low memory for boxes, glue, breakpoints, etc.
452
453 % Words of font info for TeX (total size of all TFM files, approximately).
454 font_mem_size = 200000
455
456 % Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
457 font_max = 1000
458
459 % Extra space for the hash table of control sequences (which allows 10K
460 % names as distributed).
461 hash_extra = 0
462
463 % Max number of characters in all strings, including all error messages,
464 % help texts, font names, file names, control sequences.
465 % These values apply to TeX and MP.
466 pool_size = 125000
467
468 % Minimum pool space after TeX/MP's own strings; must be at least
469 % 25000 less than pool_size, but doesn't need to be nearly that large.
470 string_vacancies = 25000
471 max_strings = 15000 % max number of strings
472 pool_free = 5000 % min pool space left after loading .fmt
473
474 % Hyphenation trie. As distributed, the maximum is 65535; this should
475 % work unless 'unsigned short' is not supported or is smaller than 16
476 % bits. This value should suffice for UK English, US English, French,
477 % and German (for example). To increase, you must change
478 % 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
479 % the trie will then consume four bytes per entry, instead of two.
480 %
481 % US English, German, and Portuguese: 30000.
482 % German: 14000.
483 % US English: 10000.
484 %
485 trie_size = 64000
486
487 % Buffer size. TeX uses the buffer to contain input lines, but macro
488 % expansion works by writing material into the buffer and reparsing the
489 % line. As a consequence, certain constructs require the buffer to be
490 % very large. As distributed, the size is 50000; most documents can be
491 % handled within a tenth of this size.
492 buf_size = 200000
493
494 % These are Omega-specific.
495 ocp_buf_size = 20000 % character buffers for ocp filters.
496 ocp_stack_size = 10000 % stacks for ocp computations.
497 ocp_list_size = 1000 % control for multiple ocp's.
498
499 % These work best if they are the same as the I/O buffer size, but it
500 % doesn't matter much. Must be a multiple of 8.
501 dvi_buf_size = 16384 % TeX
502 gf_buf_size = 16384 % MF

```

```

503
504 % It's probably inadvisable to change these. At any rate, we must have:
505 % 45 < error_line < 255;
506 % 30 < half_error_line < error_line - 15;
507 % 60 <= max_print_line;
508 % These apply to Metafont and MetaPost as well.
509 error_line = 79
510 half_error_line = 50
511 max_print_line = 79
512 stack_size = 300 % simultaneous input sources
513 save_size = 4000 % for saving values outside current group
514 param_size = 500 % simultaneous macro parameters
515 max_in_open = 15 % simultaneous input files and error insertions
516 hyph_size = 1000 % number of hyphenation exceptions, >610 and <32767
517 nest_size = 100 % simultaneous semantic levels (e.g., groups)
518 obj_tab_size = 200000 % PDF objects
519
520
521 main_memory.mpost = 1000000
522
523 main_memory.context = 1500000
524 hash_extra.context = 25000
525 pool_size.context = 750000
526 string_vacancies.context = 45000
527 max_strings.context = 55000
528 pool_free.context = 47500
529 nest_size.context = 500
530 param_size.context = 1500
531 save_size.context = 5000
532 stack_size.context = 1500
533 obj_tab_size.context = 256000
534
535 main_memory.hugetex = 1100000
536 param_size.hugetex = 1500
537 stack_size.hugetex = 1500
538 hash_extra.hugetex = 15000
539 string_vacancies.hugetex = 45000
540 pool_free.hugetex = 47500
541 nest_size.hugetex = 500
542 save_size.hugetex = 5000
543 pool_size.hugetex = 500000
544 max_strings.hugetex = 55000
545
546 main_memory.cslatex = 1100000
547 param_size.cslatex = 1500
548 stack_size.cslatex = 1500
549 hash_extra.cslatex = 15000
550 string_vacancies.cslatex = 45000
551 pool_free.cslatex = 47500
552 nest_size.cslatex = 500
553 save_size.cslatex = 5000
554 pool_size.cslatex = 500000
555 max_strings.cslatex = 55000
556 font_mem_size.cslatex= 400000
557
558 main_memory.hugelatex = 1100000
559 param_size.hugelatex = 1500
560 stack_size.hugelatex = 1500
561 hash_extra.hugelatex = 15000
562 string_vacancies.hugelatex = 45000
563 pool_free.hugelatex = 47500
564 nest_size.hugelatex = 500
565 save_size.hugelatex = 5000
566 pool_size.hugelatex = 500000
567 max_strings.hugelatex = 55000
568 font_mem_size.hugelatex= 400000
569
570 main_memory.jadetex = 1500000
571 param_size.jadetex = 1500
572 stack_size.jadetex = 1500
573 hash_extra.jadetex = 50000
574 string_vacancies.jadetex = 45000
575 pool_free.jadetex = 47500
576 nest_size.jadetex = 500

```

```

577 save_size.jadetex = 5000
578 pool_size.jadetex = 500000
579 max_strings.jadetex = 55000
580
581 main_memory.pdfjadetex = 2500000
582 param_size.pdfjadetex = 1500
583 stack_size.pdfjadetex = 1500
584 hash_extra.pdfjadetex = 50000
585 string_vacancies.pdfjadetex = 55000
586 pool_free.pdfjadetex = 47500
587 nest_size.pdfjadetex = 500
588 save_size.pdfjadetex = 5000
589 pool_size.pdfjadetex = 500000
590 max_strings.pdfjadetex = 55000
591
592 main_memory.xmltex = 1500000
593 param_size.xmltex = 1500
594 stack_size.xmltex = 1500
595 hash_extra.xmltex = 50000
596 string_vacancies.xmltex = 45000
597 pool_free.xmltex = 47500
598 nest_size.xmltex = 500
599 save_size.xmltex = 10000
600 pool_size.xmltex = 500000
601 max_strings.xmltex = 55000
602
603 main_memory.pdfxmltex = 2500000
604 param_size.pdfxmltex = 1500
605 stack_size.pdfxmltex = 1500
606 hash_extra.pdfxmltex = 50000
607 string_vacancies.pdfxmltex = 45000
608 pool_free.pdfxmltex = 47500
609 nest_size.pdfxmltex = 500
610 save_size.pdfxmltex = 10000
611 pool_size.pdfxmltex = 500000
612 max_strings.pdfxmltex = 55000
613
614 font_mem_size.pdflatex = 210000
615 main_memory.pdflatex = 1500000
616 param_size.pdflatex = 3000
617 stack_size.pdflatex = 3000
618 hash_extra.pdflatex = 15000
619 string_vacancies.pdflatex = 55000
620 pool_free.pdflatex = 47500
621 nest_size.pdflatex = 500
622 pool_size.pdflatex = 500000
623 save_size.pdflatex = 5000
624 max_strings.pdflatex = 55000
625
626 main_memory.pdfelatex = 1500000
627 param_size.pdfelatex = 1500
628 stack_size.pdfelatex = 1500
629 hash_extra.pdfelatex = 15000
630 string_vacancies.pdfelatex = 45000
631 pool_free.pdfelatex = 47500
632 nest_size.pdfelatex = 500
633 pool_size.pdfelatex = 500000
634 save_size.pdfelatex = 5000
635 max_strings.pdfelatex = 55000
636

```