

# Le Guide Pratique de T<sub>E</sub>X Live, 5ème édition

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

Avril 2000

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Les extensions de T <sub>E</sub> X . . . . .	2
<b>2</b>	<b>Structure et contenu du CD-ROM</b>	<b>3</b>
2.1	L'arborescence TDS . . . . .	3
<b>3</b>	<b>Installation et utilisation sous Unix</b>	<b>4</b>
3.1	Executer T <sub>E</sub> X Live à partir du CD-ROM . . . . .	4
3.2	Installer T <sub>E</sub> X Live à partir du disque dur . . . . .	6
3.3	Installation individuelle de composants T <sub>E</sub> X Live sur le disque dur . . . . .	8
3.4	Le programme texconfig . . . . .	9
<b>4</b>	<b>Installation et utilisation sous Windows</b>	<b>10</b>
4.1	Utiliser T <sub>E</sub> X Live à partir du CD-ROM . . . . .	10
4.2	Installation sur le disque dur . . . . .	10
<b>5</b>	<b>Installation sur un nouveau système Unix</b>	<b>11</b>
5.1	Matériel requis . . . . .	11
5.2	Configuration . . . . .	11
5.3	Exécuter make . . . . .	12
5.4	Dernières étapes de configuration . . . . .	12
<b>6</b>	<b>Le guide d'utilisation du système Web2c</b>	<b>13</b>
6.1	Kpathsea et la recherche de fichiers . . . . .	14
6.2	Les bases de données des fichiers . . . . .	18
6.3	Options à l'exécution . . . . .	26
<b>7</b>	<b>Historique et remerciements</b>	<b>26</b>
<b>8</b>	<b>Versions futures</b>	<b>28</b>

## List of Tables

1	Kpathsea file types	20
---	---------------------	----

## 1 Introduction

Cette documentation décrit les principales caractéristiques du CD-ROM **T<sub>E</sub>X Live 5** —une distribution T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X pour les systèmes d'exploitations Unix et Windows32 incluant les logiciels T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, METAFONT, MetaPost, Makeindex et BIB<sub>T</sub>E<sub>X</sub> ainsi qu'une bibliothèque étendue de macros, de fontes et de documentations conformes à la norme TDS, (*T<sub>E</sub>X Directory Standard*)—qui peut être utilisée par tout type (ou presque) d'installation T<sub>E</sub>X.

Les programmes de cette distribution de T<sub>E</sub>X utilisent une version modifiée de l'implémentation Web2c (version 7.3), ce qui facilite l'utilisation T<sub>E</sub>X et permet de profiter pleinement de l'efficacité et des possibilités de personnalisation de la bibliothèque Kpathsea créée par Karl Berry et Olaf Weber. Les programmes sont utilisables soit à partir du CD-ROM soit à partir du disque dur une fois copiés.

La plupart des systèmes exécutables du CD-ROM comprennent un large panel de pilotes et d'outils pour T<sub>E</sub>X tels que dvips (driver PostScript), xdvi (prévisualisation d'un texte sous X Window), dvilj (pilote pour HP LaserJet), lacheck (Contrôleur syntaxique de L<sup>A</sup>T<sub>E</sub>X), tex4ht (convertisseur T<sub>E</sub>X/HTML), dviconcat et dviselect, dv2dt et dt2dv (dvi vers ASCII et vice versa), et les utilitaires PostScript d'Angus Duggan.

### 1.1 Les extensions de T<sub>E</sub>X

Les systèmes exécutables du **T<sub>E</sub>X Live** contiennent trois extensions expérimentales au T<sub>E</sub>X normal.

1.  $\epsilon$ -T<sub>E</sub>X, qui ajoute un jeu petit mais puissant de nouvelles primitives et les extensions T<sub>E</sub>X--X<sub>E</sub>L pour l'écriture de gauche à droite;  $\epsilon$ -T<sub>E</sub>X est, en mode normal, 100% compatible avec le T<sub>E</sub>X standard. Pour plus de détails se référer au fichier `texmf/doc/etex/base/etex_man.tex` du CD-ROM.
2. pdfT<sub>E</sub>X, qui en option peut écrire au format Acrobat PDF au lieu du format DVI. Vous trouverez le manuel d'utilisation dans `texmf/doc/pdftex/pdftex-1.pdf`. Le fichier `texmf/doc/pdftex/samplepdf/samplepdf.tex` montre un exemple d'utilisation. L'extension hyperref de L<sup>A</sup>T<sub>E</sub>X possède une option « `pdf tex` » qui active tous les mécanismes nécessaires.
3.  $\Omega$  (Omega), qui fonctionne en utilisant Unicode et des caractères 16 bits, ce qui lui permet de travailler directement et simultanément avec presque toutes les écritures du monde entier. L'extension supporte aussi le chargement dynamique d'OTPs ( $\Omega$  Translation Processes) qui permet à l'utilisateur d'appliquer sur des flux d'entrées quelconques des transformations complexes qu'il aura définies. Voir la documentation `texmf/doc/omega/base/doc-1.8.tex` du CD-ROM.

La version (2.1) d' $\epsilon$ -T<sub>E</sub>X est stable même si de nouvelles fonctions seront ajoutées dans les prochaines versions. pdfT<sub>E</sub>X (version 0.14f) et  $\Omega$  (version 1.11) sont en développement continu et les versions du CD-ROM sont celles de la mi-mars 2000.

## 2 Structure et contenu du CD-ROM

Les répertoires importants à la racine du CD-ROM sont énumérés ci-dessous.

**bin** Les programmes de la famille  $\TeX$ , rangés dans des sous-répertoires selon les plateformes.

**tldoc** Documentation du  **$\TeX$  Live**.

**FAQ** Les Foire Aux Questions, en allemand, anglais et français.

**info** Documentation du système  $\TeX$  au format GNU info.

**man** La documentation du système  $\TeX$  accessible sous Unix avec la commande `man` ou équivalent.

**source** Le code source de tous les programmes, incluant la distribution Web2c de  $\TeX$  et METAFONT. Il s'agit d'une archive au format `tar` compressé.

**support** Quelques logiciels *non* installés par défaut en relation avec  $\TeX$  tels que Musix $\TeX$ , des programmes de maintenance, et une distribution complète de Ghostscript version 5.50.

**texmf** Le répertoire principal contenant les macros, les fontes et la documentation.

**usergrps** Des informations sur les associations d'utilisateurs de  $\TeX$ .

Deux programmes d'installation pour les systèmes Unix sont aussi fournis : `install-cd.sh` et `install-pkg.sh`. Ils seront étudiés aux chapitres 3 p. 4.

### 2.1 L'arborescence TDS

L'arborescence `texmf` de  **$\TeX$  Live** consiste en une variété de « collections », chacune comprenant un ensemble de composants dont le nombre peuvent dépasser les 400 sur le CD-ROM. L'utilisateur peut dans l'installation normale copier une « collection » entière du CD-ROM vers le disque dur mais il est aussi possible d'installer juste un composant d'une « collection ». Les « collections » sont :

**ams** Les macros et les fontes de l'American Mathematical Society.

**bibtex** Les bases de données et les styles de BIB $\TeX$ .

**doc** Les guides généraux et la documentation sous plusieurs formats, HTML et PDF inclus.

**dvips** Support pour le pilote de traduction DVI vers PostScript de Thomas Rokicki.

**etex** Support pour  $\epsilon$ - $\TeX$ .

**fonts** fichiers de fontes : sources, métriques, PostScript et bitmap.

**formats** Eplain, Rev $\TeX$ , phyzzx, texsis, alateX, text1, lollipop, etc.

**generic** Macros supplémentaires utilisables dans tous les formats.

**graphics** Composants pour tout ce qui est graphique.

**lang** Support pour les langues autre que l'anglais.

**latex** L<sup>A</sup>T<sub>E</sub>X, les outils officiels et les contributions à L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

**metapost** Support pour MetaPost.

**omega** Support pour Ω.

**pdftex** Support pour pdfT<sub>E</sub>X

**plain** Macros pour plain T<sub>E</sub>X.

**systems** Fichiers binaires pour les plateformes Unix et Win32.

**texlive** Matériel de base pour la distribution.

Chaque collection peut être divisée en 3 types : *de base* (1), *recommandée* (2) et *autre* (3). Ainsi, les composants de la collection latex1 représentent ce qu'un utilisateur doit avoir pour commencer avec L<sup>A</sup>T<sub>E</sub>X, les composants de latex2 sont recommandés pour la plupart des utilisateurs et ceux de latex3 sont optionnels. Le répertoire texmf/tpm contient des fichiers de description pour chaque composant (fichiers utilisés par les programmes d'installation).

### 3 Installation et utilisation sous Unix

L'exploitation du CD-ROM T<sub>E</sub>X Live s'effectue de trois manières :

1. Vous pouvez monter le CD-ROM sur votre système, changer votre PATH en conséquence, puis tout exécuter à partir du CD-ROM. Cela prend très peu d'espace disque et vous donne un accès immédiat à tout le contenu du CD-ROM. C'est une solution parfaitement acceptable par exemple pour des PC sous linux bien que les performances obtenues ne soient pas optimales.
2. Vous pouvez installer une partie du système sur votre disque dur. C'est la plupart du temps le meilleur choix si vous disposez d'un espace disque suffisant (10 megabytes au minimum, mais 100 megabytes pour le système recommandé).
3. Vous pouvez installer un jeu de composants choisi pour travailler sur un système T<sub>E</sub>X déjà présent ou sur un T<sub>E</sub>X Live que vous auriez déjà installé.

Chaque choix est décrit en détail dans les paragraphes qui suivent.

#### 3.1 Executer T<sub>E</sub>X Live à partir du CD-ROM

L'organisation de Web2c induit que vous pourrez exécuter les programmes simplement en indiquant dans votre PATH le sous-répertoire approprié dans le répertoire bin du CD-ROM. Les fichiers nécessaires seront alors tous visibles sans autres modifications. La liste des systèmes proposés et des sous-répertoires correspondants sont les suivants :

**ATTENTION :** Ce CD-ROM est au format ISO 9660 (High Sierra), avec les extensions « Rock Ridge and Joliet ». Votre système doit être capable de reconnaître ce type d'extensions pour une utilisation optimale du CD-ROM. Consultez votre documentation concernant la commande `mount` pour voir si c'est le cas. Si votre réseau local est constitué de machines différentes, regardez si vous pouvez monter le CD-ROM sur une machine qui supporte Rock Ridge et utiliser ce point de montage pour les autres machines.

Les systèmes Linux, FreeBSD, Sun, SGI et DEC Alpha ne devraient pas poser de problèmes d'utilisation du CD-ROM. Nous apprécierions beaucoup de recevoir des conseils détaillés de la part d'utilisateurs ayant réussi l'installation sur d'autres systèmes pour les futures versions de cette documentation.

La suite de ce document suppose que vous avez installé et monté le CD-ROM avec une compatibilité Rock Ridge parfaite.

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

Vous serez peut être ennuyé lors de création de fontes ou de changement de configuration par le fait que vous ne pouvez pas modifier les fichiers sur le CD-ROM. Cependant vous pouvez maintenir en parallèle une structure modifiable de répertoires  $\TeX$  sur votre disque dur qui sera visitée avant l'arborescence du CD-ROM. Le chemin choisi par défaut est `texmf-localconfig` sur le CD-ROM (chemin qui n'existe pas !), donc vous *devez* l'écraser en définissant la variable d'environnement `VARTEXMF`.

Ainsi, les utilisateurs d'un PC avec un processeur Intel et sous Linux avec un environnement `sh` ou `bash` peuvent monter le CD-ROM **TeX Live** au point `/cdrom` avec la commande :

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

Ensuite ils doivent désigner le répertoire contenant les fichiers exécutables correspondants à l'architecture donnée en initialisant la variable `PATH`.

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

Par commodité, ces commandes peuvent aussi être exécutées à partir du fichiers `.profile`.

En cas de doute, demandez de l'aide à l'administrateur système de votre réseau pour monter le CD-ROM ou pour vous indiquer le répertoire à utiliser pour votre système.

Les fichiers nécessaires au fonctionnement seront installés sur le disque dur la première fois où ils seront appelés. Il est recommandé d'exécuter dès le début le fichier `texconfig` pour tout mettre en place et vérifier si tout marche bien.

## 3.2 Installer T<sub>E</sub>X Live à partir du disque dur

Toutes les étapes nécessaires pour installer tout ou partie de la distribution sont réalisées en montant le CD-ROM puis en tapant à partir du répertoire racine :

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

(Sur certains systèmes Unix, vous devrez utiliser sh5 ou bash). Le programme d'installation commence par accéder à la liste des collections et des composants du CD-ROM puis essaie de deviner le système présent dans votre ordinateur. Il devrait commencer par vous retourner les messages suivants :

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

Ensuite il vous montrera un écran de contrôle principal (Figure 1), qui vous permettra de choisir 4 options :

1. le type du système tournant sur votre ordinateur ou celui que vous avez décidé d'installer;
2. la collection que vous voulez installer et à quel niveau : *de base*, *recommandé* ou *autre*;
3. les répertoires du disque dur où il faut installer les fichiers;
4. quelques options supplémentaires;

Vous choisissez les options en entrant au clavier une lettre ou un nombre suivi de return. Dans l'exemple, le système ELF sous linux est détecté. Par défaut, la collection de niveau *recommandé* est celle choisie ainsi que le répertoire `/usr/local` d'installation. Notons que l'espace disque requis pour l'installation choisie est affiché. Si vous faites l'installation suggérée, vous avez besoin d'environ 100 megabytes d'espace disque tandis que l'installation *de base* n'en a besoin que de 10 et vous pouvez ensuite l'enrichir avec les composants qui vous sont nécessaires.

À partir du répertoire d'installation choisi, le programme d'installation déposera les fichiers binaires dans un sous-répertoire de `bin` et les fichiers de support dans une arborescence ayant `texmf` pour répertoire de base.

Les cases à cocher au niveau des options vous permettent de décider d'un autre endroit pour la création et l'écriture des fontes (permet de restreindre l'accès en lecture seulement de l'arborescence principal pour l'ensemble des utilisateurs) et permettent également de générer des liens symboliques vers les pages `man` et GNU d'information.

Si vous choisissez `<C>` pour les collections, vous verrez apparaître la liste des collections disponibles, le niveau d'installation choisi et la place disque requise (Figure 2). Vous pouvez moduler les niveaux d'installations pour chaque collection de *none* jusqu'à *all*. Vous pouvez soit effectuer un choix pour toute les collections d'un coup, soit choisir un niveau pour une collection particulière (Figure 3).

Lorsque votre sélection est effectuée, retournez à l'écran principal et sélectionnez le démarrage de l'installation. Le programme d'installation considèrera chaque collection pour le système demandé, pointerà la liste des fichiers sur le CD-ROM et construira la liste principale des fichiers à transférer. Ceux-ci seront copiés sur votre disque dur. Si vous avez demandé l'installation d'un système, une séquence d'initialisation se déclenchera (création de fichiers de format...). Lorsqu'elle sera terminée, il vous restera à indiquer à votre système d'exploitation le chemin pour trouver le sous répertoire `bin` de T<sub>E</sub>X en modifiant votre path et vous pourrez alors utiliser T<sub>E</sub>X. Si vous le voulez, vous pouvez déplacer

```

=====> 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:

```

Figure 1: écran de contrôle principal

```

      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

```

Figure 2: Sélection des collections

d'un cran i.e. de /usr/local/bin/alpha-osf4.0 à /usr/local/bin. Si vous le faites cependant vous devez éditer le fichier texmf/web2c/texmf.cnf (voir Annexe 9) et changer la ligne qui se lit :

```
TEXMFMAIN = $SELFAUTOPARENT
```

en

```
TEXMFMAIN = $SELFAUTODIR
```

```

Collection: Fonts
=====
Fonts, including metrics, virtual fonts and sources
=====
<N> No packages
<B> Basic packages           [ 1023 kB]
<E> Basic + Recommended packages [ 51447 kB]
<A> All packages            [127417 kB]
=====
<R> return to collection menu
<Q> quit
Enter command:

```

Figure 3: Choix par l'utilisateur dans une collection

Si vous transférez les fichiers exécutables dans un autre répertoire vous devez alors éditer `TEXMFMAIN` pour indiquer explicitement le répertoire de base de l'arborescence et donner à la variable `TEXMFCNF` de votre environnement à `$TEXMFMAIN/texmf/web2c`.

### 3.3 Installation individuelle de composants **T<sub>E</sub>X Live** sur le disque dur

Vous voudrez peut être utiliser le CD-ROM **T<sub>E</sub>X Live** soit pour mettre à jour une installation existante, soit pour ajouter des éléments à une installation effectuée à partir du CD-ROM. Le programme d'installation est prévu pour la première installation seulement et vous devrez alors utiliser le programme `install-pkg.sh` de votre CD-ROM. Une fois le CD-ROM monté, déplacez-vous sur son répertoire racine et entrez :

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

L'installation supporte 9 options qui correspondent à : choisir les composants individuels à installer pour les 4 premières, installer la totalité de la collection (i.e.,ams2), saisir le nom du répertoire où le CD-ROM est monté, saisir le nom du répertoire contenant la liste des fichiers (normalement les 2 dernières options sont détectées automatiquement).

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

L'exécution est contrôlée par 4 options supplémentaires : avec les 2 premières, vous pouvez exclure la documentation et les fichiers sources de l'installation. La suivante empêche l'exécution de `mktexlsr` prévue par défaut pour achever la reconstruction des bases de données et la dernière ne fait rien mais envoie la liste des fichiers qui seront installés.

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

Finalement, vous pouvez demander à ce que le programme crée dans un répertoire donné une archive tar au lieu de transférer les fichiers.

`--archive=name`

Donc, si vous voulez voir la liste des fichiers qui seraient créés par l'installation du composant fancyhdr, votre commande et la sortie seraient les suivantes :

```
>> 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
```

On peut donner d'autres exemples d'utilisation :

- Installer le composant natbib :

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

- Installer le composant alg sans création de fichier sources et de documentation :

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

- Installer toutes les composants de la collection Plain T<sub>E</sub>X avec le niveau *autre*.

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

- placer tous les fichiers nécessaires à PSTricks dans un fichier tar du répertoire /tmp :

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

### 3.4 Le programme texconfig

Une fois que le programme d'installation a transféré tous les fichiers vers leur place définitive, vous pouvez configurer votre système pour l'adapter à vos propres besoins en utilisant le programme texconfig. Il peut être appelé après chaque modification de votre installation et fonctionne en mode texte ou graphique (ce qui nécessite le programme dialog, fourni dans les composants binaires). Il doit-être utilisé pour toute maintenance, que ce soit pour l'installation d'une nouvelle imprimante ou pour la reconstruction d'une base de donnée. Les deux modes de fonctionnement bénéficient d'un fichier d'aide destiné a vous guider.

## 4 Installation et utilisation sous Windows

Ce chapitre s'applique uniquement pour les systèmes Windows 9x, NT ou Windows 2000. Si vous êtes sous Windows 3.1, vous devrez installer à la main  $\epsilon\text{mT}\epsilon\text{X}$  qui se trouve dans le répertoire `systems` du CD-ROM.

Votre système Windows doit aussi supporter l'extension Microsoft Joliet pour pouvoir lire le CD-ROM. Jeter un coup d'oeil dans explorer pour voir s'il montre les noms longs de fichier. Si ce n'est le cas vous ne pourrez utiliser  $\text{T}\epsilon\text{X}$  directement depuis le CD-ROM.

$\text{T}\epsilon\text{X}$  sous Win32 contient Windvi, un nouveau prévisualisateur similaire au xdvi d'Unix. Sa documentation se trouve dans [texmf/doc/html/windvi/windvi.html](http://texmf/doc/html/windvi/windvi.html).

### 4.1 Utilisation à partir du CD-ROM

Vous pouvez exécuter les programmes  $\text{T}\epsilon\text{X}$  directement à partir du CD-ROM avec toutes les fontes et les macros nécessaires mais en contrepartie d'une plus grande lenteur d'exécution que si vous installez sur le disque dur. Pour pouvoir travailler, il suffit de modifier quelques variables d'environnement et de créer quelques petits répertoires auxiliaires sur votre disque dur. Ces répertoires contiendront les fichiers de paramétrage qui permettent à l'utilisateur de modifier le comportement des programmes et de générer des fichiers de format. De plus, les fichiers de polices générés automatiquement seront stockés au même endroit. Toutes ces étapes préliminaires sont réalisées par le programme `TeXsetup.exe` que l'on lance depuis le répertoire `setupw32/` du CD-ROM. Une fois que le programme est lancé et que la racine des répertoires auxiliaires est choisie, sélectionner l'option 'Run CD'. Une fois l'installation terminée, vous devrez probablement relancer Windows. Ensuite, vous pourrez lancer les programmes depuis une console, ou bien utiliser un éditeur  $\text{T}\epsilon\text{X}$  qui prendra en charge leur pilotage et fournira une interface agréable.

### 4.2 Installation sur le disque dur

Le CD-ROM lancera l'installation automatiquement lors de son insertion; on peut également lancer manuellement le programme `setupw32/TeXSetup.exe`. Ce programme lit la liste des collections et des composants à installer à partir du CD-ROM. Il permet aussi de sélectionner le niveau d'installation pour chaque collection (voir chapitre 2.1 pour une description des collections et des composants) et d'omettre des parties de la documentation et/ou des fichiers sources des composants au cas où votre espace disque disponible est limité.

**Attention: les utilisateurs souhaitant installer  $\text{T}\epsilon\text{X}$  sur un système de fichiers FAT doivent être avertis de la place disque importante requise. La taille des clusters pour FAT est de 32ko, alors que  $\text{T}\epsilon\text{X}$  Live installe des milliers de fichiers dont la taille est de l'ordre de 1ko à 5ko qui occupent chacun un cluster ! Une grande quantité de place est donc perdue. Nous ne saurions qu'inciter les utilisateurs à passer à FAT32 ou à NTFS qui autorisent des tailles de cluster plus faibles.**

Le nom du répertoire dans lequel seront installées la distribution et la configuration locale vous seront demandés. De plus vous aurez la possibilité d'installer un éditeur  $\text{T}\epsilon\text{X}$  et le visualiseur PostScript Ghostscript.

Le programme d'installation fournit la possibilité de télécharger depuis l'Internet certains composants qui ne sont pas fournis sur le CD-ROM pour des problèmes de licence. Cette option ne devrait être activée que si vous disposez d'une connexion rapide : certains composants sont volumineux.

Une fois l'installation complétée, il vous sera éventuellement demandé de redémarrer Windows et vous pouvez soit demander l'exécution des programmes T<sub>E</sub>X en ligne ou par l'intermédiaire d'un menu avec éditeur tel que NTEmacs ou WinEdt (si vous avez opté pour son installation).

Après une installation sur disque dur, vous aurez la possibilité d'ajouter un ou plusieurs composants, voire des collections. Pour ce faire, utiliser l'option 'Add TeX Package' du menu 'Démarrer' → 'TeX Live' → 'Maintenance'.

Enfin, la commande TeXSetup --help affiche la liste des options disponibles.

## 5 Installation sur un nouveau système Unix

Si vous avez un système Unix ne correspondant à aucuns des exécutables fourni, vous aurez besoin de compiler T<sub>E</sub>X et tous ses programmes satellites à partir de zéro. Cela n'est pas aussi difficile qu'il y paraît. Le contenu du répertoire source du CD-ROM est le seul ingrédient dont vous aurez besoin.

Vous devez tout d'abord transférer l'arborescence contenant tous les fichiers supports à partir du CD-ROM T<sub>E</sub>X Live (faire une installation de base sans choisir le types de système d'exploitation et donc les fichiers exécutables).

### 5.1 Matériel requis

Vous aurez besoin de 100 megabytes d'espace disque pour compiler la totalité de T<sub>E</sub>X. Vous aurez aussi besoin d'un compilateur C ANSI, de l'utilitaire make, d'un générateur d'analyseur lexical (lexical scanner) et d'un générateur d'analyseur syntaxique (parser generator). Les utilitaires GNU (gcc, GNU make, m4, flex, bison) ont été testés avec succès sur un grand nombre de systèmes. gcc-2.7.\* flex-2.4.7 and GNU make-3.72.1 ou des versions plus récentes devraient convenir. Vous pouvez utiliser d'autres compilateurs C et d'autres programmes make si vous maîtrisez correctement la programmation Unix pour vous sortir des problèmes. La commande uname doit renvoyer une valeur sensée.

### 5.2 Configuration

Premièrement, décompactez dans votre disque les fichiers sources tar-compressés, du répertoire source et placez vous dans le répertoire ou vous avez transféré l'ensemble. Décidez du répertoire racine ou sera placée l'installation, i.e. /usr/local ou /usr/local/TeX. Bien entendu, il doit être identique au répertoire d'où part l'arborescence contenant les fichiers supports.

Maintenant, exécutez configure afin de commencer le processus de compilation :

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

Le répertoire 'prefix' est le répertoire de base de l'arborescence. Les sous répertoires qui seront utilisés sont les suivants (où \$TEXDIR désigne le répertoire de base) :

\\$TEXDIR/man	les pages du manuel Unix	
\\$TEXDIR/share/texmf	répertoire de base pour les fontes	
	macros, etc	Vous pouvez omettre la partie 'share/'
\\$TEXDIR/info	manuels GNU sous format info	
\\$TEXDIR/bin/\\$PLATFORM	exécutables	

pour indiquer l'emplacement de texmf, car dirname\$TEXDIR/share/texmf et \$TEXDIR/texmf sont tout deux détectés par configure. Si votre choix est différent, vous devez indiquer ce répertoire avec l'option

--datadir du programme configure. Si vous ne voulez pas créer de répertoire  $\backslash\$PLATFORM$  relatif au choix du système (i.e. copier directement les exécutables dans  $\$TEXDIR/bin$ ), exécutez configure avec l'option `verb--disable-multiplatform`.

Tappez `./configure -help`, le programme vous renseignera sur les options supplémentaires possibles (comme enlever l'installation des extensions optionnelles telles que  $\Omega$  or  $\epsilon$ - $\TeX$ ).

### 5.3 Exécuter make

Assurez vous que la variable `noclobber` soit vide et entrez :

```
>> make world
```

et relax (Max)...

Il peut être utile de diriger la sortie dans un fichier log en entrant :

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

Avant de penser que tout va bien, vérifiez si le fichier log contient des messages d'erreur (le make de GNU renvoie toujours le message "Error:" à chaque erreur d'exécution) et si tous les exécutables ont été créés.

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

```
>> ls | wc
```

Le nombre de mots doit être 214, ce qui correspond à une installation complète.

Si vous avez besoins de droits spéciaux pour exécuter `make install`, vous pouvez effectuer 2 processus make séparément :

```
>> make all
```

```
>> su
```

```
>> make install strip
```

### 5.4 Dernières étapes de configuration

Mettez à jour votre variable d'environnement `PATH` pour inclure le répertoire contenant les exécutables installés (e.g. `/usr/local/TeX/bin/mips-sgi-irix6.5`). De la même façon, les variables `MANPATH` et `INFOPATH` doivent prendre en compte les sous répertoires `$TEXDIR/man` et `$TEXDIR/info`.

Le programme `texconfig` peut paramétrer les fichiers de césure, la taille du papier, la commande d'impression, le mode `METAFONT` etc qui seront utilisés par défaut. Vous pouvez lancer cette commande de manière interactive et regarder les options offertes ou encore vous pouvez simplement entrer :

```
>> texconfig help
```

Par exemple, si vous n'utilisez pas de papier au format A4 mais lettre, vous pouvez avoir l'option `lettre` par défaut en entrant :

```
>> texconfig dvips paper letter
```

```
>> texconfig xdvi paper us
```

---

<sup>0</sup>ajout du traducteur...

## 6 Le guide d'utilisation du système Web2c

Web2c contient un ensemble de programmes relatifs à  $\text{T}_{\text{E}}\text{X}$ , c-à-d  $\text{T}_{\text{E}}\text{X}$  lui-même, METAFONT, MetaPost,  $\text{BIB}_{\text{E}}\text{X}$ , etc. L'implémentation a été dirigée par Tomas Rokicki qui, en 1987, a développé un premier système  $\text{T}_{\text{E}}\text{X}$ -to-C en adaptant les fichiers sous Unix, qui était principalement le travail de Howard Trickey et Pavel Curtis. Tim Morgan assura la maintenance du système, dont le nom fut remplacé durant cette période par Web-to-C. En 1990, Karl Berry reprit le travail, assisté par des douzaines de collaborateurs, et en 1997 il passa le relais à Olaf Weber. La dernière version en date est Web2c Version 7.3, qui parut en mars 1999, et qui forme les bases du présent  **$\text{T}_{\text{E}}\text{X}$  Live** CD-ROM.

Le système Web2c 7.3 fonctionne sur Unix, Windows 3.1, 9x/NT, DOS, Amiga et de nombreux autres systèmes d'exploitation. Il utilise les sources originales de Knuth pour  $\text{T}_{\text{E}}\text{X}$  et les autres programmes de base écrits en web pour les traduire en langage C. De plus, le système offre un large éventail de macros et de fonctions développées dans le but d'améliorer le logiciel  $\text{T}_{\text{E}}\text{X}$  original. Les composantes du noyau de  $\text{T}_{\text{E}}\text{X}$  sont:

bibtex Gère les bibliographies.

dmp troff vers MPX (dessins MetaPost).

dvicopy Expansion de fontes virtuelles.

dvitomp DVI vers MPX (dessins MetaPost).

dvitype DVI vers texte.

gftodvi Fontes génériques vers épreuves.

gftopk Fontes génériques vers fontes bitmap.

gftype GF vers texte.

makempx Typographie des étiquettes MetaPost.

mf Creation de fontes.

mft Mise en page de code source METAFONT.

mpost Création de diagrammes techniques.

mpto Extraction d'étiquettes MetaPost.

newer Comparaison de dates de modification (fichiers).

patgen Creating hyphenation patterns.

pktofg Fontes bitmap vers fontes generiques.

pktype PK vers texte.

pltotf PL vers TFM.

pooltype Affiche les fichiers web pool.

tangle web vers Pascal.

tex Typographie.

fttopl TFM vers PL.

vftovp Fontes virtuelles vers PL virtuelles.

vptovf PL virtuelles vers fontes virtuelles.

weave web vers  $\text{\TeX}$ .

La syntaxe et les fonctions précises de ces programmes sont décrites dans la documentation des composants individuels ou bien dans le manuel Web2c lui-même. Toutefois, la connaissance d'un certain nombre de principes gouvernant l'ensemble de la famille de programmes peut vous aider à exploiter de façon optimale votre installation Web2c. Tous ces programmes suivent les options standard de GNU :

–help : imprime le sommaire de l'utilisation. –verbose : imprime le rapport détaillé du processus .  
–version : imprime le numéro de version, puis quitte.

Pour localiser les fichiers, les programmes Web2c utilisent la bibliothèque de recherche Kpathsea. Cette bibliothèque utilise une combinaison de variables d'environnement et un certain nombre de fichiers de configuration pour optimiser la recherche dans l'arborescence  $\text{\TeX}$ . Web2c 7.3 peut chercher plusieurs arborescences de fichiers simultanément, ce qui est utile si l'on souhaite maintenir la distribution standard de  $\text{\TeX}$  et les extensions locales dans deux arborescences distinctes. Afin d'accélérer la recherche de fichiers la racine de chaque arborescence possède un fichier ls-R, contenant une entrée donnant le nom et le chemin pour chaque fichier situé sous la racine.

## 6.1 Kpathsea et la recherche de fichiers

Décrivons en premier lieu le mécanisme de recherche de la bibliothèque Kpathsea. Nous appelons *chemin de recherche* une liste séparée avec « deux-points » ou « point-virgule », dont les éléments, appelés *éléments de chemin*, sont des noms de répertoires. Un chemin de recherche peut provenir de plusieurs sources. Pour rechercher un fichier “my-file” le long d'un chemin “. : /dir”, Kpathsea vérifie chaque élément du chemin (d'abord ./my-file, puis /dir/my-file) et retourne la (ou les) bonne(s) correspondance(s).

Afin d'optimiser l'adaptation à tous les systèmes d'exploitation, Kpathsea peut utiliser dans les noms de fichiers des séparateurs différents de “deux-points (“:”)” et “slash (“/”)” pour les systèmes non-Unix.

Pour vérifier un élément de chemin particulier  $p$ , Kpathsea vérifie d'abord si une base de données existante (voir page 18) s'applique à  $p$ , i.e. si la base de données se trouve dans un répertoire qui est un préfixe de  $p$ . Si oui, la spécification du chemin est comparée avec le contenu de la base de données.

Si la base de données n'existe pas, ne s'applique pas à cet élément de chemin ou ne contient aucune correspondance, la recherche est lancée sur tout le système de fichiers (si cela n'a pas été interdit par une commande commençant par “!!” et si le fichier cherché doit exister). Kpathsea construit la liste de répertoires qui correspondent à cet élément de chemin, puis cherche le fichier dans chaque élément de cette liste.

La condition “le fichier doit exister” entre en jeu avec les fichiers “.vf” et les fichiers d'entrée lus par la commande  $\text{\TeX}$ 's \openin. De tels fichiers peuvent ne pas exister (par exemple cmr10.vf), il est donc inutile de les rechercher sur le disque. De plus, si vous n'actualisez pas le fichier ls-R lors de l'installation d'un nouveau fichier “.vf”, il ne sera jamais trouvé. Chaque élément de chemin est

alors vérifié : d'abord dans la base de données puis sur le disque. Si une correspondance est trouvée la recherche s'arrête et le résultat est retourné.

Bien que l'élément de chemin le plus simple et banal soit un nom de répertoire, Kpathsea supporte d'autres formats dans les chemins de recherche : des valeurs par défaut différentes pour chaque programme, des noms de variables d'environnement, des valeurs de fichiers de configuration, les répertoires de l'utilisateur et la recherche récursive de sous-répertoires. Nous disons alors que Kpathsea *étend* un élément, c'est-à-dire que Kpathsea transforme toutes ces spécifications en noms de répertoires basiques. Ce qui est décrit dans les sections suivantes.

Notez que si le nom de fichier cherché est absolu ou explicitement relatif, c'est-à-dire commençant par “/”, “./” ou “. ./”, Kpathsea vérifie seulement si ce fichier existe.

### 6.1.1 Les différentes sources

Un chemin de recherche peut provenir de plusieurs sources. Dans l'ordre dans lesquels Kpathsea les utilise :

1. une variable d'environnement définie par l'utilisateur, par exemple `TEXINPUTS`. Les variables d'environnement avec une extension attachée (nom de programme) sont d'abord prises en compte : par exemple, si “`latex`” est le nom du programme exécuté, `TEXINPUTS.latex` passera avant `TEXINPUTS`.
2. Un fichier de configuration de programme spécifique, par exemple une ligne “`S /a:/b`” dans le `config.ps` de `dvips`.
3. Un fichier de configuration `texmf.cnf` de Kpathsea contenant une ligne telle que “`TEXINPUTS=/c:/d`” (voir ci-dessous).
4. La valeur par défaut à la compilation

Vous pouvez voir chacune de ces valeurs pour un chemin de recherche donné en utilisant l'option de débogage voir page 23).

### 6.1.2 Fichiers de configuration

Kpathsea lit dans les *fichiers de configuration à l'exécution* appelés `texmf.cnf`, les chemins de recherche et d'autres définitions. Le chemin pour accéder à ces fichiers dans l'arborescence est stocké dans `TEXMFCNF` (par défaut ces fichiers se trouvent dans le sous-répertoire `texmf/web2c`). *Tous* les fichiers `texmf.cnf` se trouvant dans le chemin de recherche vont être lus dans l'ordre et les définitions provenant des premiers fichiers lus écraseront celles des fichiers suivants. Par exemple, avec un chemin tel que `.: $TEXMF`, les définitions du fichier `./texmf.cnf` écrasent celles présentes dans `$TEXMF/texmf.cnf`.

En complément de la lecture du descriptif du format du fichier `texmf.cnf` ci-dessous, on doit aussi se référer à l'annexe 9, commençant page 28, qui liste le fichier `texmf.cnf` donné avec la distribution sur le CDRom.

- Les commentaires sont signalés par un “`%`” et continuent jusqu'à la fin de la ligne.
- Les lignes vierges sont ignorées.

- Un `\` à la fin d’une ligne joue le rôle d’un lien entre deux lignes, c’est à dire que la ligne courante se poursuit à la ligne suivante. Dans ce cas, les espaces présents au début de la ligne suivante ne sont pas ignorés.

- Toutes les autres lignes sont de la forme:

`variable [.prognome] [=] value`

où le “=” et les espaces autour sont optionels.

- Le nom de la “*variable*” peut contenir n’importe quels caractères exceptés les espaces, “=”, ou “.”, mais on recommande de se cantonner à “A-Za-z\_” pour éviter les problèmes.
- Si “.prognome” est présent, sa définition s’applique seulement si le programme exécuté se nomme *prognome* ou *prognome.exe*. Ceci permet à différentes distributions de T<sub>E</sub>X d’avoir des chemins de recherche différents par exemple.
- “*value*” peut contenir n’importe quels caractères exceptés “%” et “@”. L’option “*\$var.prog*” n’est pas disponible à droite; à la place, on doit utiliser une variable supplémentaire. Un “;” dans “*value*” est compris comme un “:” si on travaille sous Unix; ceci est très utile car permettant d’avoir un seul `texmf.cnf` pour les systèmes Unix, MSDOS et Windows.
- Toutes les définitions sont lues avant tout désarchivage ou décompactage, de telle façon que les variables puissent être référencées avant d’être définies.

Un extrait d’un fichier de configuration illustrant les points précédents est présenté ci-dessous:

```

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 d’un chemin de recherche

Kpathsea reconnaît certain caractères et constructions spéciaux dans les chemins de recherche, semblables à ceux disponibles dans les shells Unix. Comme exemple général, le chemin complexe, `~$USER/{foo,bar}//baz`, étend la recherche vers tous les sous-répertoires situés sous les répertoires `foo` et `bar` dans le répertoire utilisateur de `$USER` contenant un répertoire ou un fichier appelé `baz`. Ces expansions sont explicitées dans les sections suivantes.

### 6.1.4 Expansion par défaut

Si le chemin de recherche le plus prioritaire (voir “Origines des chemins” en page 15) contient un “:” *supplémentaire* (c’est à dire, en début ou fin de ligne ou double), Kpathsea insère à cet endroit le prochain chemin de priorité immédiatement inférieure défini. Si ce chemin inséré possède un “:” supplémentaire, le même processus se répète pour le prochain chemin prioritaire. Par exemple, étant donnée une variable d’environnement définie comme suit

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

et la valeur de TEXINPUTS d'après le fichier `texmf.cnf` étant

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

alors la valeur finale utilisée pour la recherche sera:

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

Comme il serait inutile d'insérer la valeur par défaut à plusieurs endroits, Kpathsea applique la substitution à seulement un ":" supplémentaire et laisse les autres inchangés: il cherche d'abord un ":" en début de ligne, puis en fin et enfin pour un double ":".

### 6.1.5 Expansion spécifiée par les accolades

Une option utile est l'expansion par le biais des accolades, ce qui signifie par exemple que,  $v\{a,b\}w$  va permettre la recherche dans  $vaw:vbw$ . Les définitions emboîtées sont autorisées. Ceci peut être utilisé pour établir des hiérarchies  $\TeX$  multiples, en attribuant une liste entre accolades à `$TEXMF`. Par exemple, dans `texmf.cnf`, on trouve (ligne 52) la définition suivante:

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

Avec ceci, on peut écrire quelque chose comme

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

ce qui signifie que, après avoir cherché dans le répertoire courant, l'arborescence complète `$TEXMFLOCAL/tex` (sur le disque) et ensuite l'arborescence `!!$TEXMFMAIN/tex` (définie dans le fichier de référence *ls-R seulement*) seront inspectés. C'est un moyen pratique permettant d'utiliser en parallèle deux distributions  $\TeX$ , une première "gelée" (sur un CD-ROM, par exemple) et une autre régulièrement mise à jour avec de nouvelles versions quand elles deviennent disponibles. En utilisant la variable `$TEXMF` dans toutes les définitions, on est toujours sûr d'inspecter d'abord l'arborescence la plus récente.

### 6.1.6 Expansion des sous-répertoires

Deux ou plus slashes consécutifs dans une partie d'un chemin suivant un répertoire  $d$  est remplacé par tous les sous-répertoires de  $d$ : d'abord les sous-répertoires directement présents dans  $d$ , ensuite les sous-répertoires de ceux-ci, et ainsi de suite. A chaque niveau, l'ordre dans lequel les répertoires sont inspectés est *indéterminé*.

Dans le cas où l'on spécifie une partie de nom de fichier après le "//", seuls sont inclus les sous-répertoires dont le nom correspond. Par exemple, `/a//b` va correspondre aux répertoires `/a/1/b`, `/a/2/b`, `/a/1/1/b`, et ainsi de suite, mais pas à `/a/b/c` ou `/a/1`.

Des "//" multiples et successifs dans un chemin sont possibles, mais "//" au début d'un chemin est ignoré.

### 6.1.7 Liste des caractères spéciaux et leur signification: un récapitulatif

La liste suivante récapitule la signification des caractères spéciaux dans les fichiers de configuration de Kpathsea.

- : Séparateur dans un chemin de recherche; au début ou à la fin d'un chemin, il remplace le chemin par défaut.
- ; Séparateur dans les systèmes non-Unix (joue le rôle de :).
- \$ Substitue le contenu d'une variable.
- ~ Représente le répertoire racine de l'utilisateur.
- {...} Expansion par les accolades, c'est à dire par exemple, a{1,2}b devient a1b:a2b.
- // La recherche concernera aussi les sous-répertoires (peut être inséré n'importe où dans un chemin sauf au début).
- % Début d'un commentaire.
- \ Caractère de continuation de ligne (permet les entrées sur de multiples lignes).
- !! Recherche *seulement* dans la base de données pour localiser le fichier et *pas* sur le disque.

## 6.2 Les bases de données des fichiers

Kpathsea a une certaine profondeur d'investigation pour minimiser les accès disque durant les recherches. Néanmoins, dans le cas de distributions comprenant beaucoup de répertoires, inspecter chaque répertoire possible pour un fichier donné peut durer excessivement longtemps (ceci est typiquement le cas si plusieurs centaines de répertoires de polices de caractères doivent être parcourus.) En conséquence, Kpathsea peut utiliser un fichier-base de données construite au préalable et appelée `named 1s-R` qui fait correspondre les fichiers à leur répertoire, ce qui permet d'éviter une recherche exhaustive sur le disque.

Un deuxième fichier-base de données `aliases` permet de donner des noms différents aux fichiers listés dans `1s-R`. Ceci peut aider à adapter ses fichiers source aux conventions "8.3" pour les noms de fichiers dans les systèmes similaires à DOS.

### 6.2.1 La base de données des fichiers

Comme expliqué ci-dessus, le nom du principal fichier-base de données doit être `1s-R`. Vous pouvez en mettre un dans la racine de chaque architecture  $\TeX$  dans votre installation que vous désirez être inspectée (`$TEXMF` par défaut); la plupart des sites ont seulement une architecture. Kpathsea cherche les fichiers `1s-R` dans le chemin spécifié dans la variable `TEXMFDBS`.

La façon recommandée de créer et mettre à jour le fichier "`1s-R`" est d'exécuter le script `mktexlsr` inclus dans la distribution. Il est invoqué par les divers scripts "`mktex`". En principe, ce script exécute uniquement la commande

```
cd /your/texmf/root && 1s -LAR ./ >1s-R
```

en supposant que la commande `1s` de votre système produit le bon format de sortie (le `1s` de GNU convient parfaitement). Pour s'assurer que la base de données est toujours à jour, le meilleur moyen est de la reconstruire en utilisant la table des `cron`, de telle façon que le fichier `1s-R` prenne automatiquement en compte les changements dans les fichiers installés—peut-être après avoir installé ou mis à jour un composant  $\LaTeX$ —.

Si un fichier n'est pas trouvé dans la base de données, par défaut Kpathsea décide de chercher sur le disque. Par contre, si un élément du chemin commence par “!”, *seulement* la base de données sera inspectée pour cet élément, et jamais le disque.

### 6.2.2 kpsewhich: programme de recherche dans une arborescence

Le programme kpsewhich effectue une recherche dans une arborescence indépendamment de toute application. Il peut être considéré comme une sorte de `find` pour localiser des fichiers dans les arborescences T<sub>E</sub>X (ceci est largement utilisé dans les scripts “`mktex`”... venant avec la distribution).

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

Les options spécifiées dans “*option*” peut commencer soit par “-” ou “-”, et n'importe quelle abréviation claire est acceptée.

Kpathsea considère tout argument non optionnel dans la ligne de commande comme un nom de fichier et renvoie la première occurrence trouvée. Il n'y a pas d'option pour renvoyer tous les fichiers ayant un nom particulier (vous pouvez utiliser le “`find`” d'Unix pour cela).

Le options les plus importantes sont décrites ci-après.

`-dpi=num` Définit la résolution à “*num*”; ceci affecte seulement les recherches des “`gf`” et “`pk`”. “`-D`” est un synonyme pour assurer la compatibilité avec `dvips`. Le défaut est 600.

`-format=name`

Définit le format de recherche à “*name*”. Par défaut, le format est supposé d'après le nom de fichier. Pour les formats qui n'ont pas de suffixe non-ambigu qui leur est associé, comme les fichiers de support MetaPost et les fichiers de configuration `dvips`, vous devez spécifier le nom correspondant à celui trouvé dans la première colonne de la Table 1, qui liste les noms reconnus à ce jour, une description, les variables d'environnement associées<sup>1</sup>, et les extensions de nom de fichier autorisées.

Les deux dernières entrées de la Table 1 sont des cas spéciaux, où les chemins et les variables d'environnement dépendent du nom du programme: le nom de la variable est construit en convertissant le nom du programme en majuscules, et en y ajoutant `INPUTS`.

Les variables d'environnement sont définies par défaut dans le fichier de configuration `texmf.cnf`. C'est seulement quand on veut redéfinir une ou plusieurs variables dans ce fichier que l'on a intérêt à les définir alors explicitement dans son environnement d'exécution.

Notez que les options de “`-format`” et “`-path`” sont mutuellement exclusives.

`-mode=string`

Définit le nom du mode comme étant “*string*”; ceci affecte seulement la recherche des “`gf`” et des “`pk`”. Pas de défaut: n'importe quel mode sera trouvé.

`-must-exist`

Fait tout ce qui est possible pour trouver les fichiers, ce qui inclut rechercher sur le disque. Par défaut, seulement la base de données `ls-R` est inspectée, dans un souci d'efficacité.

---

<sup>1</sup>Vous pouvez trouver des définitions pour ces variables d'environnement dans le fichier `texmf.cnf` (page 28)

Table 1: Kpathsea file types

<i>Noms</i>	<i>Description</i>	<i>Variables</i>	<i>Suffixes</i>
afm	Adobe font metrics	AFMFonts	.afm
base	Metafont memory dump	MFBASES, TEXMFINI	.base
bib	BIB $\TeX$ bibliography source	BIBINPUTS, TEXBIB	.bib
bst	BIB $\TeX$ style files	BSTINPUTS	.bst
cnf	Runtime configuration files	TEXMFCNF	.cnf
dvips config	dvips configuration files, e.g., config.ps and psfonts.map	TEXCONFIG	.map
fmt	$\TeX$ memory dump	TEXFORMATS, TEXMFINI	.fmt, .efmt, .efm
gf	generic font bitmap	Fonts, GFFONTS, GLYPHFonts, TEXFonts	.gf
graphic/figure	Encapsulated PostScript figures	TEXPICTS, TEXINPUTS	.eps, .epsi
ist	makeindex style files	TEXINDEXSTYLE, INDEXSTYLE	.ist
ls-R	Filename databases	TEXMFDBS	
map	Fontmaps	TEXFONTMAPS	.map
mem	MetaPost memory dump	MPMEMS, TEXMFINI	.mem
mf	Metafont source	MFINPUTS	.mf
mfpool	Metafont program strings	MFPOOL, TEXMFINI	.pool
mft	MFT style file	MFTINPUTS	.mft
mp	MetaPost source	MPINPUTS	.mp
mppool	MetaPost program strings	MPPPOOL, TEXMFINI	.pool
MetaPost support	MetaPost support files, used by DMP	MPSUPPORT	
ocp	$\Omega$ compiled process files	OCPINPUTS	.ocp
ofm	$\Omega$ font metrics	OFMFonts, TEXFonts	.ofm, .tfm
opl	$\Omega$ property lists	OPLFonts, TEXFonts	.opl
otp	$\Omega$ translation process files	OTPIINPUTS	.otp
ovf	$\Omega$ virtual fonts	OVFFONTS, TEXFonts	.ovf
ovp	$\Omega$ virtual property lists	OVPPFonts, TEXFonts	.ovp
pk	fontes bitmap compressées	programFonts (program being XDVI, etc.), PKFonts, TEXPKS, GLYPHFonts, TEXFonts	.pk
PostScript header	downloadable PostScript	TEXPSHEADERS, PSHEADERS	.pro, .enc
tex	$\TeX$ source	TEXINPUTS	.tex, .cls, .sty, .clo, .def
TeX system documentation	Documentation files for the $\TeX$ system	TEXDOCS	
TeX system sources	Source files for the $\TeX$ system	TEXSOURCES	
texpool	$\TeX$ program strings	TEXPOOL, TEXMFINI	.pool
tfm	$\TeX$ font metrics	TFMFonts, TEXFonts	.tfm
Troff fonts	Troff fonts, used by DMP	TRFonts	
truetype fonts	TrueType outline fonts	TTFONTS	.ttf, .ttc
type1 fonts	Type 1 PostScript outline fonts	T1Fonts, T1INPUTS, TEXPSHEADERS, DVIPSHEADERS	.pfa, .pfb
type42 fonts	Type 42 PostScript outline fonts	T42Fonts	
vf	virtual fonts	VFFONTS, TEXFonts	.vf
web2c files	Web2c support files	20WEB2C	
other text files	fichiers texte utilisés par 'foo'	FOOINPUTS	
other binary files	fichiers binaire utilisés par 'foo'	FOOINPUTS	

- path=string  
recherche dans le chemin “*string*” (séparé par des deux-points comme d’habitude), au lieu de supposer le chemin à partir du nom de fichier. “//” et toutes les expansions habituelles sont supportées. Les options “-path” et “-format” sont mutuellement exclusives.
- progrname=name  
Définit le nom de programme comme étant “*name*”. ceci peut affecter les chemins de recherche via l’option “. *progrnam*” dans les fichiers de configuration. Le d’éfaut est “kpsewhich”.
- show-path=name  
montre le chemin utilisé pour la recherche des fichiers de type “*name*”. C’est soit une extension de fichier (“.pk”, “.vf”, etc.), soit un nom de fichier, comme avec l’option “-format”.
- debug=num  
Définit les options de débogages comme étant “*num*”.

### 6.2.3 Exemples d’utilisation

Jetons un coup d’oeil à Kpathsea en action.

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

Nous recherchons le fichier `article.cls`. Puisque le suffixe “.cls” est non-ambigu, nous n’avons pas besoin de spécifier que nous voulons rchercher un fichier de type “tex” (répertoires des fichiers sources de  $\TeX$ ). Nous le trouvons dans le sous-répertoire `tex/latex/base` du répertoire racine “`TEXMF`”. De la même façon, le suffixe non-ambigu permet de trouver facilement les autres.

```
>> 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
```

Le dernier est une base de données de bibliographie pour  $\text{BIB}\TeX$  servant aux articles de *TUGBoat*.

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

Les fichiers de glyphes de fontes bitmaps, de type `.pk`, sont utilisés pour l’affichage par des programmes comme `dvips` et `xdvi`. Rien n’est renvoyé dans ce cas puisque il n’y a pas de fichiers Computer Modern “.pk” pré-crées sur nos systèmes (puisque nous utilisons les versions Type1 sur le CD-ROM).

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

Pour les fichiers Computer Modern étendues, nous avons dû créer les fichiers “.pk”, et puisque le mode METAFONT par défaut sur notre installation est `ljfour` avec une résolution de base de 600 dpi (dots per inch), cette instance est retournée.

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

Dans ce cas, lorsque l’on spécifie que nous sommes intéressés par une résolution de 300dpi (`-dpi=300`) nous voyons qu’aucune fontes pour cette résolution n’est disponible sur le système. En fait, un programme comme `dvips` ou `xdvi` ne s’en préoccuperaient pas et créeraient les fichiers .pk à la résolution demandée en utilisant le script `mktexpk`.

Regardons à présent l’entête `dvips` et les fichiers de configuration. Regardons en premier le fichier communément utilisé `tex.pro` pour le support de T<sub>E</sub>X, avant de regarder le fichier de configuration générique (`config.ps`) et la carte des fontes PostScript `psfonts.map`. Comme le suffixe “.ps” est ambigu, nous devons spécifier quel type particulier du fichier `config.ps` nous considérons (“`dvips config`”).

```
>> 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
```

Regardons plus en détail les fichiers de support URW Times PostScript. Leur nom dans le schéma d’appellation de fontes de Berry est “utm”. Le premier fichier que nous voyons est le fichier de configuration, qui contient le nom du fichier d’organisation:

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

Le contenu de ce fichier est

```
p +utm.map
```

qui pointe vers le fichier `utm.map`, que nous cherchons à localiser ensuite.

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

Ce fichier d’organisation définit les noms des fichiers contenant les fontes PostScript de Type1 dans la collection URW. Son contenu ressemble à (nous ne montrons qu’une partie des lignes):

```
utmb8r NimbusRomNo9L-Medi    ... <utmb8a.pfb  
utmbi8r NimbusRomNo9L-MediItal... <utmbi8a.pfb  
utmr8r NimbusRomNo9L-Regu    ... <utmr8a.pfb  
utmri8r NimbusRomNo9L-ReguItal... <utmri8a.pfb  
utmbo8r NimbusRomNo9L-Medi    ... <utmb8a.pfb  
utmro8r NimbusRomNo9L-Regu    ... <utmr8a.pfb
```

Par exemple, prenons le cas de Times Regular `utmr8a.pfb` et trouvons sa position dans l’arborescence `texmf` en utilisant une recherche applicable aux fichiers de fontes de Type1:

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

Il devrait être clair d'après ces quelques exemples qu'il est facile de trouver l'endroit où se cache un fichier donné. C'est particulièrement important si vous suspectez que c'est, pour une raison quelconque, la mauvaise version du fichier qui est utilisée, puisque `kpsewhich` va vous montrer le premier fichier trouvé.

#### 6.2.4 Opérations de débogage

Quelquefois il est nécessaire de savoir comment un programme référence les fichiers. Pour permettre cela d'une manière pratique, `Kpathsea` offre plusieurs niveaux de débogage:

- 1 `stat` calls (tests de fichier). Lors d'une exécution utilisant une base de données `ls-R` à jour, ce niveau ne devrait donner presque aucune information en sortie.
- 2 Références aux tables de découpage (comme la base de données `ls-R`, les fichiers d'organisation, les fichiers de configuration).
- 4 Opérations d'ouverture et de fermeture des fichiers.
- 8 Information générale sur la localisation des types de fichier recherchés par `Kpathsea`. Ceci est utile pour trouver où a été défini le chemin particulier pour un fichier.
- 16 La liste des répertoires pour chaque élément du chemin (utilisé uniquement en cas de recherche sur le disque).
- 32 Recherche de fichiers.

Une valeur de `-1` activera toutes les options ci-dessus; dans la pratique, vous utiliserez probablement ces niveaux si vous avez besoin de déboguer.

De la même façon, avec le programme `dvips`, en utilisant une combinaison d'option de débogage, on peut savoir en détail où les différents fichiers sont pris. En parallèle, lorsqu'un fichier n'est pas trouvé, le log du débogage montre les différents répertoires dans lesquels le programme va chercher tel ou tel fichier, donnant ainsi des indices sur le problème.

Généralement, comme la plupart des programmes appellent la bibliothèque `Kpathsea` en interne, on peut sélectionner une option de débogage en utilisant la variable d'environnement `KPATHSEA_DEBUG`, et en la fixant égale à (une combinaison) des valeurs décrites dans la liste ci-dessus.

Considérons comme exemple un petit fichier source  $\LaTeX$ , `hello-world.tex`, dont le contenu est le suivant.

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

Ce petit fichier utilise simplement la fonte `cmr10`, aussi allons voir comment `dvips` prépare le fichier PostScript (nous voulons utiliser la version Type1 des fontes Computer Modern, d'où l'option `-Pcms`).

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

Dans ce cas, nous avons combiné le niveau 4 de débogage de dvips (chemins des fontes) avec l'option d'expansion des éléments du chemin de Kpathsea (voir dvips Reference Manual, [texmf/doc/html/dvips/dvips\\_toc.html](http://texmf/doc/html/dvips/dvips_toc.html)). Nous obtenons quelque chose comme (nous avons réarrangé la sortie pour un affichage commode):

dvips commence par localiser ses fichiers de fonctionnement. D'abord, `texmf.cnf` est trouvé, ce qui donne les définitions pour les chemins de recherche servant à localiser les autres fichiers, ensuite le fichier base de données `ls-R` (pour optimiser la recherche des fichiers) et le fichier `aliases`, qui permet de déclarer plusieurs noms (e.g., un nom DOS de type "8.3" court et une version longue plus naturelle) pour le même fichier. Ensuite dvips continue en cherchant le fichier de configuration générique `config.ps` avant de rechercher le fichier de paramétrisation `.dvipsrc` (qui, dans notre cas, *n'est pas trouvé*). Enfin, dvips localise le fichier de configuration pour les fontes PostScript Computer Modern `config.cms` (ceci est lancé par l'option `-Pcms` de la commande dvips). Ce fichier contient la liste des fichiers d'organisation qui définissent la relation entre les noms des fontes selon T<sub>E</sub>X, PostScript et le système de fichiers.

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

dvips va chercher à trouver tous ces fichiers, y compris le fichier d'organisation générique `psfonts.map`, qui est toujours chargé (il contient des déclarations pour les fontes PostScript les plus communément utilisées; voir la dernière partie de la Section 6.2.3 pour plus de détails sur la gestion du fichier d'organisation PostScript).

Arrivé là, dvips s'identifie à l'utilisateur:

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

pour continuer ensuite en cherchant le fichier prolog `texc.pro`,

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

Après avoir trouvé ce fichier, dvips affiche la date et l'heure, et nous informe qu'il va générer le fichier `hello-world.ps`, puis qu'il a besoin du fichier de fonte `cmr10`, et que ce dernier est déclaré comme "resident":

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

Maintenant la recherche concerne le fichier `cmr10.tfm`, qui est trouvé, puis quelques fichiers prolog de plus (non montrés) sont référencés, et finalement le fichier du cas Type1 `cmr10.pfb` de la fonte est localisé et inclus dans le fichier de sortie (voir la dernière ligne).

```

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:/usr/local/TeX/TeX/texmf/web2c:).
kdebug:start search(file=ls-R, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(ls-R) =>/usr/local/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(aliases) => /usr/local/texmf/aliases
kdebug:start search(file=config.ps, must_exist=0, find_all=0,
  path=~/.tex:!!/usr/local/texmf/dvips//).
kdebug:search(config.ps) => /usr/local/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc, must_exist=0, find_all=0,
  path=~/.tex:!!/usr/local/texmf/dvips//).
search(file=/home/goossens/.dvipsrc, must_exist=1, find_all=0,
  path=~/.tex/dvips//:!!/usr/local/texmf/dvips//).
kdebug:search($HOME/.dvipsrc) =>
kdebug:start search(file=config.cms, must_exist=0, find_all=0,
  path=~/.tex/dvips//:!!/usr/local/texmf/dvips//).
kdebug:search(config.cms)
=>/usr/local/texmf/dvips/cms/config.cms

```

Figure 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

```

Figure 5: Finding the prolog file

```

kdebug:start search(file=cmr10.tfm, must\_exist=1, find\_all=0,

```

```

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 Options à l'exécution

Web2c 7.3 offre la possibilité de contrôler à l'exécution bon nombre des paramètres concernant la mémoire et en particulier la taille des tableaux utilisés; il suffit de spécifier les valeurs dans le fichier `texmf.cnf` qui est lu par Kpathsea. Le listing de ce fichier est fourni en annexe 9, à partir de la page 28; les paramètres en question se trouvent dans la troisième partie de ce fichier. Les variables les plus importantes sont:

`main_memory` Nombre total de mots mémoire disponibles pour T<sub>E</sub>X, METAFONT and MetaPost. Vous devez générer un nouveau fichier de format pour chaque nouveau paramétrage. Par exemple, vous pouvez générer une version large de T<sub>E</sub>X, et appeler le fichier de format `hugutex.fmt`. En utilisant la méthode supportée par Kpathsea (suffixer la variable par le nom du programme), la valeur particulière de la variable `main_memory` destinée à ce fichier de format sera lue dans le fichier `texmf.cnf` (comparer la valeur générique à la valeur spécifique pour le format `hugutex`).

`extra_mem_bot` Espace supplémentaire pour certaines structures de données de T<sub>E</sub>X : boîtes, glu, points d'arrêt ... Surtout utile si vous utilisez P<sub>I</sub>C<sub>T</sub>E<sub>X</sub> par exemple.

`font_mem_size` Nombre de mots mémoire disponibles pour décrire les polices. C'est plus ou moins l'espace occupé par les fichiers TFM lus.

`hash_extra` Espace supplémentaire pour la table de hachage des noms de séquences de contrôle. Environ 10000 de ces noms peuvent être stockés dans la table principale; si vous avez un document très volumineux avec beaucoup de références croisées, il se peut que ce ne soit pas suffisant. Vous pouvez remarquer que aussi bien `hugutex` que `pdflatex` demandent 15,000 séquences de contrôle supplémentaires (la valeur par défaut est 0).

évidemment, cette possibilité ne remplace pas une vraie allocation dynamique des tableaux et de la mémoire, mais puisque c'est complexe à implémenter, ces paramètres lus à l'exécution fournissent un compromis pratique qui procure une certaine souplesse.

## 7 Historique et remerciements

Cette distribution sous forme de CD-ROM est un effort groupé du T<sub>E</sub>X Users Group, du UK T<sub>E</sub>X Users Group, du French T<sub>E</sub>X Users (GUTenberg), et du German T<sub>E</sub>X Users (DANTE e.V.), avec l'aide des

groupes d'utilisateurs de  $\text{\TeX}$  Tchèques/Slovaques, Néerlandais, Indien et Polonais. La discussion commença à la fin de 1993 quand le Groupe des Utilisateurs Néerlandais de  $\text{\TeX}$  commençait à travailler à sont 4All $\text{\TeX}$  CD-ROM pour les utilisateurs de MS-DOS, et on espérait à ce moment sortir un seul CD-ROM pour tous les systèmes. Ceci était un but bien trop ambitieux, mais il permit la naissance du 4All $\text{\TeX}$  CD-ROM, projet couronné de succès, mais aussi du TUG Technical Council working group on a  $\text{\TeX}$  Directory Structure, qui spécifia la gestion sous une forme logique des fichiers de support pour  $\text{\TeX}$ . La mouture finale du TDS fut publiée dans le numéro de Décembre 1995 des *TUGboat*, et il était clair depuis un certain temps qu'un produit souhaitable était de proposer une structure modèle sur CD-ROM. Le CD-ROM que vous possédez est le résultat direct des délibérations du groupe de travail. Il était aussi clair que le succès des 4All $\text{\TeX}$  CD-ROM montrait que les utilisateurs d'Unix trouveraient leur bonheur avec une distribution plus simple, et ceci a été l'autre direction du  **$\text{\TeX}$  Live**.

Nous avons entrepris de créer un nouveau CD-ROM TDS Unix à l'automne 1995, et avons rapidement choisi le  $\text{\TeX}$  de Thomas Esser comme étant la configuration idéale, car il supportait déjà le multiplateforme et il avait été construit en gardant à l'esprit la portabilité entre les systèmes de fichier. Thomas accepta d'aider, et commença sérieusement à travailler au début de 1996. La première édition sortie en Mai 1996. Au début de 1997, Karl Berry finit une nouvelle distribution de son composant Web2c, qui incluait presque toutes les caractéristiques que Thomas Esser avait ajoutées dans  $\text{\TeX}$ , et nous avons décidé de baser la deuxième édition du CD-ROM sur le standard Web2c, en y ajoutant le script `texconfig` de  $\text{\TeX}$ . La troisième édition du CD-ROM était basée sur une version majeure de Web2c, 7.2, par Olaf Weber; en même temps, une nouvelle version révisée de  $\text{\TeX}$  était accomplie, et  **$\text{\TeX}$  Live** partage presque toutes ses caractéristiques. La quatrième édition a suivi le même schéma, en utilisant une nouvelle version de  $\text{\TeX}$ , et une nouvelle version de Web2c (7.3).

Pour la cinquième édition (Mars 2000), beaucoup de parties du CD-ROM ont été vérifiées et révisées, des centaines d' composants ont été mises à jour. On trouve de nouvelles versions de Omega et de pdf $\text{\TeX}$ , et certains programmes de support de  $\text{\TeX}$  (xdvi, dvips, tex4ht par exemple) ont été améliorés.

Le changement majeur dans cette cinquième édition : tout le logiciel non libre de droits a été supprimé. Tout ce qui se trouve sur ce CD-ROM devrait être compatible la licence Debian (*Debian Free Software Guidelines*) (<http://www.debian.org/intro/free>); nous avons fait de notre mieux pour vérifier les termes des licences de chaque composant, et nous apprécierions que toute erreur nous soit rapportée.

Presque toutes les parties du CD-ROM ont été révisées et vérifiées, avec un soin particulier pour supprimer les fichiers dupliqués, et la classification des composants.

Nous sommes particulièrement reconnaissant à:

- Karl Berry, pour ses conseils, ses encouragements, et (bien sûr) car il donna la distribution originelle de Web2c;
- Mimi Burbank, qui favorisa l'accès au Florida State University Supercomputer Research Institute à différents ordinateurs pour compiler  $\text{\TeX}$ , et joua le rôle de cobaye chaque fois qu'on le lui demanda;
- Kaja Christiansen, qui testa les versions, fournit des remarques essentielles et assista à la documentation et à la compilation;
- Thomas Esser, car sans son merveilleux package  $\text{\TeX}$ , ce CD-ROM n'existerait probablement pas, et dont l'aide continuelle aida à l'améliorer constamment;

- Eitan Gurari, dont le  $\TeX$ 4ht a été utilisé pour créer la version HTML de cette documentation, et qui travailla inlassablement à l'améliorer en peu de temps;
- Fabrice Popineau, qui travailla sans répit à la partie win32 du package, et contribua de différentes manières grâce à des idées, des conseils et du code;
- Staszek Wawrykiewicz, qui contribua énormément aux tests et coordina la distribution polonaise du CD-ROM;
- Olaf Weber, pour son patient assemblage et sa maintenance de Web2c 7.3;
- Graham Williams, du travail duquel dépend le catalogue des composants.

Alain Rabaute, Pascal Quignon, Gerhard Wilhelms, Fabrice Popineau, Janka Chlebíková, Staszek Wawrykiewicz, Erik Frambach, et Ulrik Vieth ont gentiment traduit la documentation dans leurs langues respectives, vérifièrent d'autre documentation, et donnèrent un retour très appréciés.

## 8 Versions futures

*Ce CD-ROM n'est pas un produit parfait!* Nous prévoyons de le renouveler une fois par an, et aimerions fournir plus d'aide, plus de fonctionnalités, plus de programmes d'installation, et (bien sûr) une arborescence améliorée et vérifiée de macros et de fontes. Ce travail est effectué par des volontaires débordés dans la limite de leur temps libre, et beaucoup reste à faire. Si vous pouvez aider, n'hésitez pas à avancer votre nom!

Corrections, suggestions et ajouts pour les revisions futures doivent être envoyés à:

Sebastian Rahtz  
 7 Stratfield Road  
 Oxford OX2 7BG  
 United Kingdom  
[rahtz@tug.org](mailto:rahtz@tug.org)

Mise à jour, notes, et suggestions seront rendues disponibles sur CTAN dans `info/texlive`. Une page WWW pour information et détails pour commander est à <http://www.tug.org/tex-live.html>.

## 9 The texmf.cnf file

```

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:othertext}
36 % expands to /usr/local/mytex:/usr/local/othertext. Instead of the path
37 % separator you can use a comma: /usr/local/{mytex,othertext} also expands
38 % to /usr/local/mytex:/usr/local/othertext. 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 VARTEXFONTS = $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 TEXMFDBS = $TEXMF
91
92 % It may be convenient to define TEXMF like this:
93 % TEXMF = {$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 TEXMFDBS
96 % like this:
97 % TEXMFDBS = $HOMETEXMF;$TEXMFLocal;$TEXMFMAIN;$VARTEXFONTS
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 VARTEXFONTS like this
108 % VARTEXFONTS = $VARTEXMF/fonts
109 % and do not mention it in TEXMFDBS (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 = .;$TEXMF/{etex,tex}/{plain,generic,}//
142
143 % PDFTeX. This form of the input paths is borrowed from tEX. A certain
144 % variant of TDS is assumed here, unaffected by the build variables.
145 TEXINPUTS.pdfetexinfo = .;$TEXMF/{pdfetex,tex}/{texinfo,plain,generic,}//
146 TEXINPUTS.pdfplatex = .;$TEXMF/{pdfetex,tex}/{latex,generic,}//
147 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,tex}/{plain,generic,}//
148 TEXINPUTS.pdfelatex = .;$TEXMF/{pdfetex,pdfetex,etex,tex}/{latex,generic,}//
149 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,pdfetex,etex,tex}/{plain,generic,}//
150
151 % Omega.

```

```

152 TEXINPUTS.lambda = .;$TEXMF/{omega,tex}/{lambda,latex,generic,}//
153 TEXINPUTS.omega = .;$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 = .;$TEXMF/tex/{csplain,plain,generic,}//
161 TEXINPUTS.pdfcslatex = .;$TEXMF/{pdftex,tex}/{cslatex,csplain,latex,generic,}//
162 TEXINPUTS.pdfcsplain = .;$TEXMF/{pdftex,tex}/{csplain,plain,generic,}//
163
164 % Polish
165 TEXINPUTS.platex = .;$TEXMF/tex/{platex,latex,generic,}//
166 TEXINPUTS.pdfmex = .;$TEXMF/{pdftex,tex}/{mex,plain,generic,}//
167 TEXINPUTS.mex = .;$TEXMF/tex/{mex,plain,generic,}//
168 TEXINPUTS.pdfplatex = .;$TEXMF/{pdftex,tex}/{platex,latex,generic,}//
169
170 % french
171 TEXINPUTS.frtex = .;$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 = .;$TEXMF/{pdftex,tex}/{xmltex,latex,generic,}//
185
186 % Sebastian Rahtz' jadetex for DSSSL
187 TEXINPUTS.pdfjadetex = .;$TEXMF/{pdftex,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
208 % Device-independent font metric files.
209 VFFONTS = .;$TEXMF/fonts/vf//
210 TFMFONTS = .;$TEXMF/fonts,$VARTEXFONTS}/tfm//
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 = .;$TEXMF/fonts
224
225 % For texfonts.map and included map files used by mktexpk.
226 % See ftp://ftp.tug.org/tex/fontname.tar.gz.
227 TEXFONTMAPS = .;$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 = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
239 TEXPSHEADERS.pdfcslatex = .;$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.pdfTeX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
247 TEXPSHEADERS.pdfetexinfo = .;$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 = .;$TEXMF/source//
289
290 % allo for compressed files, and various extensions
291 TEXDOCSSUFFIX = .dvi:.ps:.html:.txt
292 TEXDOCSCOMPRESS = .gz:.bz2:.zip:.Z
293 TEXDOCEXT = {$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_prognam.
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\32.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   -l%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 parenend 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 DVIPSMAKEPK 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 inimg&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 ocps.
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
```