

# Guide pratique de T<sub>E</sub>X Live, 7<sup>e</sup> édition

Sebastian Rahtz

sebastian.rahtz@oucs.ox.ac.uk

Mai 2002

## Table des matières

<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	Composants et collections	4
<b>3</b>	<b>Installation et utilisation sous Unix</b>	<b>6</b>
3.1	Exécuter T <sub>E</sub> X Live à partir du CD-ROM	6
3.2	Installer T <sub>E</sub> X Live sur le disque dur	7
3.3	Installation de composants T <sub>E</sub> X Live individuellement sur le disque dur	10
3.4	Le programme texconfig	11
<b>4</b>	<b>Installation et utilisation sous Windows</b>	<b>11</b>
4.1	Le programme TeXLive.exe	12
4.2	Utiliser T <sub>E</sub> X Live à partir du CD-ROM	12
4.3	Installer un éditeur ou un composant support	14
4.4	Installation sur le disque dur	15
<b>5</b>	<b>Maintenance et autres aspects de l'installation de TeXLive sous Windows</b>	<b>19</b>
5.1	Différences entre la version Win32 de Web2c et la version standard	19
5.2	Ajouter des composants à votre système	20
5.3	Enlever T <sub>E</sub> X Live de votre disque dur	21
5.4	Utiliser TeXSetup.exe depuis la ligne de commande	21
5.5	Installation en réseau	22
5.6	Personnalisation de l'installation	22
5.7	Tests	25
5.8	Impression	25
5.9	Pour aller plus loin avec WinShell	26
5.10	Trucs et astuces à propos de la plateforme Win32	28
5.11	En cas de problème	31
5.12	Compiler les fichiers sources	32

<b>6</b>	<b>Installation sur un nouveau système Unix</b>	<b>33</b>
6.1	Prérequis	33
6.2	Configuration	33
6.3	Exécuter make	34
6.4	Dernières étapes de configuration	34
<b>7</b>	<b>Guide d'utilisation du système Web2c</b>	<b>35</b>
7.1	Kpathsea et la recherche de fichiers	36
7.2	Les bases de données	40
7.3	Options à l'exécution	48
<b>8</b>	<b>Remerciements</b>	<b>48</b>
<b>9</b>	<b>Historique</b>	<b>49</b>
<b>10</b>	<b>Versions futures</b>	<b>50</b>
<b>11</b>	<b>The texmf.cnf file</b>	<b>51</b>

## Liste des tableaux

1	Kpathsea file types	41
---	---------------------	----

## 1 Introduction

Cette documentation décrit les principales caractéristiques du CD-ROM **T<sub>E</sub>X Live 7** — une distribution T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X pour les systèmes d'exploitations Unix, Linux, MacOSX 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 B<sub>I</sub>B<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 l'implémentation Web2c (version 7.3.7), ce qui facilite l'utilisation de 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 l'installation faite.

La plupart des systèmes exécutables du CD-ROM comprennent un large éventail de pilotes et d'outils pour T<sub>E</sub>X tels que dvips (pilote pour sortie PostScript), dvipdfm (pilote pour sortie PDF), 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 du programme T<sub>E</sub>X standard.

1.  $\epsilon$ -T<sub>E</sub>X ajoute un jeu, petit mais puissant, de nouvelles primitives et les extensions T<sub>E</sub>X--X<sub>E</sub>T 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 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 « `pdftex` » qui active tous les mécanismes nécessaires.
3.  $\Omega$  (Omega) 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'OTP ( $\Omega$  Translation Processes) qui permet à l'utilisateur d'appliquer sur des flux d'entrée quelconques des transformations complexes qu'il aura définies. Voir la documentation `texmf/doc/omega/base/doc-1.8.tex` du CD-ROM (pas nécessairement à jour).

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 pre-1.00b) est également stable, mais encore en développement.  $\Omega$  (version 1.23) est en cours de développement et la version du CD-ROM est celle de mai 2002.

## 2 Structure et contenu du CD-ROM

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

**bin** Programmes de la famille T<sub>E</sub>X, rangés dans des sous-répertoires selon les plate-formes.

**Books** Exemples tirés de livres sur T<sub>E</sub>X.

**FAQ** Les questions les plus fréquemment posées, en allemand, anglais et français.

**info** Documentation du système T<sub>E</sub>X au format GNU info.

**man** Documentation du système T<sub>E</sub>X accessible sous Unix avec la commande `man` ou équivalent.

**setupw32** Répertoire contenant des fichiers nécessaires pour l'installation et l'utilisation sous Windows (voir la section 4 page 11).

**source** Le code source de tous les programmes, incluant la distribution Web2c de T<sub>E</sub>X et METAFONT. Il s'agit d'une archive au format tar compressé par bzip2.

**support** Quelques logiciels *non* installés par défaut en relation avec T<sub>E</sub>X, des programmes de maintenance et une distribution complète de Ghostscript version 7.05. On y trouvera d'autres programmes (éditeurs orientés T<sub>E</sub>X) qui ne font pas habituellement partie des installations Windows, ils intéresseront les débutants. Ils peuvent être installés par le programme d'installation `TeXSetup.exe` pour Windows.

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

**usergrps** Informations sur les associations d'utilisateurs de T<sub>E</sub>X.

Deux programmes d'installation pour les systèmes Unix sont aussi fournis : `install-cd.sh` et `install-pkg.sh`. Ils seront étudiés au chapitre 3 p. 6.

## 2.1 Composants et collections

L'arborescence `texmf` de **T<sub>E</sub>X Live** consiste en une variété de « collections », chacune comprenant un ensemble de « composants » dont le nombre peut dépasser les sept cents sur le CD-ROM. L'utilisateur peut, dans l'installation normale, copier l'ensemble des « collections » sur le disque dur mais il est aussi possible de n'installer qu'un « composant » d'une « collection ».

Les collections ajoutent de nouvelles fonctionnalités au système T<sub>E</sub>X standard. L'une d'entre elles, nommée « tex-basic », est nécessaire pour presque toutes les tâches de T<sub>E</sub>X et deux autres, appelées « tex-latex » et « tex-pdftex », sont hautement recommandées pour la plupart des utilisateurs. Les collections (définies par les fichiers XML situés dans `texmf/tpm/collections`) accompagnées d'une brève description sont listées ci-dessous :

**tex-basic** Ce sont les fichiers indispensables pour tout système T<sub>E</sub>X. Ils comprennent le format plain de T<sub>E</sub>X, les fontes Computer Modern et la configuration pour certains pilotes classiques.

**tex-bibtexextra** Une bibliothèque extensive de styles pour BIB<sub>T<sub>E</sub>X</sub>.

**tex-chemistry** Nécessaire pour la chimie.

**tex-context** Le puissant ensemble de macros de Hans Hagen : ConText.

**tex-documentation** Les guides généraux et la documentation.

**tex-etex** Fichiers de support pour l'extension  $\epsilon$ -T<sub>E</sub>X.

**tex-extrabin** Plusieurs programmes utiles mais pas indispensables. On y trouve des programmes et des macros pour le système texinfo, des programmes pour la manipulation de fichiers dvi, etc.

**tex-fontbin** Des programmes de conversion entre différents format de fontes, ainsi que des programmes de test (des programmes pour les fontes virtuelles, manipulation des `.gf` et `.pk`, `mft`, `fontinst`, etc.).

**tex-fontsextra** Toutes sortes de fontes supplémentaires.

**tex-formatsextra** Un ensemble de « formats » T<sub>E</sub>X, c'est-à-dire de gros ensembles de fichiers de macros conçus pour être convertis en un fichier `.fmt`.

**tex-games** Des outils pour mettre en forme plusieurs jeux (notamment les échecs).

**tex-genericextra** Un ensemble de macros et de fontes inclassables.

**tex-htmlxml** Des composants pour la conversion de L<sup>A</sup>T<sub>E</sub>X vers XML/HTML et pour la mise en page de documents XML/SGML.

**tex-langafrican** Support pour quelques langues africaines.

**tex-langarmenian** Nécessaire pour l'arménien.

**tex-langcjk** Extension CJK (macros et fontes) pour le chinois, le japonais et le coréen.

**tex-langcroatian** Nécessaire pour le croate.

**tex-langcyrilic** Macros et fontes pour les textes cyrilliques.

**tex-langczechslovak** Installer ceci si vous voulez les fontes et autres composants pour les textes tchèques et slovaques.

**tex-langdanish** Nécessaire pour le danois.

**tex-langdutch** Nécessaire pour le hollandais.

**tex-langfinnish** Nécessaire pour le finnois.

**tex-langfrench** Nécessaire pour le français.

**tex-langgerman** Nécessaire pour l'allemand.

**tex-langgreek** Nécessaire pour le grec.

**tex-langhungarian** Nécessaire pour le hongrois.

**tex-langindic** Nécessaire pour l'hindou.

**tex-langitalian** Nécessaire pour l'italien.

**tex-langlatin** Nécessaire pour le latin.

**tex-langmanju** Nécessaire pour le manju.

**tex-langmongolian** Nécessaire pour le mongole.

**tex-langnorwegian** Nécessaire pour le norvégien.

**tex-langothier** D'autres fichiers de césure.

**tex-langpolish** Nécessaire pour le polonais.

**tex-langportuguese** Nécessaire pour le portugais.

**tex-langspanish** Nécessaire pour l'espagnol.

**tex-langswedish** Nécessaire pour le suédois.

**tex-langtibetan** Polices et support pour le tibétain.

**tex-langukenglish** Nécessaire pour l'anglais « britannique », différent de l'américain installé par défaut.

**tex-langvietnamese** Nécessaire pour le vietnamien.

**tex-latex** Les composants de cette collection sont exigés par le noyau dur de l'équipe L<sup>A</sup>T<sub>E</sub>X, ou plus simplement très largement recommandés.

**tex-latexextra** Une bibliothèque de composants pour L<sup>A</sup>T<sub>E</sub>X.

**tex-mathextra** Des outils pour les maths.

**tex-metapost** Macros pour dessiner avec MetaPost (et MetaFont).

**tex-music** Des composants pour la mise en forme de partitions musicales.

**tex-omega** Ω, une extension de T<sub>E</sub>X aux caractères sur 16-bits due à John Plaice et Yannis Haralambous.

**tex-pdf<sub>tex</sub>** Les fichiers pour l'extension de T<sub>E</sub>X qui permet de générer du PDF due à Hàn Thé Thanh.

**tex-pictures** Des composants pour tout ce qui est graphique.

**tex-plainextra** Une bibliothèque de macros pour le format plain T<sub>E</sub>X.

**tex-psfonts** Des fontes PostScript supplémentaires.

**tex-psutils** Des outils pour la manipulation des fichiers PostScript.

**tex-publishers** Des fichiers nécessaires pour divers éditeurs.

**tex-tlutils** Outils pour manipuler les fichiers des fontes Type1.

**tex-textbooks** Exemples et autres fichiers accompagnant différents livres sur T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X.

**tex-theses** Fichiers de styles de thèses de différentes universités.

**tex-ttfutils** Utilitaires pour manipuler les fichiers des fontes TrueType.

**tl<sub>doc</sub>** Documentation du T<sub>E</sub>X Live.

**win32-support** Dans cette collection, vous pouvez choisir individuellement des outils pour Windows. Ceux-ci sont des éditeurs orientés T<sub>E</sub>X, des outils pour manipuler des images, etc.

Le répertoire `texmf/tpm/packages` contient la liste détaillée de tous les fichiers dans chaque collection et dans chaque composant (ces fichiers sont utilisés par les programmes d'installation).

**ATTENTION :** Ce CD-ROM est au format ISO 9660 (High Sierra), avec les extensions « Rock Ridge » et « 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.

### 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, exécuter le script `install-cd.sh`, puis sélectionner l’option <R> (ne pas copier les fichiers, l’exécution se fera à partir du CD-ROM). Cela prend très peu d’espace disque et vous donne accès immédiatement 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. Vous pourriez également copier tout le contenu du CD-ROM sur votre disque dur et travailler de cette manière.
2. Vous pouvez installer tout ou partie du système sur votre disque dur. C’est la plupart du temps le meilleur choix si vous disposez d’un espace disque suffisant (200 Mo au minimum, mais on recommande 500 Mo).
3. Vous pouvez installer un jeu de composants choisis pour travailler avec un système **T<sub>E</sub>X** déjà présent ou avec un **T<sub>E</sub>X Live** que vous auriez déjà installé.

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

N.B. : les droits de « root » sont souvent requis pour monter le CD-ROM. Selon le type d’Unix utilisé, il vous faudra probablement connaître le mot de passe de « root » ou tout au moins utiliser la commande « sudo » pour effectuer l’installation. En particulier le droit d’écriture sur le(s) répertoire(s) d’installation sont évidemment indispensables.

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

L’organisation de Web2c vous permet d’exécuter les programmes en indiquant simplement dans votre PATH le sous-répertoire approprié dans le répertoire bin du CD-ROM. Les fichiers nécessaires seront alors tous trouvés sans autres modifications. La liste des systèmes proposés et des sous-répertoires correspondants est fournie ci-dessous. **Seules les architectures x86 Linux, Mac OSX et Windows sont disponibles sur le CD 1. Vous devrez demander le CD 2 si vous avez besoin d’autres systèmes.**

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

Vous pouvez vous inquiéter de ce qui va arriver lors de création de fontes ou de changement de configuration du fait que vous ne pouvez pas modifier les fichiers sur le CD-ROM. En fait, 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-var` sur le CD-ROM (chemin qui n'existe pas !), donc vous *devez* redéfinir 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 /mnt/cdrom
```

Ensuite ils doivent changer le répertoire courant pour `/mnt/cdrom`, puis exécuter :

```
>> sh install.sh
```

et sélectionner l'option `<R>` (*ne pas installer les fichiers, préparer pour une exécution depuis le CD-ROM*). À la suite de quoi, ils doivent modifier la variable `PATH` pour y inclure le répertoire qui contient les binaires pour l'architecture voulue.

```
PATH=/mnt/cdrom/bin/i386-linux:$PATH
```

```
export PATH
```

```
VARTEXMF=/usr/TeX/texmf-var
```

```
export VARTEXMF
```

Par commodité, ces commandes peuvent aussi être exécutées à partir du fichier `.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. Vous pouvez éditer et modifier les fichiers de configuration locaux qui sont stockés dans le répertoire désigné par `$VARTEXMF`. Les fichiers de formats nécessaires seront générés et stockés ici.

## 3.2 Installer $\TeX$ Live sur le 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 type de votre ordinateur. Il devrait commencer par vous retourner les messages suivants :

```

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

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

Enter command:

```

FIGURE 1 – Écran de contrôle principal

```

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

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

Press key to toggle status of collection:

```

FIGURE 2 – Sélection des collections

```

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 parmi cinq options.

1. Changer le type du système utilisé par votre ordinateur ou que vous avez décidé d'installer.
2. Sélectionner le *schéma d'installation* que vous désirez installer (complet, recommandé, basique, GUTenberg, etc.).



```

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

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

Press key to toggle status of collection:

```

FIGURE 3 – Choix par l'utilisateur dans une collection

3. Changer des collections par rapport au schéma d'installation sélectionné (elles sont organisées en deux catégories : *collections standards* et *collections linguistiques*).
4. Définir les répertoires du disque dur où vous souhaitez installer les fichiers.
5. Changer quelques paramètres de fonctionnement lors de l'utilisation.

Pour choisir les options, tapez une lettre ou un nombre puis `return`. Dans l'exemple de la figure 1, le système ELF sous Linux a été détecté. Les collections par défaut seront installées dans le répertoire d'installation (par défaut) `/usr/Tex`. Notons que l'espace disque requis pour l'installation sélectionnée est affiché. Si vous choisissez l'installation recommandée, vous avez besoin d'environ 400 mégaoctets d'espace disque ; cependant l'installation *de base* n'en a besoin que de 100 à 200 et vous pouvez ensuite l'enrichir avec les composants qui vous sont nécessaires.

À partir du répertoire d'installation choisi, le script d'installation écrira les fichiers binaires dans un sous-répertoire du répertoire `bin` et créera l'arborescence de nom `texmf`. Une arborescence supplémentaire `texmf-var` contiendra une copie des fichiers de configuration (sauf pour le fichier `texmf.cnf` principal), fichiers qui pourront être modifiés par le programme `texconfig`. Cette arborescence contiendra aussi les fichiers de format générés pour  $\text{\TeX}$ , METAFONT, etc.

Quand vous choisissez `<C>` pour les *collections standard*, la liste des collections disponibles s'affiche (figure 2). Chaque collection – fichiers de macros  $\text{\TeX}$ , famille de fontes METAFONT, etc. – consiste en un ensemble de composants. Vous pouvez choisir l'inclusion, ou non, de la collection en pressant au clavier la lettre correspondante. Attention le programme différencie les lettres minuscules des lettres capitales pour les touches de sélection.

Quand vous choisissez `<L>` pour les *collections linguistiques*, la liste des collections pour toutes sortes de langues s'affiche (figure 3). Chaque collection consiste en un ensemble de composants qui gèrent les césures ou des fichiers de fontes spécifiques.

La touche de sélection `<O>` pour *options* vous permet de choisir un autre endroit pour la création des nouveaux fichiers de fontes (si vous voulez que l'arborescence `texmf` principale soit en mode en lecture uniquement) et également de choisir si vous voulez créer des liens symboliques pour les binaires, pour les pages `man` et `GNU info` vers les répertoires « standards ». Pour ce faire, vous aurez besoin des permissions de l'administrateur système « root ».

Lorsque votre configuration est terminée, retournez à l'écran principal et sélectionnez le démarrage de l'installation. Le programme d'installation va lire chaque collection et système demandé, consulter la liste des fichiers sur le CD-ROM et construire la liste globale des fichiers à transférer. Ceux-ci seront alors 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 ...). Lorsque cette séquence sera terminée, il vous restera à indiquer à votre système d'exploitation le chemin pour trouver le sous-répertoire de bin dans votre installation 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 remonter les binaires d'un niveau, i.e. de /usr/local/bin/alphaev5-osf4.0d à /usr/local/bin. Si vous le faites, vous devrez éditer le fichier texmf/web2c/texmf.cnf (voir annexe 11) et changer la ligne

```
TEXMFMAIN = $SELFAUTOPARENT
```

en

```
TEXMFMAIN = $SELFAUTODIR
```

Si vous transférez l'installation entière dans un autre répertoire vous devez alors éditer TEXMFMAIN pour indiquer le nouveau répertoire et donner à la variable TEXMFCNF de votre environnement la valeur \$TEXMFMAIN/texmf/web2c.

### 3.3 Installation de composants T<sub>E</sub>X Live individuellement 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 par la suite 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 neuf options ; les quatre premières permettent de choisir le composant individuel à installer, la collection complète (i.e. tex-mathextra), le nom du répertoire du CD-ROM, 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 quatre options supplémentaires : avec les deux premières, vous pouvez exclure la documentation et les fichiers sources de l'installation. La troisième empêche l'exécution de mktexlsr prévue par défaut pour reconstruire les bases de données et la dernière ne fait rien que créer la liste des fichiers qui seraient 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 sa 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/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
```

Voici d'autres exemples d'utilisation :

- Installer le composant natbib :  
`>> sh install-pkg.sh --package=natbib`
- Installer le composant alg sans ses fichiers sources et sans documentation :  
`>> sh install-pkg.sh --package=alg --nosrc --nodoc`
- Installer tous les composants de la collection supplémentaire Plain T<sub>E</sub>X.  
`>> sh install-pkg.sh --collection=teX-plainextra`
- Placer tous les fichiers nécessaires à PSTricks dans un fichier tar du répertoire /tmp :  
`>> 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 besoins propres en utilisant le programme texconfig. Il peut être appelé n'importe quand pour modifier votre configuration et fonctionne en mode ligne de commande ou plein écran (ce qui nécessite alors 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 à vous guider.

## 4 Installation et utilisation sous Windows

Cette section concerne uniquement les systèmes Windows 9x, ME, NT, 2K or XP.

Votre système Windows doit aussi supporter l'extension Microsoft Joliet pour pouvoir lire le CD-ROM. Ouvrez l'explorateur de fichiers pour lire le CD-ROM et voir si les noms de fichiers longs apparaissent, avec des lettres minuscules et majuscules. Si ce n'est le cas, vous ne pourrez utiliser T<sub>E</sub>X directement depuis le CD-ROM.

Ce système T<sub>E</sub>X pour Win32 contient Windvi, un nouveau visualiseur similaire dans son utilisation au x<sub>dv</sub>i d'Unix. Sa documentation se trouve dans <texmf/doc/html/windvi/windvi.html>.

## 4.1 Le programme TeXLive.exe

Si votre ordinateur est configuré avec la fonction d'autodémarrage des CD-ROM, vous allez voir apparaître à l'écran une boîte de dialogue munie d'un menu vous permettant de choisir plusieurs fonctions :

- Installer T<sub>E</sub>X sur votre disque dur,
- Installer des éditeurs de texte orientés T<sub>E</sub>X sur votre disque dur,
- Installer des programmes complémentaires sur votre disque dur (Ghostscript, NetPBM, etc.)
- Effectuer une maintenance de votre installation,
- Enlever le système T<sub>E</sub>X,
- Utiliser T<sub>E</sub>X depuis le CD-ROM,
- Nettoyer les fichiers temporaires créés lors de l'utilisation depuis le CD-ROM,
- Mettre à jour certaines DLL de votre système,
- Accéder à certaines pages de documentation : documentation du **T<sub>E</sub>X Live**, pages Web de TUG, pages Web de la distribution fpT<sub>E</sub>X,
- Lancer le programme TeXdocTK qui vous permettra de trouver de la documentation spécifique sur le CD-ROM.

Si votre CD-ROM ne démarre pas automatiquement lors de l'insertion dans la machine, vous pouvez déclencher l'exécution du programme TeXLive.exe directement depuis l'explorateur de fichiers en double cliquant sur le fichier bin/win32/TeXLive.exe sur le CD-ROM.

## 4.2 Utilisation à partir du CD-ROM

Vous pouvez exécuter les programmes T<sub>E</sub>X directement à partir du CD-ROM et avoir accès à toutes les fontes et macros nécessaires mais en contrepartie, l'exécution sera plus lente que si vous aviez installé **T<sub>E</sub>X Live** 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 configuration 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.

Si vous souhaitez utiliser T<sub>E</sub>X de cette manière, vous devrez suivre les étapes ci-dessous :

1. Depuis le menu, choisir Explore CD-Rom, ensuite Select a text editor, une boîte de dialogue s'ouvrira pour vous permettre de sélectionner un programme exécutable (.exe ou .com).

Le programme que vous sélectionnerez *doit être* un éditeur de textes capable de lancer la compilation T<sub>E</sub>X, le visualiseur ainsi que tout autre outil nécessaire. Si vous n'avez pas encore muni votre système d'un tel outil, vous pouvez en installer un à partir du CD-ROM. Vous trouverez plus de détails en section 4.3.

*Le programme TeXLive.exe n'a aucun moyen de savoir si l'éditeur sélectionné est effectivement un éditeur de textes, donc soyez attentif à votre choix.* Voici une liste d'éditeurs couramment utilisés conjointement avec T<sub>E</sub>X :

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



FIGURE 4 – Fenêtre d’accueil du CD-ROM **T<sub>E</sub>X Live**

Le programme sélectionné sera mémorisé pour de futures utilisations.

2. Depuis le menu, choisir **Explore CD-Rom**, puis **Run TeX off CD-Rom**. L’environnement sera modifié, un répertoire temporaire sera créé et quelques fichiers de configuration y seront copiés. Ensuite, l’éditeur sélectionné sera lancé et vous pourrez entrer du texte, laisser T<sub>E</sub>X le composer, puis le visualiser ou l’imprimer.

Si Ghostscript n’est pas détecté sur votre machine, vous recevrez un message d’avertissement spécifiant que la visualisation risque d’être de mauvaise qualité. Vous pouvez installer Ghostscript en sélectionnant **Install**, puis **Support** dans le menu. Reportez-vous à la section 4.3 pour plus de détails.

3. Vous pouvez choisir un autre éditeur de textes à tout moment.
4. Si vous choisissez l’option **Cleanup CD-Rom setup**, tout ce dont T<sub>E</sub>X a eu besoin sera effacé de votre machine, y compris le choix de votre éditeur de textes. En revanche, les composants supplémentaires que vous aurez pu installer (WinShell, NTEmacs, etc.) ne seront pas enlevés.

L’éditeur est lancé à l’intérieur d’un environnement modifié. Un répertoire conforme à la norme TDS est créé dans une zone temporaire de votre disque dur. Ce répertoire est nécessaire pour enregistrer les fichiers qui pourraient être créés au vol comme des fichiers de police pk ou des fichiers de format. Les

fichiers de configuration existant sur le CD-ROM y sont également copiés, de manière à ce que vous puissiez les éditer si nécessaire. Ensuite de quoi le fichier `ls-R` est construit pour ce répertoire. Les variables `PATH` et `TEXMFCNF` sont positionnées localement, puis l'éditeur est lancé dans l'environnement de ces variables. À l'intérieur de votre éditeur<sup>1</sup>, vous aurez accès à l'environnement complet **T<sub>E</sub>X Live**, tous les fichiers étant référencés sur le CD-ROM.

**[Pour utilisateurs avertis :]** Alternativement et pour des besoins plus spécifiques, vous pouvez utiliser le fichier de commandes `setupw32\mkloctex.bat` fourni par [Staszek Wawrykiewicz](#). Depuis le menu « Démarrer », sélectionnez « Exécuter », puis parcourez l'arborescence du CD-ROM et sélectionnez `mkloctex.bat` dans le répertoire `setupw32`. Avant de le lancer, vous devez ajouter deux paramètres, séparés par un espace : la lettre désignant votre lecteur CD et celle désignant le disque dur où vous souhaitez installer le répertoire T<sub>E</sub>X. La ligne devrait ressembler à : `d:\setupw32\mkloctex.bat d c`. Lorsque l'installation est terminée, lisez attentivement les informations à l'écran, puis vous devrez redémarrer Windows si vous avez Windows 9X/ME.

### 4.3 Installer un éditeur ou un composant support

Vous pouvez utiliser le programme `TeXSetup.exe` pour installer un composant qui n'est pas dépendant de T<sub>E</sub>X, et ce même si l'installation de **T<sub>E</sub>X Live** sur le disque dur n'a pas encore eu lieu. Ce composant peut être un éditeur de textes comme WinShell ou NTEmacs, ou bien un composant support comme NetPBM (manipulation d'images) ou Ghostscript.

Certains de ces composants ne sont pas libres de droits, ou n'ont pas une licence compatible avec celle du CD-ROM. Ces composants sont malgré tout disponibles à travers l'Internet. Vous devez démarrer une connexion Internet pour pouvoir les télécharger. Le choix de l'option `Enable Internet access` déclenche la recherche d'une connexion Internet active. Si aucune n'est trouvée, il vous sera proposé d'en établir une. *Si votre ordinateur n'est pas connecté à l'Internet, il se peut que la recherche dure assez longtemps avant d'échouer : 30s ou plus.* Donc, ne choisissez cette option que si vous êtes certain d'avoir une machine connectée à l'Internet.

Seuls quelques composants support et éditeurs sont disponibles sur le CD-ROM, mais ce sont les plus importants et ils sont suffisants dans la majorité des cas : NTEmacs et WinShell pour les éditeurs, Ghostscript et NetPBM pour le support. NetPBM est entre autres nécessaire pour l'utilisation de T<sub>E</sub>X4ht.

Les composants téléchargeables sont parfois très volumineux : Perl fait 10Mo, XEmacs plus de 50Mo, donc faites attention, car ils leur transfert peut prendre beaucoup de temps. Le programme `TeXSetup.exe` ne fournit pas encore d'estimation du temps de téléchargement en fonction du débit de la connexion.

L'installation de ces composants par `TeXSetup.exe` se fait en mode automatique, sans intervention humaine. En revanche, certains d'entre eux ont leur propre programme d'installation – c'est le cas de Ghostscript et de WinEdt – et dans ce cas, puisque c'est celui-ci qui sera utilisé, votre interaction avec la machine sera requise.

Les composants qui n'ont pas d'installateur spécifique seront décompactés puis configurés pour votre système. Le programme vous demandera de spécifier l'endroit où vous voulez les mettre sur votre disque dur. Le répertoire à sélectionner doit être la racine de toute votre installation. Par exemple, si vous voulez installer NTEmacs et NetPBM, les fichiers archive contiennent déjà la partie `NTEmacs\` et `NetPBM\` du chemin, donc le répertoire que vous devez sélectionner est quelque chose comme `c:\Local` ou `c:\Program Files`.

---

<sup>1</sup>En fait, vous pouvez utiliser un autre programme qu'un éditeur de textes, et par exemple sélectionner votre interpréteur de commandes. Vous ouvrirez ainsi une console avec le bon environnement pour exécuter T<sub>E</sub>X depuis le CD-ROM.

## 4.4 Installation sur le disque dur

L'installation sur le disque dur s'effectue soit en choisissant l'option TeXLive depuis le menu Install dans le programme TeXLive.exe, soit en exécutant directement le programme bin\win32\TeXSetup.exe sur le CD-ROM. Le programme d'installation est un assistant Windows qui se déroule en plusieurs pages.

**Welcome Page** Vous pouvez sélectionner une installation rapide, dans ce cas le programme se déroulera sans aucune intervention manuelle, avec les paramètres par défaut (figure 5, sur la gauche). Cependant, si vous choisissez un composant qui possède son propre programme d'installation, celui-ci pourra nécessiter votre intervention. Si vous avez les droits de l'administrateur ou d'un utilisateur avec pouvoirs (sous une version de Windows où cela existe), vous pouvez choisir une installation pour tous les utilisateurs ou seulement pour vous-même en cochant la case prévue à cet effet.

L'assistant d'installation



Répertoires source pour les fichiers du **T<sub>E</sub>X Live**

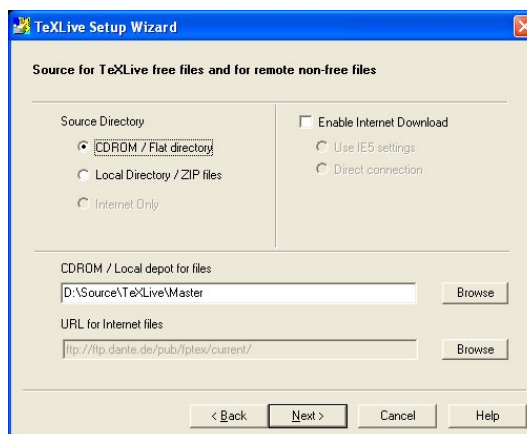


FIGURE 5 – La page de définition des répertoires source du **T<sub>E</sub>X Live**

**Source Page** Cette page est un peu complexe et vous permet de sélectionner deux répertoires source pour votre système **T<sub>E</sub>X Live**(figure 5, sur la droite). Vous aurez besoin d'un répertoire source *local* et éventuellement d'un répertoire source *distant*.

Pourquoi le besoin de ces deux répertoires ? Les fichiers stricto-sensu du **T<sub>E</sub>X Live** sont sur le CD-ROM, mais un certain nombre d'autres composants utiles sous Win32 ne s'y trouvent pas, soit par manque de place, soit parce que leur licence n'est pas compatible, même en étant gratuits. Pour avoir accès à ces composants supplémentaires, il faut autoriser le téléchargement depuis Internet.

Pour vous rassurer : les paramètres par défaut vous permettent une installation complète depuis le CD-ROM uniquement. Simplement, vous n'aurez pas accès par exemple à WinEdt, mais vous pourrez l'installer par la suite.

Vous pouvez prendre les fichiers source depuis :

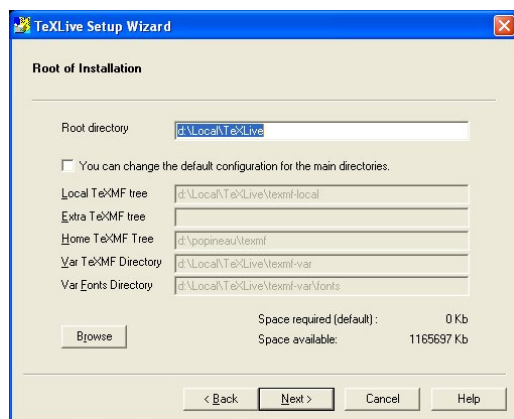
- le CD-ROM ou toute arborescence équivalente accessible par un répertoire standard (ce qui veut dire que le CD-ROM peut être chargé dans une machine distante et son contenu rendu disponible par partage réseau),



- un ensemble de fichiers .zip (cas de la distribution fp $\text{\TeX}$ ),
  - Internet, auquel cas le programme prend soin de télécharger les fichiers .zip pour vous.
- Cette dernière option n'est valide que si vous avez autorisé le téléchargement de fichiers depuis Internet en cochant la case appropriée. Il faut également configurer ce téléchargement en choisissant une connexion utilisant Internet Explorer 5 (et `wininet.dll`), ou une connexion directe (`ftp`, `http`).

**Root Page** Cette page vous permet de définir où vous voulez installer les fichiers (Figure 6, sur la gauche). Seul le répertoire racine est réellement important et, par défaut, les autres sont déterminés en fonction de celui-ci. Vous pouvez vouloir spécifier un répertoire pour  $\$TEXMFEXTRA$  si besoin est, ou donner une valeur différente pour  $\$HOMETEXMF$  qui est déterminée par défaut en fonction de ce que Windows considère être votre répertoire « HOME ».

#### Répertoires pour l'installation



#### Sélection du schéma d'installation

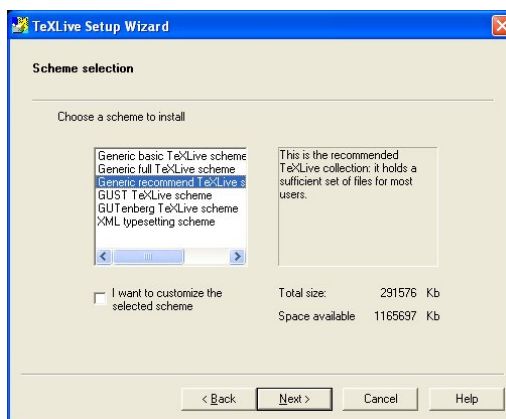


FIGURE 6 –  $\text{\TeX}$  Live-Setup : Répertoires/Schémas

**Get TPM Page** Cette page ne demande aucune intervention manuelle. Les fichiers .tpm décrivant les collections et les composants sont retrouvés, le cas échéant sur Internet, décompressés, puis analysés.

**Schemes Page** Ici, vous sélectionnerez le schéma global que vous souhaitez installer. Un schéma correspond à un ensemble de fichiers destinés à un certain type d'application. Il y a trois schémas génériques pour une installation standard de base, recommandée ou complète. Les autres sont dédiés aux LUGs (ce que GUTenberg ou GUST recommande à ses adhérents) ou à des applications (XML et  $\text{\TeX}$ ).

Lorsqu'un schéma est sélectionné, il est encore possible de modifier et de raffiner la sélection en cochant la case appropriée. Si vous le faites, vous passerez à la page de sélection des composants (Packages Page), sinon vous sauterez à la page de dernière vérification avant installation (Review Page).

**Packages Page** Les collections et les composants vous sont présentés sous forme arborescente (figure 7, sur la gauche). Les liens dans l'arbre sont des liens de dépendance. Les collections *dépendent* d'un certain nombre de composants et éventuellement d'autres collections ; il en est de même

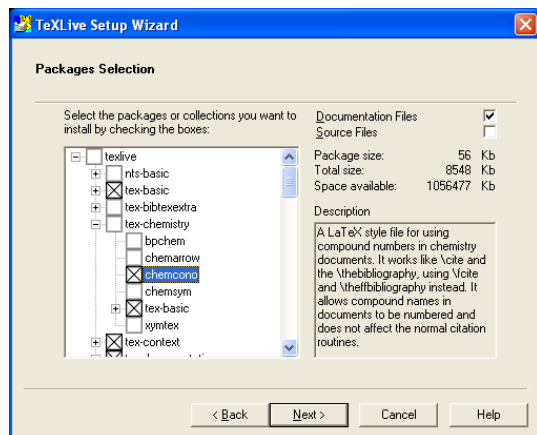


pour chaque composant. Vous pouvez choisir de désélectionner individuellement un composant ou une collection, mais votre demande ne sera prise en compte que si l'objet en question n'est pas requis par un autre qui est également sélectionné. Par exemple, vous ne pouvez pas désélectionner `tex-basic` sans désélectionner toutes les collections qui la requièrent.

Il existe une collection `win32-support` spécifique à Win32 (sur la droite). Elle contient un certain nombre de packages « bonus » qui peuvent être installés individuellement : l'interpréteur Postscript Ghostscript, des éditeurs orientés  $\text{\TeX}$ , des outils tels que Perl,  $\text{\LaTeX}$ 2HTML, etc. *Aucun de ces composants n'est choisi par défaut.* Parmi les composants disponibles dans cette collection support, certains marqués par une icône doivent être récupérés sur Internet par le programme d'installation, et ils ne seront donc disponibles que si cette option a été activée auparavant. *Cette collection ne peut pas être sélectionnée intégralement en une fois, vous devez sélectionner ses composants un par un.* Ceci pour éviter des téléchargements longs et non désirés sur Internet.

Vous avez également sur cette page les informations sur la place disque nécessaire pour chaque objet et pour l'ensemble de ceux sélectionnés, ainsi que la place libre sur la partition choisie pour l'installation. Enfin, vous pouvez choisir d'installer ou non les fichiers de documentation et les fichiers sources pour chaque composant.

Support Win32



Revue des paramètres

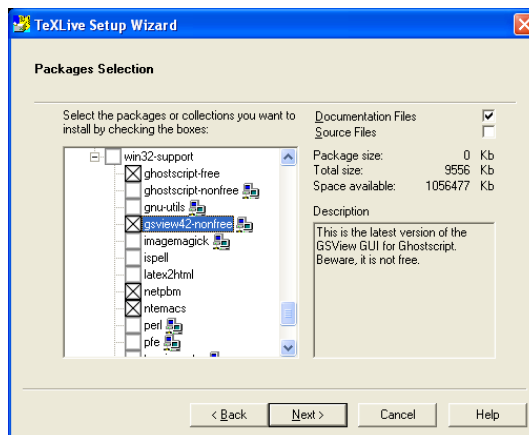


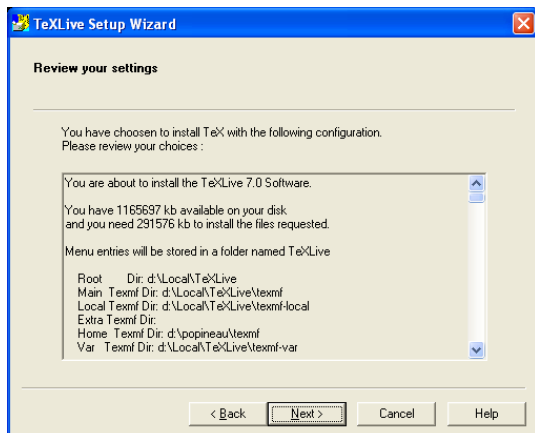
FIGURE 7 – Sélection des composants/Composants spécifiques Win32

**Review Page** Vous avez ici un résumé des informations concernant vos choix (figure 8, sur la gauche). Il est encore temps de revenir en arrière pour les modifier.

**Files Copy Page** Les fichiers correspondants à votre sélection sont copiés sur votre disque dur. Le cas échéant, tous les fichiers non encore disponibles sur votre machine sont téléchargés depuis le site distant choisi, puis tous les composants sont décompressés et installés.

**Configuration Page** Certains composants nécessitent une phase spéciale pour les rendre utilisables (figure 9, sur la gauche) et  $\text{\TeX}$  Live a besoin de certaines opérations pour finaliser l'installation (génération des formats, des bases de données de fichiers, etc.). Toutes ces opérations sont effectuées ici et peuvent prendre un temps considérable.

Page de confirmation des paramètres



Page de copie des fichiers

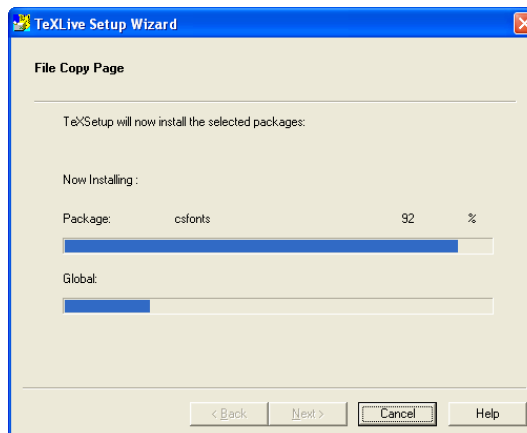
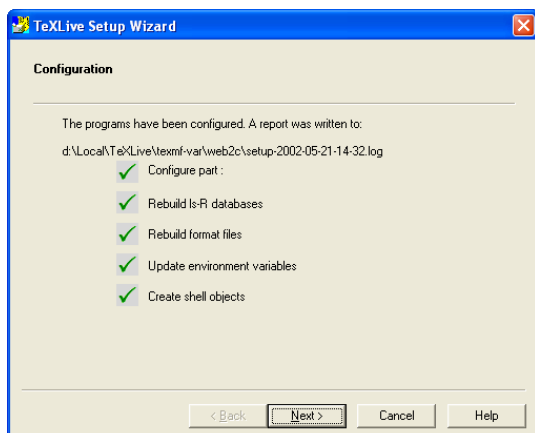


FIGURE 8 – Confirmation des paramètres d’installation / Copie des fichiers

**Final Page** L’installation terminée (figure 9), vous pouvez choisir d’afficher la documentation dédiée à Windows (format HTML) ou le fichier log de votre installation. Si c’est nécessaire (cas de Win9x/WinME), il vous sera demandé de redémarrer votre machine.

Page de configuration



Page finale

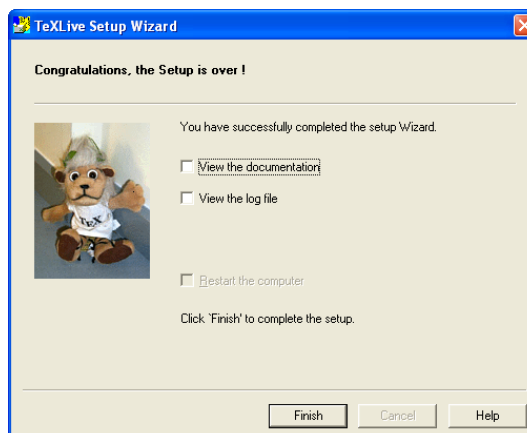


FIGURE 9 – Configuration / Fin de l’installation

Attention : les utilisateurs souhaitant installer T<sub>E</sub>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, et T<sub>E</sub>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 clusters plus faibles.

## 5 Maintenance et autres aspects de l'installation de T<sub>E</sub>X Live sous Windows

### 5.1 Différences entre la version Win32 de Web2c et la version standard

La version Win32 de Web2c possède quelques spécificités qui méritent d'être notées.

**Kpathsea** les tables de hachage<sup>2</sup> que Kpathsea construit (à partir des fichiers `ls-R` entre autres) sont très volumineuses pour le T<sub>E</sub>X Live. Dans le but de diminuer le temps d'initialisation des applications qui utilisent Kpathsea, ces tables ont été mises en mémoire partagée. De cette manière, quand vous exécutez des chaînes de programmes, telles que `tex` qui appelle `mpost` qui appelle `tex`, le temps d'initialisation de chaque programme est diminué (sauf pour le premier). Ce comportement est invisible pour l'utilisateur, sauf si celui-ci positionne le niveau de trace de Kpathsea à la valeur `-1` : dans ce cas, tout accès au bloc de mémoire partagée est tracé, ce qui n'est pas souhaitable (ce bloc est accédé très souvent !). Cette situation évoluera probablement dans le futur.

**kpsecheck** cette commande fournit des fonctionnalités qui ne rentraient pas bien dans `kpsewhich`. En particulier, elle permet de lister tous les fichiers qui apparaissent plusieurs fois à travers les multiples arborescences `texmf/`. Ceci à l'air pratique, sauf qu'en réalité, la sortie est polluée par des occurrences multiples de fichiers qui ne sont pas utiles à T<sub>E</sub>X à proprement parler (comme des douzaines de fichiers `README`)<sup>3</sup>. Pour cette raison, vous pouvez combiner l'option `-multiple-occurrences` avec deux autres options pour inclure ou exclure de la recherche, tout fichier conforme à un certain motif (plusieurs motifs d'inclusion ou d'exclusion peuvent être spécifiés).

La commande `kpsecheck` peut aussi indiquer le statut d'utilisation de la mémoire partagée : en utilisation ou non-utilisée. Cette information peut être très utile, car si le statut rapporté est « en utilisation », cela signifie qu'un ou plusieurs processus tournent et utilisent le bloc de mémoire partagée. Dans ce cas, une réinitialisation des tables de hachage basée sur les fichiers `ls-R`, comme la commande `mktexlsr` l'effectue, sera automatiquement repoussée jusqu'à ce que tous les processus utilisant la version courante en mémoire partagée soient terminés. Il est prévu d'enlever cette limitation dans une version future, mais la version actuelle de Kpathsea ne permet pas de faire facilement cette réinitialisation.

Enfin, la même commande `kpsecheck` peut indiquer l'endroit où Kpathsea pense pouvoir trouver la DLL de Ghostscript. En effet, sous Win32, il est souvent plus simple de travailler directement avec la DLL de Ghostscript, et de la trouver en utilisant la clé appropriée dans la base de registre, que d'utiliser `gswin32c.exe` et de modifier le `PATH` qui a une longueur limitée.

**Web2c** les moteurs T<sub>E</sub>X ont quelques options supplémentaires par rapport à la version standard, et une option dont le comportement diffère de la version standard :

- l'option `-fmt` se comporte différemment. Dans les versions précédentes et avec la version standard de Web2c, cette option possède deux significations différentes selon que vous êtes en

---

<sup>2</sup>Une table de hachage est une structure de donnée qui permet de rechercher un élément par une clé dans un ensemble. Elle doit donc mémoriser des paires de type (*clé, valeur*) et permettre un accès rapide à une valeur étant donné sa clé.

<sup>3</sup>Tous ces fichiers sont susceptibles de produire des collisions lors de l'accès à la hash-table ; heureusement, comme T<sub>E</sub>X ne les utilise pas, ils ne sont pas pris en compte !

mode « ini » ou « vir ». Sous Win32, elle a dorénavant une seule signification : charger le fichier de format donné en argument. La définition du nom du format construit en mode « ini » s'obtient dorénavant par l'option -job-name.

- l'option -job-name permet de définir le nom du fichier résultant de la compilation par  $\text{\TeX}$ . En mode normal, cette option changera la racine de tous les fichiers créés (.aux, path.dvi, etc.), alors qu'en mode « ini », c'est le nom du fichier de format construit qui sera modifié.
- halt-on-error arrête la compilation à la première erreur.
- job-time positionne la date des fichiers créés à la date de référence fournie par celle du fichier donné en argument.
- output-directory permet de créer tous les fichiers de sortie dans le répertoire spécifié.
- time-statistics affiche des statistiques à propos du temps d'exécution. Il est à noter que Win9x n'étant pas un système multi-tâche préemptif, il ne possède pas d'horloge interne suffisamment précise pour mesurer ces temps d'exécution et donc la valeur affichée n'est qu'une approximation. Sous NT/2K/XP, le résultat est assez précis et comprend le temps utilisateur et le temps système alloués à la compilation. Pour les utilisateurs d'Unix : la commande time n'est pas disponible sous Windows.

## 5.2 Ajouter des composants à votre système

Vous avez une option dans le menu TeXLive (Démarrer -> Programmes -> TeXLive) qui vous permet de lancer à nouveau le programme TeXSetup.exe en mode maintenance. Les étapes à franchir sont pratiquement les mêmes que celles de l'installation.

Le seul point différent concerne la sélection des composants. En mode maintenance, une comparaison est effectuée avec les composants installés sur votre disque dur. Vous verrez alors s'afficher en vert les composants dont vous ne disposez pas encore, en rouge ceux dont vous disposez mais pour lesquels une version plus récente est disponible et en noir ceux qui sont déjà installés.

Ainsi vous pouvez choisir d'ajouter ou de mettre à jour des composants soit depuis votre CD-ROM, soit depuis Internet auquel cas il est probable que vous trouverez également des composants plus récents que ceux que vous avez déjà installés.

Il vous appartient de faire votre choix parmi l'ensemble des composants. Pour le reste, les étapes sont identiques à celles de l'installation initiale.

Si vous choisissez d'ajouter des fichiers qui ne proviennent pas de la distribution **TeX Live** (ou  $\text{\fpTeX}$ ), il vous est fortement recommandé de les mettre dans le répertoire  $\$TEXMFLOCAL$ . De cette manière, vous serez certain qu'il n'y aura pas de problème lors d'une mise à jour de **TeX Live**.

L'arborescence pointée par  $\$TEXMFLOCAL$  est initialement vide. Si vous souhaitez y ajouter les fichiers de style pour supporter le logiciel de calcul formel Maple, vous devrez mettre ces fichiers dans le répertoire `c:\Program Files\TeXLive\texmf-local\tex\latex\maple\` et les fichiers de documentation dans

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

Ensuite, **n'oubliez pas de reconstruire les bases de données de fichiers**, soit en utilisant le menu prévu à cet effet (Démarrer -> Programmes -> TeXLive -> Maintenance), soit en lançant manuellement la commande `mktexlsr`. Si vous l'oubliez, vos fichiers ne seront pas trouvés.

### 5.3 Enlever T<sub>E</sub>X Live de votre disque dur

La procédure de désinstallation est disponible soit depuis le programme `TeXLive.exe`, soit depuis le menu `TeXLive`, soit depuis le panneau de contrôle (menu Démarrer -> Panneau de Contrôle, option Ajouter/Enlever des programmes). Cette procédure nettoie votre disque dur de la plupart des fichiers qui y ont été mis lors de l'installation initiale. Cependant, T<sub>E</sub>X est un système qui génère maints fichiers et, pour l'instant, il n'est pas prévu de mécanisme pour en garder la trace. D'autre part, les composants supplémentaires spécifiques à Windows possèdent leur propre procédure de désinstallation qu'il faudra lancer séparément. Enfin, les éventuels fichiers que vous aurez mis dans le répertoire `$TEXMFLOCAL` ne seront pas inquiétés. Donc, même si la majeure partie des fichiers est nettoyée automatiquement, il vous restera quelques opérations manuelles à effectuer.

### 5.4 Utiliser `TeXSetup.exe` depuis la ligne de commande

Le programme `TeXSetup.exe` possède un certain nombre d'autres options intéressantes. Vous pouvez en obtenir la liste en lançant :

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

Voici leur description :

- automatic-reboot réinitialise la machine sans attendre confirmation par l'utilisateur une fois l'installation terminée ;
- dry-run ne rien installer, juste écrire dans le fichier de trace ce qui aurait été fait sans cette option ;
- quick fonctionne en mode automatique jusqu'à la phase de réinitialisation, avec les options par défaut si rien de plus n'est spécifié ;
- net-method (=ie5/direct) permet de télécharger les composants avec une licence restrictive ou non-disponibles sur le CD depuis l'Internet (soit en utilisant une connexion directe, soit en utilisant les DLL de Internet Explorer 5) : vous devez disposer d'une connexion Internet et certains composants sont très volumineux ;
- remote-source-directory <url> permet de spécifier l'url de base pour les composants distants ;
- local-source-directory <dir> par défaut, `TeXSetup.exe` essaiera de deviner la racine du répertoire source où se trouvent les fichiers à copier, mais si vous voulez utiliser une mise-à-jour de ce programme, vous ne pourrez pas copier la nouvelle version sur votre CD-ROM, donc vous devrez utiliser cette option pour spécifier le répertoire racine de votre CD-ROM ;
- installation-directory <dir> permet de spécifier la racine du répertoire destination, tous les fichiers seront copiés en-dessous de ce répertoire. La valeur par défaut est `c:\Program Files\TeXLive` ;
- texmfmain-directory <dir>
- texmflocal-directory <dir>
- texmfextra-directory <dir>
- texmfhome-directory <dir>
- vartexmf-directory <dir>

- vartextfonts-directory <dir> ce sont les répertoires utilisés pour configurer l'endroit où se trouvent vos fichiers. Ils correspondent directement aux variables trouvées en tête du fichier `texmf.cnf`.
- with-source-files(=yes/no) copier les fichiers source pour les composants  $\text{\TeX}$  (fichiers source pour les fichiers de macros), la valeur par défaut est « no » ;
- with-documentation-files(=yes/no) copier les fichiers de documentation pour les composants  $\text{\TeX}$ . La valeur par défaut est « yes ». Attention : il s'agit de la documentation spécifique des composants, la documentation générale sera toujours installée ;
- program-folder <folder> le nom du groupe dans lequel vous trouverez les menus ;
- add-package <pkg> cette option est utilisée pour ajouter ou mettre à jour un composant après une première installation ;
- scheme <pkg> sélectionne le schéma indiqué par défaut, au lieu du schéma `texlive-recommended` ;
- maintenance la même chose que --add-package sans spécifier quel composant ajouter ;
- uninstall cette option permet d'effacer tout ce qui est relatif au  **$\text{\TeX}$  Live** et qui provient du CD-ROM de votre disque dur, mais il se peut que certains fichiers générés au vol (fontes, formats) ou que vous auriez ajoutés (styles) restent après la désinstallation. Il en est de même des composants qui possèdent leur propre installateur (support, éditeurs)<sup>4</sup> ;
- help Cette option affiche la liste des options du programme `TeXSetup.exe`.

## 5.5 Installation en réseau

Kpathsea est compatible avec les noms UNC, donc vous pouvez les utiliser pour récupérer votre répertoire `texmf` principal depuis le réseau. Mais encore mieux que cela, tous les fichiers, y compris ceux de configuration et excepté les binaires dans `bin/win32` sont compatibles et partageables avec  $\text{\TeX}$  ou le  **$\text{\TeX}$  Live** Unix. Cela signifie que vous pouvez utiliser Samba, soit pour monter votre distribution Unix sur votre client Windows, ou votre client Unix depuis un NT serveur. Plusieurs stratégies sont possibles :

- tout mettre sur le serveur. Juste être certain de mettre dans `bin/` les binaires pour toutes les couples systèmes d'exploitation / architectures nécessaires. Par exemple, ajouter `bin/win32` et `bin/i386-linux`. Ensuite configurer vos variables principales ; Vous pouvez utiliser des noms UNC pour pointer sur les bons répertoires sous Win32.
- Installer une copie locale pour les binaires et les fichiers de format. Dans ce cas, assigner `$TEXMFMAIN` au répertoire principal `texmf` qui sera pris sur le réseau. Faire pointer `$VARTEXMF` vers un répertoire local qui contiendra les fichiers de configuration locaux et les fichiers générés au vol.

## 5.6 Personnalisation de l'installation

### 5.6.1 WinShell

Winshell devrait automatiquement trouver le visualiseur `Windvi`. Si ce n'était pas le cas, lancez Winshell depuis le menu Démarrer ou le raccourci sur le bureau, et allez dans **Options->Program Calls** :

- dans l'onglet `DVIWin`, si le programme mentionné est **yap**, remplacez le par **windvi**,

---

<sup>4</sup>Cette option est encore un peu expérimentale à ce jour (28 mai 2002).

- dans l’onglet Ghostview, vérifiez que le chemin donné pour `gsview32.exe` est correct :  
`c:\ghostgum\gsview\gsview32.exe` (pour les versions à partir de 3.6)  
ou  
`c:\gstools\gsview\gsview32.exe` (pour les versions antérieures).
- Tapez OK.

Remarquez que l’installateur de WinShell associe les fichiers avec extension `.tex` à WinShell. Ceci est correct, à moins que vous ne souhaitiez utiliser un autre éditeur (tel que WinEdt ou Emacs).

Il manque un correcteur orthographique à WinShell. Si vous avez installé la collection `tex-extrabin` et le composant `ispell`, vous avez alors accès au correcteur **Ispell** disponible sur la plupart des systèmes Unix. Le programme est sur votre PATH, donc il sera trouvé, même si vous l’appellez à partir d’une console (fenêtre DOS). Si vous avez installé la documentation, référez vous à `c:\Program Files\TeXLive\texmf\doc\html\manpages\ispell.html` pour plus d’informations (fichier également disponible sur le CD-ROM). Si vous utilisez habituellement un vérificateur d’orthographe, vous pouvez souhaiter ajouter une icône pour Ispell à WinShell (voir la sous-section 5.9.4).

Pour un excellent (mais commercial, quoique peu onéreux) correcteur orthographique, se reporter à <http://www.microspell.com>.

D’autres informations sur WinShell en section 5.9.

## 5.6.2 Dvips

Le fichier de configuration de `dvips` se trouve par défaut en `c:\Program Files\TeXLive\texmf-var\dvips\config\config.ps`. Vous pouvez l’ouvrir avec n’importe quel éditeur de texte (comme WinShell) pour modifier certains paramètres :

**fontes** vous pouvez changer la résolution et le mode de l’imprimante à laquelle sont destinés vos fichiers dans le cas où `dvips` aurait besoin de générer des fontes bitmap. Par défaut, les fontes Type1 CM interpolées sont utilisées, et donc `mktexpk` ne devrait pas être appelé trop souvent.

**imprimante** vous pouvez spécifier où vous voulez imprimer par défaut. Si l’option ‘o’ n’est pas suivie d’un nom d’imprimante, un fichier est créé avec l’extension `.ps`. Sinon, vous pouvez spécifier un nom d’imprimante tel que :

```
o lpt1:
% o | lpr -S server -P myprinter
% o \\server\myprinter
```

**papier** vous pouvez changer le format de papier retenu par défaut (A4), par exemple en format US letter, le premier format mentionné. Allez vers les lignes débutant par `@`. Déplacez les lignes adéquates, ainsi votre fichier commencera par les lignes qui suivent.

```
@ letterSize 8.5in 11in

@ letter 8.5in 11in
@+ %%BeginPaperSize: Letter
@+ letter
@+ %%EndPaperSize
```

### 5.6.3 Pdftex

Le fichier de configuration pour Pdftex se trouve en `c:\Program Files\TeXLive\texmf-var\pdftex\config\pdftex.cfg`. Vous pouvez l'ouvrir dans un éditeur de texte pour modifier son contenu, par exemple remplacer le format de papier A4 par le format US letter ou tout autre. Ouvrez le fichier de configuration dans un éditeur de texte et remplacez les valeurs des paramètres `page_width` et `page_height` pour spécifier les valeurs voulues, par exemple :

```
page\_width 8.5 true in
page\_height 11 true in
```

Sauvegarder le fichier et sortir de l'éditeur.

### 5.6.4 Gsview

À partir des versions compatibles avec Ghostscript 6.50, Ghostview n'est plus un logiciel libre, mais un shareware. Il n'est donc plus fourni sur le CD-ROM.

Si vous voulez changer le format du papier, ouvrez GSView à partir du menu Démarrer, Programmes. Depuis le sous-menu Media, sélectionnez Letter. Le sous menu Display Settings vous permet également d'améliorer la netteté du rendu en positionnant les deux valeurs **Text Alpha** et **Graphics Alpha** à 4 bits.

Pour ce qui est de l'impression, se référer à la sous-section 5.8.

Les fichiers `.ps` et `.eps` seront ouverts automatiquement par GSView.

### 5.6.5 Windvi

Lors de l'installation, les fichiers `.dvi` sont associés avec Windvi.

Vous pouvez lancer Windvi à partir du menu Démarrer, Programmes, TeXLive ou en double cliquant un fichier `.dvi` dans l'explorateur de fichiers..

Pour sélectionner un format de papier US Letter, allez dans le menu View, Options de Windvi et sélectionnez « US (8.5"x11") » dans la liste déroulante Paper Type. Cliquez OK et fermez Windvi.

La première fois que vous ouvrirez un fichier `.dvi`, vous pouvez trouver le facteur de zoom trop important, réduisez le en tapant sur la touche « moins » du clavier numérique jusqu'à ce qu'il soit à votre goût.

Vous pouvez modifier le format de papier dans les options de Windvi (menu View, Options) ainsi qu'un certain nombre d'autres paramètres comme par exemple la capacité à exécuter des commandes externes spécifiées dans des `\special{}`.

Les fichiers :

```
c:\Program Files\TeXLive\texmf\doc\windvi\Examples\wtest.dvi
c:\Program Files\TeXLive\texmf\doc\windvi\Examples\wtest.tex
```

démontrent l'ensemble des possibilités de Windvi.

L'impression avec Windvi nécessite pour l'instant que les paramètres de mode METAFONT et de résolution soient positionnés en accord avec les caractéristiques de votre imprimante.

Vous pouvez les modifier dans le menu View, Options de Windvi ou vous trouverez une liste déroulante de modes (MF mode) ainsi que la possibilité de fournir une résolution (pixels per inches). Vous pouvez également les assigner une fois pour toutes en lançant Windvi en ligne de commande :

```
c:\>windvi -p 360 -mfmode canonbjc foo.dvi
```



Lorsque vous quitterez Windvi, les paramètres seront sauvegardés. les modes disponibles sont définis dans le fichier

```
c:\Program Files\TeXLive\textmf\metafont\misc\modes.mf
```

Les paramètres de Windvi sont sauves dans un fichier du nom de `$HOME/windvi.cnf`. Vous pouvez le localiser de la manière suivante :

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

Si vous avez des problèmes avec Windvi, il est conseillé d’effacer le fichier de configuration, puis de refaire un test dans la configuration par défaut.

## 5.7 Tests

Vous pouvez tester rapidement votre installation basée sur WinShell en ouvrant le fichier

```
c:\Program Files\TeXLive\textmf\tex\latex\base\sample2e.tex
```

Le source  $\text{\LaTeX}$  doit apparaître à l’écran. Compilez-le en cliquant sur l’icône  $\text{\LaTeX}$  dans la barre d’outils, ensuite visualisez-le en cliquant sur l’icône Preview (Windvi).

La première fois que vous visualiserez un document avec Windvi, il va créer les fichiers de fontes bitmaps qui ne sont pas installées. Après avoir visualisé quelques fichiers, vous aurez créé la plupart de ces fichiers et vous ne verrez plus souvent apparaître la fenêtre de création de fontes. Retournez à WinShell, et essayez Dvips puis GSView.

En cas de problèmes, reportez-vous à la sous-section 5.11.

## 5.8 Impression

Il est possible d’imprimer depuis Windvi. Dans ce cas, l’impression utilise le pilote unifié d’impression de Windows, il est donc par définition compatible avec toutes les imprimantes. Cependant, il y a un inconvénient : cette impression génère des fichiers (spool) très importants, quelques versions anciennes de Windows le supportent mal. L’avantage est que vous pouvez tirer parti de l’impression d’images BMP ou WMF par exemple. Il faut également faire bien attention à ce que les paramètres de l’imprimante soient correctement définis (sous-section 5.6.5) sous peine d’avoir un effet d’échelle (imprimer à 600 dpi sur une imprimante qui fait réellement 300 dpi résulte en un seul quart de la page visible).

L’impression est souvent plus rapide en utilisant dvips, puis en imprimant le fichier .ps depuis GSView. Pour imprimer depuis GSView, sélectionner **Print...** dans le menu **File**. Une fenêtre de dialogue pour l’impression apparaît.

Si vous utilisez une imprimante Postscript, *soyez sûr de sélectionner **PostScript Printer*** en choisissant cette option dans **Print Method** en bas à gauche de la boîte de dialogue, faute de quoi l’impression échouera. Vous pouvez ensuite sélectionner une imprimante quelconque parmi celles installées.

Si vous utilisez une imprimante qui ne supporte pas PostScript, sélectionnez **Ghostscript Device** dans **Print Method**. Ensuite cliquez sur le bouton **djet500** et sélectionnez votre imprimante.

Si vous utilisez WinShell et une imprimante PostScript, le plus pratique est de définir un bouton Impression dans la barre d’outils de WinShell qui appellera dvips avec les options envoyant directement le résultat à l’imprimante. Pour des instructions détaillées sur la manière de procéder, reportez-vous à la sous-section 5.9.3.

## 5.9 Pour aller plus loin avec WinShell


### 5.9.1 Installation de correctifs

L'auteur de WinShell ([Ingo de Boer](#), merci à lui) publie des versions beta de la prochaine version qui sont également des correctifs. On peut les récupérer sur <http://www.winshell.de>. Il s'agit en général de fichiers archives aux format .zip qu'il suffit de décompresser dans le répertoire où se trouve WinShell, soit à l'aide d'un utilitaire tel que WinZip, ou bien d'une ligne de commande avec unzip. Si vous avez récupéré un correctif du nom de winshellbugfix.zip et que vous l'avez sauvegardé dans le répertoire de WinShell, alors vous devez exécuter :

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

Répondre « oui » quand il vous est demandé si des fichiers doivent être écrasés.

### 5.9.2 Gestion des projets

Si votre document comporte plusieurs fichiers (une thèse par exemple), essayez d'utiliser la capacité de WinShell à gérer des projets. Depuis le menu **Project**, attribuez un nom à votre projet, donnez le nom du document maître et ajoutez les autres fichiers qui composent votre projet. Ces fichiers apparaissent dans la fenêtre de gauche, vous pouvez cliquer dessus pour les visualiser et passer de l'un à l'autre. L'icône  provoque la compilation du fichier maître.

À noter les icônes de la barre d'outils pour commuter l'affichage du panneau concernant le projet (à gauche) et de la fenêtre de log (en bas). Si vous n'utilisez pas la fonctionnalité de projet, vous pouvez souhaiter faire disparaître la fenêtre de gauche pour travailler sur toute la largeur de l'écran.

### 5.9.3 Imprimer sur une imprimante PostScript depuis WinShell

L'utilisation de l'icône Dvips de la barre d'outils de WinShell écrit la sortie PostScript dans un fichier que vous pouvez ensuite visualiser avec GSView et éventuellement imprimer. Cependant, il est encore plus pratique d'ajouter à WinShell un appel spécifique à dvips qui envoie la sortie directement sur l'imprimante. Les étapes ci-dessous montrent comment le faire pour l'imprimante *vc/w* ; vous devez substituer à *vc/w* le nom de l'imprimante que vous utilisez le plus fréquemment (vous pouvez utiliser un nom UNC pour une imprimante en réseau).

1. Intégrez l'imprimante à WinShell :
  - Ouvrez WinShell, allez dans le menu **Options -> Program Calls -> User Defined**.
  - Cliquez sur **Tool 1** dans la liste sur la droite et remplissez les champs comme suit :

```
Name : Print
exe file : dvips
cmd-line : -D600 %m -o \emph{vc/w}
```

Décochez l'option pour « DVIPS first »
  - Cliquez **OK**.
2. Ajoutez l'outil Imprimer à la barre d'outils :
  - Allez dans le menu **Options -> View -> Customize**.
  - Dans le cadre **Category**, choisissez **User-Programs**.

- Sélectionnez **Print** et glissez-le sur la barre d'outils, lâchez-le juste à droite de l'icône de GSView.
- Vous avez ensuite le choix entre « Image only », « Text only », « Image and Text ». Le plus simple est de sélectionner « Text only » et de cliquer **OK**. Vous devriez ensuite voir **Print** sur la barre d'outils. Si vous préférez, vous pouvez aussi choisir « Image only », ensuite « Edit », puis éditer l'image à votre goût.

Maintenant, vous pouvez imprimer un document sur l'imprimante *vc/w* en cliquant simplement sur l'icône **Print**. Si vous voulez utiliser une imprimante différente, il faut repasser par *dvips* pour imprimer dans un fichier, puis utiliser GSView pour envoyer ce fichier sur l'imprimante de votre choix.

#### 5.9.4 Configurer WinShell pour utiliser Ispell

1. Ajouter Ispell aux outils utilisateur :
  - Ouvrez WinShell, allez dans le menu **Options -> Program Calls -> User Defined**.
  - Cliquez sur **Tool 1** (ou **Tool 2** si vous avez déjà utilisé le premier) dans la liste sur la gauche et remplissez les champs comme suit :
 

```
Name : Ispell
exe file : ispell
cmd-line : -t -d francais %c.tex
```

 Décochez les options pour « LaTeX first » et pour « DVIPS first »
  - Cliquez **OK**.
2. Ajouter Ispell à la barre d'outils :
  - Allez dans le menu **Options -> View -> Customize**.
  - Dans le cadre **Category**, choisissez **User-Programs**.
  - Sélectionnez **Ispell** et glissez-le sur la barre d'outils, lâchez-le juste à droite de l'icône de GSView (ou de la dernière icône que vous avez ajoutée).
  - Vous avez ensuite le choix entre « Image only », « Text only », « Image and Text ». Le plus simple est de sélectionner « Text only » et de cliquer **OK**. Vous devriez ensuite voir **Ispell** sur la barre d'outils. Si vous préférez, vous pouvez aussi choisir « Image only », ensuite « Edit », puis éditer l'image à votre goût.

Maintenant, lorsqu'un document  $\text{\LaTeX}$  est ouvert, vous pouvez cliquer sur l'icône **Ispell** pour réaliser une vérification orthographique. Ispell va ouvrir une nouvelle fenêtre et afficher le premier mot inconnu sur la gauche avec le nom de fichier sur la droite. En dessous, vous verrez le contexte dans lequel ce mot inconnu apparaît. La plupart du temps, plusieurs propositions de remplacement vous sont offertes. Pour remplacer le mot, entrez le numéro de la proposition choisie. D'autres possibilités existent, par exemple vous pouvez presser la barre d'espace pour ignorer cette détection d'erreur. Pour plus d'informations sur Ispell, référez-vous au manuel :

`c:\Program Files\TeXLive\texmf\doc\html\manpages\ispell.html`

Quand vous choisissez de remplacer un mot dans Ispell, le remplacement ne sera effectif dans WinShell qu'une fois le processus Ispell terminé et le fichier sauvegardé (cliquez sur le X dans le coin supérieur droit de la fenêtre), il vous faudra ouvrir le fichier à nouveau dans WinShell.

## 5.10 Trucs et astuces à propos de la plateforme Win32

### 5.10.1 Différentes déclinaisons de Win32

Ce que recouvre la dénomination Win32 n'est pas un système d'exploitation. C'est un ensemble de fonctions très vaste<sup>5</sup> que vous pouvez utiliser pour écrire des programmes pour différentes versions des systèmes d'exploitation de la famille Windows.

Windows se décline en plusieurs versions :

- Win95, Win98 et WinME, qui *ne sont pas de vrais systèmes d'exploitation multitâches et multi-threads*. Ils sont en fait les dernières incarnations de DOS. On peut s'en apercevoir, car en lançant le PC, l'interpréteur de commandes `command.com` est chargé et si vous arrêtez le processus à ce moment, vous pouvez demander la version courante (de DOS) et il vous est répondu quelque chose du style "MS-DOS 7.0", au moins pour les versions anciennes de Win9x ;
- Windows NT est un système d'exploitation écrit de zéro, avec un vrai multitâche préemptif et des fonctionnalités de très haut niveau ;
- Windows 2K est écrit sur une base NT, on l'a doté de toutes les facilités offertes par Win98 ;
- Windows XP, existe en versions Personnelle et Professionnelle. C'est la dernière étape dans la fusion entre les deux lignes de produits (basée sur Win9x et basée sur NT). XP est écrit sur une base NT également.

Win9x est capable de faire tourner des programmes 32 bits et des programmes 16 bits en même temps. Mais le système d'exploitation lui-même n'est pas entièrement écrit en mode 32 bits et ne fournit pas une protection mémoire entre les applications : les applications 16 bits peuvent écraser des parties du système d'exploitation en mémoire ! Des parties du système telles que le GDI (Graphical Device Interface) ne se voient allouer que des ressources de taille très limitée pour gérer les bitmaps, les pincesaux et les polices, et ces ressources sont allouées de manière globale pour tous les programmes qui tournent de manière concurrente. Par exemple, toutes les entêtes de bitmaps utilisés par tous les programmes qui tournent simultanément ne doivent pas requérir plus que 64ko de mémoire. Ceci explique le comportement du moniteur de performance, et le fait que vous pouvez mettre votre système à genoux en utilisant de manière intensive les objets graphiques.

NT, 2K et XP ne souffrent pas de ces limitations, et d'aucune autre limitation de Win9x. Ce sont de vrais environnements multitâches, avec une vraie mémoire protégée. Ils répondent de manière plus fluide que Win9x de par leur meilleure gestion de la mémoire, leur système de gestion de fichiers plus performant, etc.

### 5.10.2 La ligne de commande

Vous allez vous demander : mais pourquoi diable devrais-je me préoccuper d'une ligne de commande alors que j'ai Windows ?

Bonne question. le problème est de nature très générale. Toutes les opérations ne peuvent pas être accomplies très facilement à l'aide de la seule interface graphique. La ligne de commande vous donne la puissance de la programmation – si vous avez un bon interpréteur de commandes.

Mais le problème est plus fondamental :  $\text{\TeX}$  est un outil qui fonctionne *en batch*, de manière non-interactive.  $\text{\TeX}$  a besoin de calculer la meilleure mise en page pour chaque page, de résoudre les références croisées, etc. Ceci ne peut être réalisé que par un traitement global du document. Ce n'est pas encore une tâche qui peut être réalisée interactivement.

---

<sup>5</sup>Environ 12000 fonctions dans les fichiers d'en-tête du SDK Microsoft

Ceci implique que vous devriez utiliser  $\text{\TeX}$  depuis la ligne de commande. En fait la situation n'est pas si catastrophique. Il y a un avantage à écrire des outils en ligne de commande pour des tâches complexes : ils sont bien plus fiables, parce qu'ils n'héritent pas de la complexité inhérente aux interfaces graphiques. Il est ensuite possible de concevoir des outils graphiques qui servent d'interface aux outils en ligne de commande. C'est le cas de  $\text{\TeX}$  : vous interagirez avec lui la plupart du temps au travers d'un éditeur de textes qui possède une interface graphique – voir la section 5.6.1 par exemple.

Cependant, il se peut que vous ayez besoin d'utiliser la ligne de commande dans un certain nombre de situations, par exemple en cas de problèmes, parce que vous avez besoin de trouver une erreur dans votre installation – voir la section 5.11.

**Win9x, WinME** Vous ouvrirez une ligne de commande soit en cherchant l'icône MS-DOS dans le menu « Démarrer -> Programmes », soit en choisissant « Démarrer -> Exécuter », puis en tapant `command.com`.

**NT, 2K, XP** Vous ouvrirez une ligne de commande en cherchant « Invite de commandes » dans le menu « Démarrer -> Accessoires »<sup>6</sup>. Vous pouvez aussi choisir le menu « Démarrer -> Exécuter » et taper `cmd.exe`, qui est le nom du nouvel interpréteur de commandes pour NT<sup>7</sup>.

### 5.10.3 Les séparateurs de chemins

L'API Win32 admet les deux caractères / et \ comme séparateurs pour les noms de fichiers. Mais pas les interpréteurs de commande ! Donc, chaque fois qu'un nom de fichier est utilisé par un programme, vous pouvez utiliser l'un ou l'autre séparateur, mais sur la ligne de commande, vous devez utiliser \ comme unique séparateur. Ce qui explique que vous pouvez taper :

```
c:\>set TEXMFCNF=c:/Program Files/TeXLive/texmf-var/web2c
```

mais pas :

```
c:\>dir "c:/Program Files/TeXLive"
```

Dans le premier cas, seuls des programmes utiliseront le chemin que vous avez fourni, dans le deuxième c'est l'interpréteur de commandes qui va vouloir s'en servir directement.

Tout ceci pour dire : ne soyez pas surpris de voir des noms de fichiers écrits avec des / en guise de séparateurs, à la Unix ;  $\text{fp\TeX}$  est un portage de Web2c, dont l'objectif est d'être compatible avec toutes les plateformes. Pour cette raison, les fichiers de configuration utilisent la convention Unix des séparateurs dans les noms de fichiers.

Il est aussi à noter qu'il existe un autre séparateur qui sert à séparer plusieurs chemins, comme c'est le cas dans la variable d'environnement PATH. Sous Win32, ce séparateur est un ; alors que c'est le caractère : sous Unix. Étant donné que le : sert de séparateur de nom de périphérique dans plusieurs systèmes d'exploitation, ici c'est Unix qui a adopté la convention de Win32 en utilisant le caractère ; pour séparer plusieurs chemins. En fait Unix peut utiliser ; ou :, alors que Win32 ne peut pas utiliser : qui serait ambigu.

---

<sup>6</sup>Ces endroits peuvent changer d'une version à l'autre de Windows.

<sup>7</sup>Ce qui explique pourquoi il est erroné d'appeler cette console une boîte *DOS* sous NT !

### 5.10.4 Les systèmes de gestion de fichiers

Une des plus mauvaises caractéristiques de Win9x vis-à-vis de  $\text{\TeX}$  est probablement ce qu'on appelle le système de fichiers FAT.  $\text{\TeX}$  utilise une myriade de petits fichiers dont la taille varie entre 1ko et 5ko. Le système FAT est ancien et date d'une époque bien antérieure à l'apparition des disques de plusieurs Go qui sont monnaie courante aujourd'hui. Tout ceci pour dire qu'il n'est pas possible de gérer efficacement les 30000 fichiers de  **$\text{\TeX}$  Live** sur un disque dur formaté en FAT. Les fichiers se verront allouer chacun 32ko au minimum, donc l'installation de  **$\text{\TeX}$  Live** utilisera beaucoup plus de place que nécessaire.

Le seul moyen d'éviter ce problème consiste à passer en FAT32 ou NTFS. Ces systèmes sont plus récents et n'ont pas l'inconvénient de FAT. La taille des clusters par défaut y est de 4ko, leur accès est plus performant. NTFS est protégé, redondant et on peut même ajuster la taille des clusters jusqu'à 512 octets à la création.

### 5.10.5 Comment ajouter un répertoire à votre PATH

Il existe dans votre système des variables qui agissent un peu comme des variables globales à tous vos programmes. On appelle cet ensemble de variables *l'environnement*. Chaque programme hérite à son démarrage d'une copie de l'environnement. Il peut modifier les valeurs des variables, ajouter ou enlever des variables, mais les modifications ne sont effectives que pour sa propre copie et ne sont pas propagées aux autres programmes, sauf à ceux qu'il lance lui-même.

Votre variable PATH est une variable spéciale de l'environnement utilisée pour chercher les programmes lorsque vous en demandez l'exécution. Il y a une procédure différente pour modifier cette variable selon que vous êtes sous Win9x, ME ou NT/2K/XP.

**Windows 95/98** Éditez votre fichier autoexec.bat. Dans ce fichier, vous trouverez une ligne commençant par PATH= et suivie par une liste de répertoires séparés par des ;. Ajoutez le répertoire contenant les programmes exécutables au bout de cette ligne. Après avoir fait cela, cette ligne ressemblera à la suivante :

```
PATH=c:\windows;c:\windows\system;c:\"Program Files"\TeXLive\bin\win32
```

Les modifications ne prendront effet qu'après redémarrage de la machine.

**Windows ME** Vous devez utiliser le programme c:\windows\system\msconfig.exe pour pouvoir modifier une des variables d'environnement. Dans ce programme, choisissez l'onglet Environnement, ensuite ajoutez ou modifiez la variable de votre choix. Vous devrez redémarrer la machine pour que les modifications prennent effet.

**Windows NT/2K/XP** Ouvrez le Panneau de Contrôle, accessible depuis le menu Démarrer, Paramètres. Ouvrez l'icône Système, puis choisissez l'onglet Environnement ou cherchez un bouton « Variables d'environnement » parmi les différentes boîtes de dialogue offertes. Il est à noter que les variables d'environnement système, communes à tous les utilisateurs sont également affichées. Vous ne pouvez les modifier que si vous avez les droits d'administrateur. Si c'est le cas, vous pouvez modifier la variable PATH pour tous les utilisateurs du système.

Si il y a déjà un PATH défini pour votre compte utilisateur, cliquez dessus. Dans le champ Variable apparaît PATH et dans le champ Valeur, la liste courante de répertoires séparés par des ;. Ajoutez les répertoires où se trouvent vos exécutables (i.e. c:\Program Files\TeXLive\bin\win32). Si la variable PATH n'est pas encore définie, il suffit de taper son nom dans le champ Valeur et la valeur initiale que vous souhaitez lui donner dans le champ Valeur. Important : cliquez sur le

bouton Appliquer avant de cliquer sur Ok, de cette manière, les modifications seront propagées immédiatement à votre session.

Le meilleur moyen de savoir si une variable a été correctement définie consiste à ouvrir une console et à taper

```
set VARIABLE
```

la valeur correspondante devrait vous être retournée.

### 5.10.6 Les moteurs $\text{\TeX}$

En lisant la documentation de Web2c, vous verrez que les différents programmes dérivés de  $\text{\TeX}$  utilisent le même moteur de base. Par exemple, `tex.exe` et `latex.exe` sont des copies exactes du même programme, mais chacun utilise un fichier de format différent, en se basant sur le nom par lequel il a été invoqué.

Sous Unix, ce mode de fonctionnement est réalisé en faisant appel aux *liens symboliques*. On peut économiser un peu d'espace disque, car plusieurs moteurs de base sont utilisés avec différents fichiers de format.

L'API Win32 ne connaît pas les liens symboliques. Dans le but d'économiser tout de même un peu de place disque, tous les moteurs  $\text{\TeX}$  de base ont été mis dans des DLL (*Dynamic Linked Library*). Ceci se traduit par l'aspect suivant pour les fichiers :

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

et le fichier `latex.exe` n'est ni plus ni moins qu'une copie conforme du fichier `tex.exe`, utilisant le même moteur `tex.dll`. La même astuce a été utilisée pour la famille de programmes `mktex*.exe` qui utilisent tous la bibliothèque `mktex.dll`.

Il existe même un outil générique appelé `lnexe.exe` qui permet de simuler les liens durs de Unix sous Win32, mais uniquement pour les fichiers `.exe`.

## 5.11 En cas de problème

### 5.11.1 Que faire si `latex` ne trouve pas vos fichiers ?

- `kpsewhich` est l'outil de choix pour trouver la source de n'importe quel problème. Malheureusement, `kpsewhich` écrit la trace de ses recherches sur le flux `stderr`, et la console de Windows ne sait pas rediriger ce flux vers un fichier<sup>8</sup>. Pour des besoins de diagnostic, vous pouvez positionner temporairement une variable d'environnement (dans une console) :

```
SET KPATHSEA_DEBUG_OUTPUT=err.log
```

Vous pouvez également définir un niveau de trace :

```
SET KPATHSEA_DEBUG=-1
```

---

<sup>8</sup>En fait, NT/2K/XP savent le faire, grâce à leur nouvel interpréteur de commandes, mais le truc spécifique à Kpathsea-Win32 marche sur toutes les consoles.

La trace de l'exécution des commandes suivantes sera conservée dans le fichier `err.log`. Si vous voulez rediriger le flux `stderr` sur le flux `stdout`, ce qui n'est normalement possible sous aucune version de Windows, il vous suffit de faire :

```
SET KPATHSEA_DEBUG_OUTPUT=con:
```

De cette manière, vous pourrez rediriger à la fois `stdout` et `stderr` dans le même fichier.

- En supposant que l'installation a été faite dans `c:/Program Files/TeXLive`, vérifiez les valeurs suivantes :

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

- Si vous avez des variables d'environnement liées à TeX positionnées dans votre environnement, enlevez-les, car elles masquent les valeurs existant dans le fichier `texmf.cnf`.
- Vérifiez les valeurs de :
 

```
kpsewhich cmr10.tfm c:/Program Files/TeXLive/texmf/fonts/tfm/public/cm/cmr10.tfm
kpsewhich latex.fmt c:/Program Files/TeXLive/texmf/web2c/latex.fmt
```
- À ce point, si tout est correct, `tex.exe` et tous les autres programmes devraient fonctionner. Si ce n'était pas le cas, vous devrez poursuivre vos investigations avec l'option `-debug=n` de `kpsewhich` et contrôler toutes les valeurs. Essayez d'identifier et de signaler le problème.

### 5.11.2 Que faire si votre installation ne fonctionne toujours pas selon vos attentes ?

Il y a plusieurs questions immédiates à se poser :

1. Est-ce que `tex.exe` est bien dans mon PATH ?
2. Est-ce que la variable d'environnement `TEXMFCNF` est bien positionnée à `c:/Program Files/TeXLive/texmf-var/web2c` (valeur par défaut) ?
3. Est-ce qu'il y a des erreurs dans le fichier de log généré lors de l'installation ? Les erreurs sont repérées par la séquence « Error ».
4. On peut aussi consulter les pages <http://www.tug.org/tex-live.html> pour y trouver d'éventuels correctifs de bugs.
5. La distribution pour Windows disponible sur le CD-ROM n'étant autre que la distribution `fpTeX`, vous pouvez également consulter les pages Web <http://www.fptex.org>, ou vous abonner à la liste de diffusion correspondante en allant sur <http://www.tug.org/mailman/listinfo/fptex>.

Le logiciel **TeX Live** est complexe, car composé de plus de 250 programmes et environ 40000 fichiers d'origines très différentes. Il est pratiquement impossible de prédire toutes les causes possibles de problèmes. Néanmoins, nous ferons notre possible pour vous aider dans tous les cas.

## 5.12 Compiler les fichiers sources

La totalité des fichiers source, y compris pour Windows, est disponible dans l'archive `source/source.tar.bz2` sur le CD-ROM. Pour recompiler l'ensemble de la distribution pour Windows, il vous faut :



- Windows 2K ou XP
  - Microsoft Visual Studio .Net
  - un ensemble d’outils *à la* Unix (*sed*, *grep*, *gawk*, etc.) ainsi que Perl, Flex et Bison,
  - modifier les chemins dans le fichier `win32/make/common.mak` suivant votre installation,
  - modifier les chemins dans le fichier Perl `win32/perl/build.pl`,
  - lancer la compilation depuis le répertoire `win32/` par la commande :
- ```
c:\texlive\source\win32>perl ./perl/build.pl --install --log=install.log
```

Il reste beaucoup de travail à faire pour rendre la compilation plus simple.

## 6 Installation sur un nouveau système Unix

Si vous avez un système Unix ne correspondant à aucun des exécutables fournis, vous aurez besoin de compiler  $\TeX$  et tous ses programmes satellites à partir de zéro. Cela n’est pas aussi difficile qu’il y paraît. Tout ce dont vous aurez besoin est contenu dans le répertoire `source`.

Vous devez tout d’abord transférer l’arborescence contenant tous les fichiers supports à partir du CD-ROM  **$\TeX$  Live** (faire une installation de base sans choisir le type de système d’exploitation et donc les fichiers exécutables).

### 6.1 Prérequis

Il vous faudra 100 Mo d’espace sur votre disque pour compiler la totalité de  $\TeX$  et de ses programmes. Vous aurez aussi besoin d’un compilateur C ANSI, de l’utilitaire `make`, d’un générateur d’analyseur lexical et d’un générateur d’analyseur syntaxique. Les utilitaires GNU (`gcc`, `GNUmake`, `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 sous Unix pour vous sortir des problèmes. La commande `uname` doit renvoyer une valeur sensée.

### 6.2 Configuration

Premièrement, décompactez sur votre disque le fichier `tar` (archive) compressé trouvé dans le répertoire `source` et placez-vous dans le répertoire où vous avez transféré l’ensemble. Décidez du répertoire racine où sera placée l’installation, c.-à-d. `/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. L’ensemble des répertoires qui seront utilisés sont les suivants (où `$TEXDIR` désigne le répertoire de base) :

|                                      |                                                 |
|--------------------------------------|-------------------------------------------------|
| <code>\$TEXDIR/man</code>            | pages du manuel Unix                            |
| <code>\$TEXDIR/share/texmf</code>    | répertoire de base pour les fontes, macros, etc |
| <code>\$TEXDIR/info</code>           | manuels GNU sous format info                    |
| <code>\$TEXDIR/bin/\$PLATFORM</code> | exécutables.                                    |

Vous pouvez omettre la partie « `share/` » pour le répertoire `texmf`, car `$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 `$PLATFORM` relatif au choix du système (c.-à-d. copier directement les exécutables dans `$TEXDIR/bin`), exécutez `configure` avec l'option `--disable-multiplatform`.

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

### 6.3 Exécuter make

Assurez-vous que la variable `noclobber` n'est pas définie et entrez :

```
>> make world
```

et allez prendre un café...

Il peut être utile de garder la trace dans un fichier log en entrant :

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

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

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

```
>> ls | wc
```

Le résultat doit être 213, ce qui correspond à une installation complète.

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

```
>> make all
```

```
>> su
```

```
>> make install strip
```

### 6.4 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 configurer 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 simplement entrer :

```
>> texconfig help
```

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

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

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

## 7 Guide d'utilisation du système Web2c

Web2c contient un ensemble de programmes relatifs à  $\text{\TeX}$ , c.-à-d.  $\text{\TeX}$  lui-même, METAFONT, MetaPost,  $\text{\BibTeX}$ , etc. La première implémentation a été réalisée par Tomas Rokicki qui, en 1987, a développé un premier système  $\text{\TeX}$ -to-C en adaptant les fichiers sous Unix qui étaient 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 dizaines de contributeurs, et en 1997 il passa le relais à Olaf Weber. La dernière version en date est Web2c Version 7.3, parue en mars 1999, qui forme la base du présent  **$\text{\TeX}$  Live** CD-ROM. Notre version diffère légèrement et s'identifie comme étant la version 7.3.7.

Le système Web2c 7.3 fonctionne sur Unix, Windows 3.1, 9x/ME/NT/2K/XP, DOS, et de nombreux autres systèmes d'exploitation. Il utilise les sources originales de D.E. Knuth pour  $\text{\TeX}$  et d'autres programmes de base écrits en web qui sont tous traduits en langage C. De plus, le système offre un large éventail de macros et de fonctions développées dans le but d'améliorer le logiciel  $\text{\TeX}$  original. Les composantes du noyau de  $\text{\TeX}$  sont :

**bibtex** Gère les bibliographies.  
**dmp** troff vers MPX (dessins MetaPost).  
**dvicopy** Copie le fichier DVI en supprimant les fontes virtuelles.  
**dvitomp** Convertit le fichier DVI en MPX (dessins MetaPost).  
**dvitype** Convertit le fichier DVI en un texte lisible.  
**gftodvi** Visualisation de fontes génériques GF.  
**gftopk** Convertit les fontes génériques GF en fontes bitmap PK.  
**gftype** Convertit le fichier GF en un texte lisible.  
**makempx** Typographie des étiquettes MetaPost.  
**mf** Création 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** Création de motifs de césure.  
**pktofb** Convertit les fontes bitmap PK en fontes génériques GF.  
**pktype** Convertit les fontes PK en un texte lisible.  
**pltotf** Convertit les fichiers PL (lisibles) en TFM.  
**pooltype** Affiche les fichiers web pool.  
**tangle** web vers Pascal.  
**tex** Composition de textes.  
**tftopl** Convertit les fichiers TFM en PL (lisibles).  
**vftovp** Convertit les fontes virtuelles VF en VPL (lisibles).  
**vptovf** Convertit les fontes VPL en fontes virtuelles VF.

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

La syntaxe et les fonctions précises de ces programmes sont décrites dans la documentation des composants individuels et dans le manuel Web2c lui-même. Toutefois, connaître un certain nombre de principes gouvernant l'ensemble de la famille de programmes peut aider à exploiter de façon optimale votre installation Web2c. Presque 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 seulement le numéro de version.

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 paramètres pour optimiser la recherche dans l'arborescence  $\text{\TeX}$ . Web2c peut exécuter une recherche dans plusieurs arborescences 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.

## 7.1 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 par « deux-points » ou « point-virgule », d'éléments, appelés *éléments de chemin*, qui 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 renvoie la première occurrence (plus largement : toutes les occurrences).

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 40) contient  $p$ , c.-à-d. 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, si elle ne s'applique pas à cet élément de chemin ou si elle 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 est censé 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 est censé exister » entre en jeu avec les fichiers « .vf » et les fichiers d'entrée lus par la commande  $\text{\TeX}$  \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 occurrence est trouvée, la recherche s'arrête et le résultat est obtenu.

Bien que l'élément de chemin le plus simple et le plus fréquent soit un nom de répertoire, Kpathsea supporte d'autres types d'éléments 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 de base. Ce qui est décrit dans les sections suivantes.

Notons que si le nom de fichier cherché est absolu ou explicitement relatif, c'est-à-dire commençant par « / », « ./ » ou « ../ », Kpathsea ne vérifie que l'existence de ce fichier.

### 7.1.1 Les différentes sources

Un chemin de recherche peut provenir de plusieurs sources. Voici l'ordre dans lequel 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 obtenue à la compilation.

Vous pouvez voir chacune de ces valeurs pour un chemin de recherche donné en utilisant l'option de débogage (voir page 45).

### 7.1.2 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 et les définitions provenant de fichiers précédents écraseront celles des fichiers suivants. Par exemple, avec un chemin tel que `./: $TEXMF`, les définitions du fichier `./texmf.cnf` écrasent celles de `$TEXMF/texmf.cnf`.

En sus de la lecture du descriptif du format du fichier `texmf.cnf` ci-dessous, on doit aussi se référer à l'annexe 11, à la page 51, qui liste le fichier `texmf.cnf` de la distribution sur le CD-ROM.

- Les commentaires sont signalés par un « % » et se terminent 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 optionnels.
- Le nom de la « *variable* » peut contenir n'importe quel caractère autre que les espaces, « = », ou « . », mais on recommande d'utiliser « 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 par exemple à différentes variantes de T<sub>E</sub>X d’avoir des chemins de recherche différents.
- « *value* » peut contenir n’importe quel caractère excepté « % » 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 et permet 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 peuvent être référencées avant d’être définies.

Voici un fichier de configuration illustrant les points précédents

```

TEXMF          = {$TEXMFLOCAL;!!$TEXMFMAIN}
TEXINPUTS.latex = .;$TEXMF/tex/{latex;generic;}//
TEXINPUTS.fontinst = .;$TEXMF/tex//;$TEXMF/fonts/afm//
% e-TeX related files
TEXINPUTS.elatex = .;$TEXMF/{etex;tex}/{latex;generic;}//
TEXINPUTS.etex   = .;$TEXMF/{etex;tex}/{eplain;plain;generic;}//

```

### 7.1.3 Expansion d’un chemin de recherche

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

### 7.1.4 Expansion par défaut

Si le chemin de recherche le plus prioritaire (voir section 7.1.1) contient un « : » supplémentaire (c.-à-d. en début ou fin de ligne ou double), Kpathsea insère à cet endroit le chemin suivant dont la priorité définie est immédiatement inférieure. Si ce chemin inséré possède un « : » supplémentaire, le même processus se répète pour le chemin prioritaire suivant. Par exemple, étant donné une variable d’environnement définie ainsi

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

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 est inutile d’insérer la valeur par défaut en 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 de ligne et enfin un double « : ».

### 7.1.5 Expansion spécifiée par les accolades

Option utile, l'expansion par le biais des accolades signifie, par exemple, que `v{a,b}w` va permettre la recherche dans `vaw:vw`. Les définitions emboîtées sont autorisées. Ceci peut être utilisé pour établir des hiérarchies  $\text{T}_{\text{E}}\text{X}$  multiples en attribuant une liste entre accolades à `$TEXMF`. Par exemple, dans `texmf.cnf`, on trouve (ligne 52) la définition suivante :

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

Avec ceci, on peut écrire quelque chose comme

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

ce qui signifie que, après avoir cherché dans le répertoire courant, les arborescences complètes `$HOMETEXMF` suivie de `$TEXMFLOCAL/tex` (sur le disque) et ensuite les arborescences `!!$VARTEXMF` et `!!$TEXMFMAIN/tex` (définies dans le fichier de référence `ls-R` *seulement*) seront inspectées. C'est un moyen pratique permettant d'utiliser en parallèle deux distributions  $\text{T}_{\text{E}}\text{X}$ , une « 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.

### 7.1.6 Expansion des sous-répertoires

Deux slashes ou plus consécutifs dans une partie d'un chemin suivant un répertoire *d* sont remplacés 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. À chaque niveau, l'ordre dans lequel les répertoires sont inspectés est *non-déterminé*.

Dans le cas où l'on spécifie une partie de nom de fichier après le « `//` », seuls sont inclus les sous-répertoires auxquels 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.

### 7.1.7 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, 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 plusieurs lignes).
- !! Cherche *seulement* dans la base de données pour localiser le fichier et *ne cherche pas* sur le disque.

## 7.2 Les bases de données

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 quand plusieurs centaines de répertoires de polices de caractères doivent être parcourus). En conséquence, Kpathsea peut utiliser un fichier appelé `ls-R` — en fait une base de données construite au préalable — qui fait correspondre les fichiers à leur répertoire, ce qui permet d'éviter une recherche exhaustive sur le disque.

Un deuxième fichier appelé `aliases` (qui est également une base de données) permet de donner des noms différents aux fichiers listés dans `ls-R`. Ceci peut aider à adapter ses fichiers source aux conventions « 8.3 » pour les noms de fichiers dans les systèmes similaires à DOS.

### 7.2.1 Le fichier base de données

Comme nous l'avons expliqué ci-dessus, le nom du principal fichier-base de données doit être `ls-R`. Dans votre installation, vous pouvez en mettre un à la racine de chaque arborescence  $\TeX$  que vous désirez voir inspecter (`$TEXMF` par défaut) ; la plupart des sites ont seulement une arborescence  $\TeX$ . Kpathsea cherche les fichiers `ls-R` dans le chemin spécifié dans la variable `TEXMFDBS`.

La meilleure façon de créer et mettre à jour le fichier « `ls-R` » est d'exécuter le script `mktexlsr` inclus dans la distribution. Il est appelé par les divers scripts « `mktex` ». . . En principe, ce script exécute uniquement la commande

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

en supposant que la commande `ls` de votre système produise le bon format de sortie (le `ls` 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 `ls-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 le chercher sur le disque. Par contre, si un élément du chemin commence par « `!!` », *seule* la base de données sera inspectée pour cet élément, jamais le disque.

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

Le programme `kpsewhich` effectue une recherche dans une arborescence indépendamment de toute application. On peut le considérer comme une sorte de `find` pour localiser des fichiers dans les arborescences  $\TeX$  (ceci est largement utilisé dans les scripts « `mktex` ». . . de la distribution).

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



Les options spécifiées dans « *option* » peuvent 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).

Les 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 pour la recherche à « *name* ». Par défaut, le format est estimé en fonction du nom de fichier. Pour les formats qui n'ont pas de suffixe clair 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. Cette table regroupe les noms reconnus à ce jour, accompagnés d'une description et des variables d'environnement associées<sup>9</sup> et les extensions possibles.

TAB. 1: Kpathsea file types

| <i>Name</i>                   | <i>Description</i>                                                                                   | <i>Variables</i>          | <i>Suffixes</i>                                               |
|-------------------------------|------------------------------------------------------------------------------------------------------|---------------------------|---------------------------------------------------------------|
| <code>afm</code>              | Adobe font metrics                                                                                   | AFMFonts                  | <code>.afm</code>                                             |
| <code>base</code>             | Metafont memory dump                                                                                 | MFbases, TEXMFINI         | <code>.base</code>                                            |
| <code>bib</code>              | BIB <sub>TEX</sub> bibliography source                                                               | BIBINPUTS, TEXBIB         | <code>.bib</code>                                             |
|                               | bitmap fonts                                                                                         | GLYPHFonts, TEXFonts      |                                                               |
| <code>bst</code>              | BIB <sub>TEX</sub> style files                                                                       | BSTINPUTS                 | <code>.bst</code>                                             |
| <code>cnf</code>              | Runtime configuration files                                                                          | TEXMFCNF                  | <code>.cnf</code>                                             |
| <code>dvips config</code>     | <code>dvips</code> configuration files, e.g.,<br><code>config.ps</code> and <code>psfonts.map</code> | TEXCONFIG                 | <code>.map</code>                                             |
| <code>fmt</code>              | <sub>TEX</sub> memory dump                                                                           | TEXFORMATS, TEXMFINI      | <code>.fmt</code> , <code>.efmt</code> ,<br><code>.efm</code> |
| <code>gf</code>               | generic font bitmap                                                                                  | GFFonts                   | <code>.gf</code>                                              |
| <code>graphic/figure</code>   | Encapsulated PostScript figures                                                                      | TEXPICTS, TEXINPUTS       | <code>.eps</code> , <code>.epsi</code>                        |
| <code>ist</code>              | makeindex style files                                                                                | TEXINDEXSTYLE, INDEXSTYLE | <code>.ist</code>                                             |
| <code>ls-R</code>             | Filename databases                                                                                   | TEXMFDBS                  |                                                               |
| <code>map</code>              | Fontmaps                                                                                             | TEXFONTMAPS               | <code>.map</code>                                             |
| <code>mem</code>              | MetaPost memory dump                                                                                 | MPMEMS, TEXMFINI          | <code>.mem</code>                                             |
| <code>mf</code>               | Metafont source                                                                                      | MFINPUTS                  | <code>.mf</code>                                              |
| <code>mfpool</code>           | Metafont program strings                                                                             | MFPOOL, TEXMFINI          | <code>.pool</code>                                            |
| <code>mft</code>              | MFT style file                                                                                       | MFTINPUTS                 | <code>.mft</code>                                             |
|                               | miscellaneous fonts                                                                                  | MISCFonts                 |                                                               |
| <code>mp</code>               | MetaPost source                                                                                      | MPINPUTS                  | <code>.mp</code>                                              |
| <code>mppool</code>           | MetaPost program strings                                                                             | MPPOOL, TEXMFINI          | <code>.pool</code>                                            |
| <code>MetaPost support</code> | MetaPost support files, used by<br>DMP                                                               | MPSUPPORT                 |                                                               |
| <code>ocp</code>              | $\Omega$ compiled process files                                                                      | OCPINPUTS                 | <code>.ocp</code>                                             |

<sup>9</sup>Vous pouvez trouver des définitions pour ces variables d'environnement dans le fichier `texmf.cnf` (page 51)

Kpathsea file types *continued*

| <i>Name</i>              | <i>Description</i>                                  | <i>Variables</i>                                                                               | <i>Suffixes</i>              |
|--------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------|
| ofm                      | Ω font metrics                                      | OFMFonts, TEXFonts                                                                             | .ofm, .tfm                   |
| opl                      | Ω property lists                                    | OPLFonts, TEXFonts                                                                             | .opl                         |
| otp                      | Ω translation process files                         | OTPinputs                                                                                      | .otp                         |
| ovf                      | Ω virtual fonts                                     | OVFFonts, TEXFonts                                                                             | .ovf                         |
| ovp                      | Ω virtual property lists                            | OVPFFonts, TEXFonts                                                                            | .ovp                         |
| pk                       | packed bitmap fonts                                 | <i>program</i> Fonts ( <i>program</i> being XDVl, etc.), PKFonts, TEXPKS, GLYPHFonts, TEXFonts | .pk                          |
| PostScript header        | downloadable PostScript                             | TEXPSHEADERS, PSHEADERS                                                                        | .pro, .enc                   |
| tex                      | T <sub>E</sub> X source                             | TEXInputs                                                                                      | .tex, .cls, .sty, .clo, .def |
| TeX system documentation | Documentation files for the T <sub>E</sub> X system | TEXDOCS                                                                                        |                              |
| TeX system sources       | Source files for the T <sub>E</sub> X system        | TEXSOURCES                                                                                     |                              |
| texpool                  | T <sub>E</sub> X program strings                    | TEXPOOL, TEXMFINI                                                                              | .pool                        |
| tfm                      | T <sub>E</sub> X font metrics                       | TFMFonts, TEXFonts                                                                             | .tfm                         |
| Troff fonts              | Troff fonts, used by DMP                            | TRFonts                                                                                        |                              |
| truetype fonts           | TrueType outline fonts                              | TTFFonts                                                                                       | .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                                       | VFFFonts, TEXFonts                                                                             | .vf                          |
| web2c files              | Web2c support files                                 | WEB2C                                                                                          |                              |
| other text files         | text files used by ‘foo’                            | FOOInputs                                                                                      |                              |
| other binary files       | binary files used by ‘foo’                          | FOOInputs                                                                                      |                              |

Les deux dernières entrées de la table 1 sont des cas spéciaux, car 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 majuscule, 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 propre environnement d’exécution.

Notons que les options de « --format » et « --path » s’excluent mutuellement.

--mode=*string*

Définit le nom du mode comme étant « *string* » ; ceci affecte seulement la recherche des « gf » et des « pk ». Pas d’option par défaut, n’importe quel mode sera trouvé.

--must-exist

Fait tout ce qui est possible pour trouver les fichiers, ce qui inclut une recherche sur le disque. Par

défaut, seule la base de données ls-R est inspectée, dans un souci d'efficacité.

`--path=string`

Recherche dans le chemin « *string* » (séparé par deux-points comme d'habitude), au lieu de prendre le chemin à partir du nom de fichier. « `//` » et toutes les expansions habituelles sont supportées. Les options « `--path` » et « `--format` » s'excluent mutuellement.

`--progrname=name`

Définit que le nom de programme est « *name* ». Ceci peut affecter les chemins de recherche via l'option « `.progrname` » dans les fichiers de configuration. Le défaut est « `kpsewhich` ».

`--show-path=name`

Montre le chemin utilisé pour la recherche des fichiers de type « *name* ». On peut utiliser 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* ».

### 7.2.3 Exemples d'utilisation

Jetons un coup d'œil 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 rechercher un fichier de type « `tex` » (répertoires des fichiers sources de  $\text{\TeX}$ ). Nous le trouvons dans le sous-répertoire `tex/latex/base` du répertoire racine « `TEXMF` ». De même, le suffixe non-ambigu permet de trouver facilement les autres fichiers.

```
>> 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 exemple est une base de données bibliographiques pour  $\text{\BibTeX}$  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 de fontes 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 trouvée.

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

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

Intéressons-nous à présent aux fichiers d'entête et de configuration pour dvips. 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 liste 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 de la liste :

```
>> 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 liste les noms des fichiers des 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
```

Prenons par exemple 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.

#### 7.2.4 Opérations de débogage

Il est quelquefois nécessaire de savoir comment un programme référence les fichiers. Pour permettre cela de 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 différentes tables (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 globale sur la localisation des types de fichiers recherchés par `Kpathsea`. Ceci est utile pour trouver où a été défini le chemin particulier pour un fichier.
- 16 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 ; en 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'options de débogage, on peut suivre en détail la localisation des différents fichiers. De plus, lorsqu'un fichier n'est pas trouvé, la trace 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 définissant égale à valeur ou une combinaison des valeurs décrites dans la liste ci-dessus.

(Note à l'intention des utilisateurs de Windows : il n'est pas facile de rediriger les messages d'erreur vers un fichier sur ces systèmes. À des fins de diagnostic, vous pouvez temporairement affecter `KPATHSEA_DEBUG_OUTPUT=err.log` pour capturer le flux standard d'erreur dans le fichier `err.log`.)

Considérons comme exemple un petit fichier source  $\text{\LaTeX}$ , `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 ressemblant à la figure 10 (nous avons réarrangé la sortie pour un affichage plus 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 (p.ex., 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 qui définissent la relation entre les noms des fontes selon  $\TeX$ , selon PostScript et dans 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 veut chercher tous ces fichiers, y compris le fichier générique d'association `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 7.2.3 pour plus de détails sur la gestion du fichier d'association 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 ; finalement le fichier de la fonte Type1 `cmr10.pfb` 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/././teTeX/TeX/texmf/web2c:).
kdebug:start search(file=ls-R, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(ls-R) => /usr/local/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(aliases) => /usr/local/texmf/aliases
kdebug:start search(file=config.ps, must_exist=0, find_all=0,
  path=~/.tex:/usr/local/texmf/dvips/).
kdebug:search(config.ps) => /usr/local/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc, must_exist=0, find_all=0,
  path=~/.tex:/usr/local/texmf/dvips/).
search(file=/home/goossens/.dvipsrc, must_exist=1, find_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/).
kdebug:search($HOME/.dvipsrc) =>
kdebug:start search(file=config.cms, must_exist=0, find_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/).
kdebug:search(config.cms)
=>/usr/local/texmf/dvips/cms/config.cms

```

FIGURE 10 – Finding configuration files

```

kdebug:start search(file=texc.pro, must\_exist=0, find\_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/./:
    ~/.tex/fonts/type1/./usr/local/texmf/fonts/type1/).
kdebug:search(texc.pro) => /usr/local/texmf/dvips/base/texc.pro

```

FIGURE 11 – Finding the prolog file

```

kdebug:start search(file=cmr10.tfm, must\_exist=1, find\_all=0,
  path=~/.tex/fonts/tfm/./usr/local/texmf/fonts/tfm/./:
    /var/tex/fonts/tfm/).
kdebug:search(cmr10.tfm) => /usr/local/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must\_exist=0, find\_all=0,
  ...
<texps.pro>
kdebug:start search(file=cmr10.pfb, must\_exist=0, find\_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/./:
    ~/.tex/fonts/type1/./usr/local/texmf/fonts/type1/).
kdebug:search(cmr10.pfb) => /usr/local/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]

```

FIGURE 12 – Finding the font file

```

kdebug:start search(file=cmr10.tfm, must_exist=1, find_all=0,
  path=.:~/tex/fonts/tfm/#!/usr/local/texmf/fonts/tfm/#!/var/tex/fonts/tfm/).
kdebug:search(cmr10.tfm) => /usr/local/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must_exist=0, find_all=0,
  ...
<texps.pro>
kdebug:start search(file=cmr10.pfb, must_exist=0, find_all=0,
  path=.:~/tex/dvips/#!/usr/local/texmf/dvips/#!/tex/fonts/type1/#!/usr/local/texmf/fonts/type1/).
kdebug:search(cmr10.pfb) => /usr/local/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]

```

### 7.3 Options à l'exécution

Web2c offre la possibilité de contrôler à l'exécution bon nombre des paramètres concernant la mémoire (en particulier la taille des tableaux utilisés) à partir du fichier `texmf.cnf` qui est lu par Kpathsea. Le listing de ce fichier est fourni en annexe 11, à partir de la page 51 ; 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 et 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 `hugetex.fmt`. En utilisant la méthode supportée par Kpathsea qui consiste à 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` sous le nom `main_memory.hugetex` (comparer la valeur générique à la valeur spécifique pour le format `hugetex`).

**extra\_mem\_bot** Espace supplémentaire pour certaines structures de données de T<sub>E</sub>X : boîtes, glue, points d'arrêt... Surtout utile si vous utilisez P<sub>l</sub>CT<sub>E</sub>X 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 `hugetex` 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.

## 8 Remerciements

Cette édition du TeXLive a été coordonnée par Sebastian Rahtz ; les principaux contributeurs étant Fabrice Popineau, qui a travaillé sans relâche à la partie Win32 de la distribution (surtout le programme d'installation !) et a contribué de différentes manières par ses idées, ses conseils et du code), et Staszek Wawrykiewicz, qui a fourni un énorme travail sous forme de rapports de tests et a coordonné



les contributions polonaises. Kaja Christiansen a eu un rôle important en recompilant sans cesse la distribution sur tout un assortiment de plate-formes Unix, et Robin Laasko a coordonné la production pour le TUG. Vladimir Volovich a fait un superbe travail en nettoyant les sources et en apportant des améliorations, tandis que Gerben Wierda a fait tout le travail pour MacOSX.

Pour leur aide présente et passée, nous sommes particulièrement reconnaissants à :

- Dante e.V., le groupe allemand d'utilisateurs de  $\text{\TeX}$  qui nous fournit une machine sur laquelle le CD-ROM source est développé et maintenu, ainsi que Rainer Schöpf et Reinhard Zierke qui veillent à son bon fonctionnement ;
- Perforce, qui nous a fourni gratuitement une licence de l'excellent logiciel de développement concurrent que l'on utilise pour gérer le contenu du CD-ROM ;
- Karl Berry, pour ses conseils et ses encouragements, et son aide. Il fut à l'origine de la première distribution de Web2c ;
- Mimi Burbank, qui nous favorisa l'accès au Florida State University School of Computational Science and Information Technology à différents ordinateurs pour compiler  $\text{\TeX}$  et qui joua le rôle de cobaye chaque fois qu'on le lui demanda ;
- Kaja Christiansen, qui fournit des remarques essentielles et nous assista pour la documentation et la compilation ;
- Thomas Esser, car sans lui, sa merveilleuse distribution  $\text{te}\text{\TeX}$  et son aide continuelle, ce CD-ROM n'existerait probablement pas ;
- Michel Goossens, en tant que co-auteur de la documentation ;
- Eitan Gurari, dont le  $\text{\TeX}4\text{ht}$  a été utilisé pour créer la version HTML de cette documentation, et qui travailla inlassablement à l'améliorer en peu de temps ;
- Art Ogawa et Pat Monohon qui ont coordonné la sortie des versions pour le TUG ;
- Petr Olsak, qui coordonna et vérifia minutieusement toute la partie tchèque et slovaque ;
- Olaf Weber, pour son patient assemblage et la maintenance de Web2c ;
- Graham Williams, du travail duquel dépend le catalogue des composants.

Gerhard Wilhelms, Volker Schaa, Fabrice Popineau, Janka Chlebíková, Staszek Wawrykiewicz, Erik Frambach, and Ulrik Vieth ont traduit la documentation dans leurs langues respectives, relu les autres documentations et nous ont beaucoup apporté.

## 9 Historique

Cette distribution sous forme de CD-ROM est le résultat des efforts collectifs des groupes d'utilisateurs de  $\text{\TeX}$ . La discussion commença à la fin de 1993 quand le Groupe des Utilisateurs Néerlandais de  $\text{\TeX}$  commençait à travailler à son 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 d'un groupe de travail « TUG Technical Council » pour mettre en place une TDS ( *$\text{\TeX}$  Directory Structure*), qui spécifia la gestion sous une forme logique des fichiers pour  $\text{\TeX}$ . La mouture finale de la TDS fut publiée dans le numéro de Décembre 1995 de *TUGboat*, et il était clair depuis un certain temps qu'il était souhaitable de proposer un produit contenant 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 but du  **$\text{\TeX}$  Live**.

Nous avons entrepris de créer un nouveau CD-ROM TDS Unix à l'automne 1995, et nous avons

rapidement choisi le  $\text{te}\text{\LaTeX}$  de Thomas Esser comme étant la configuration idéale, car il supportait déjà plusieurs plate-formes et il avait été construit en gardant à l'esprit la portabilité entre les systèmes. Thomas accepta de nous aider, et commença sérieusement à travailler au début de 1996. La première édition est sortie en mai 1996. Au début de 1997, Karl Berry acheva une nouvelle distribution de son composant Web2c, qui incluait presque toutes les caractéristiques que Thomas Esser avait ajoutées dans  $\text{te}\text{\LaTeX}$ , et nous avons décidé de baser la deuxième édition du CD-ROM sur le standard Web2c, en y ajoutant le script `texconfig` de  $\text{te}\text{\LaTeX}$ . 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  $\text{te}\text{\LaTeX}$  était achevée, et  **$\text{\LaTeX}$  Live** partage presque toutes ses caractéristiques. La quatrième édition a suivi le même schéma, en utilisant une nouvelle version de  $\text{te}\text{\LaTeX}$ , et une nouvelle version de Web2c (7.3). Le système incluait dorénavant un programme d'installation pour Windows.

Pour la cinquième édition (Mars 2000), beaucoup de parties du CD-ROM avaient été vérifiées et révisées, des centaines de composants avaient été mis à jour. Le contenu détaillé des composants était décrit par des fichiers XML. Mais le changement majeur de cette cinquième édition a été la suppression de tout le logiciel non libre de droits. Tout ce qui se trouve sur ce CD-ROM devrait être compatible avec 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.

La sixième édition (juillet 2001) contient les révisions disponibles de tous les éléments. Le changement majeur de cette version tient dans la refonte du processus d'installation : l'utilisateur peut désormais choisir de manière plus exacte les collections. Les collections concernant les langues ont été totalement réorganisées, aussi le choix d'une langue installe non seulement les macros, les fontes, etc. mais prépare également un fichier `language.dat` adéquat.

La septième édition (mai 2002) a comme ajout majeur, une installation pour MacOSX, et l'habituelle myriade de mises à jour de composants et de programmes. Un objectif important a été rempli : rejoindre le développement de  $\text{te}\text{\LaTeX}$  à nouveau, alors que les versions 5 et 6 s'en étaient éloignées.

Gerhard Wilhelms, Volker Schaa, Fabrice Popineau<sup>10</sup>, Janka Chlebíková, Staszek Wawrykiewicz, Erik Frambach, et Ulrik Vieth ont gentiment traduit la documentation dans leurs langues respectives, vérifièrent d'autres documentations, et donnèrent un retour très apprécié.

## 10 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 vous annoncer !

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

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

---

<sup>10</sup> Assisté de différents membres de GUTenberg.

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

## 11 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:othertex}
36 % expands to /usr/local/mytex:/usr/local/othertex. Instead of the path
37 % separator you can use a comma: /usr/local/{mytex,othertex} also expands
38 % to /usr/local/mytex:/usr/local/othertex. However, the use of the comma
39 % instead of the path separator is deprecated.
40 %
41 % The text above assumes the 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 VARTEXTFonts =              $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;$TEXMFMAIN;$VARTEXTFonts
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 VARTEXTFonts like this
108 %   VARTEXTFonts =          $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 = .;$TEXMF/{etex,tex}/{plain,generic,}//
142
143 % PDFTeX. This form of the input paths is borrowed from TeTeX. A certain
144 % variant of TDS is assumed here, unaffected by the build variables.
145 TEXINPUTS.pdfetexinfo = .;$TEXMF/{pdfetex,tex}/{texinfo,plain,generic,}//
146 TEXINPUTS.pdfplatex = .;$TEXMF/{pdfetex,tex}/{latex,generic,}//
147 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,tex}/{plain,generic,}//
148 TEXINPUTS.pdfelatex = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{latex,generic,}//
149 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{plain,generic,}//
150
151 % Omega.
152 TEXINPUTS.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/{pdfetex,tex}/{cslatex,csplain,latex,generic,}//
162 TEXINPUTS.pdfcsplain = .;$TEXMF/{pdfetex,cstex,tex}/{csplain,plain,generic,}//
163
164 % Polish
165 TEXINPUTS.platex = .;$TEXMF/{tex}/{platex,latex,generic,}//
166 TEXINPUTS.pdfplatex = .;$TEXMF/{pdfetex,tex}/{platex,latex,generic,}//
167 TEXINPUTS.pdfmex = .;$TEXMF/{pdfetex,tex}/{mex,plain,generic,}//
168 TEXINPUTS.pdfmex = .;$TEXMF/{pdfetex,pdftex,tex}/{mex,plain,generic,}//
169 TEXINPUTS.mex = .;$TEXMF/tex/{mex,plain,generic,}//
170
171 % french
172 TEXINPUTS.frlatex = .;$TEXMF/{mltex,tex}/{french,latex,generic,}//
173 TEXINPUTS.frtex = .;$TEXMF/{mltex,tex}/{french,plain,generic,}//
174 TEXINPUTS.frpdlatex = .;$TEXMF/{mltex,pdftex,tex}/{french,latex,generic,}//
175 TEXINPUTS.frpdlatex = .;$TEXMF/{mltex,pdftex,tex}/{french,plain,generic,}//
176
177 % MLTeX
178 TEXINPUTS.mltex = .;$TEXMF/{mltex,tex}/{plain,generic,}//
179 TEXINPUTS.mllatex = .;$TEXMF/{mltex,tex}/{latex,generic,}//
180
181 % odd formats needing their own paths
182 TEXINPUTS.lollipop = .;$TEXMF/tex/{lollipop,generic,plain,}//
183 TEXINPUTS.lamstex = .;$TEXMF/tex/{lamstex,generic,plain,}//
184
185 % David Carlisle's xmltex
186 TEXINPUTS.xmltex = .;$TEXMF/tex/{xmltex,latex,generic,}//
187 TEXINPUTS.pdfxmltex = .;$TEXMF/{pdfetex,tex}/{xmltex,latex,generic,}//
188
189 % Sebastian Rahtz' jadetex for DSSSL
190 TEXINPUTS.pdfjadetex = .;$TEXMF/{pdfetex,tex}/{jadetex,generic,plain,}//
191 TEXINPUTS.jadetex = .;$TEXMF/tex/{jadetex,generic,plain,}//
192
193 % Earlier entries override later ones, so put this last.
194 TEXINPUTS = .;$TEXMF/tex/{generic,}//
195
196 % Metafont, MetaPost inputs.

```

```

197 MFINPUTS =                .;$TEXMF/metafont//;{$TEXMF/fonts,$VARTEXFONTS}/source//
198 MPINPUTS =                .;$TEXMF/metapost//
199
200 % mft
201 MFTINPUTS =                .;$TEXMF/mft//
202
203 % Web and CWeb input paths.
204 WEBINPUTS =                .;$TEXMF/web//
205 CWEBINPUTS =               .;$TEXMF/cweb//
206
207 % Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see web2c/INSTALL),
208 % and string pools (.pool) for ini{tex,mf,mp}. It is silly that we have six
209 % paths and directories here (they all resolve to a single place by default),
210 % but historically ...
211 TEXFORMATS =               .;$TEXMF/web2c
212 MFBASES =                  $TEXFORMATS
213 MPMEMS =                   $TEXFORMATS
214 TEXPOOL =                  $TEXFORMATS
215 MFPOOL =                   $TEXFORMATS
216 MPPPOOL =                  $TEXFORMATS
217
218 % Device-independent font metric files.
219 VFFONTS =                  .;$TEXMF/fonts/vf//
220 TFMFONTS =                 .;{$TEXMF/fonts,$VARTEXFONTS}/tfm//
221
222 % The $MAKETEX_MODE below means the drivers will not use a cx font when
223 % the mode is ricoh. If no mode is explicitly specified, kpse_prog_init
224 % sets MAKETEX_MODE to /, so all subdirectories are searched. See the manual.
225 % The modeless part guarantees that bitmaps for PostScript fonts are found.
226 PKFONTS =                  .;{$TEXMF/fonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modelless}//
227
228 % Similarly for the GF format, which only remains in existence because
229 % Metafont outputs it (and MF isn't going to change).
230 GFFONTS =                  .;$TEXMF/fonts/gf/$MAKETEX_MODE//
231
232 % A backup for PKFONTS and GFFONTS. Not used for anything.
233 GLYPHFONTS =               .;$TEXMF/fonts
234
235 % For texfonts.map and included map files used by mktexpk.
236 % See ftp://ftp.tug.org/tex/fontname.tar.gz.
237 TEXFONTMAPS =              .;$TEXMF/fontname
238
239 % BibTeX bibliographies and style files.
240 BIBINPUTS =                .;$TEXMF/bibtex/{bib,}//
241 BSTINPUTS =                .;$TEXMF/bibtex/{bst,}//
242
243 % PostScript headers, prologues (.pro), encodings (.enc) and fonts;
244 % this is also where pdftex finds included figures files!
245
246 TEXPSHEADERS.pdflatex =    .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
247 TEXPSHEADERS.pdfelatex =   .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
248 TEXPSHEADERS.pdftexinfo =  .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
249 TEXPSHEADERS.pdfslatex =   .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
250 TEXPSHEADERS.pdfcsplain =  .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
251 TEXPSHEADERS.pdfetex =     .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
252 TEXPSHEADERS.pdfjadetex =  .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
253 TEXPSHEADERS.pdfplatex =   .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
254 TEXPSHEADERS.pdfxmltex =    = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
255 TEXPSHEADERS.pdfmex =      .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
256 TEXPSHEADERS.pdfTeX =      .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
257 TEXPSHEADERS.pdftexinfo =  .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
258 TEXPSHEADERS.cont-de =     .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
259 TEXPSHEADERS.cont-en =     .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
260 TEXPSHEADERS.cont-nl =     .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
261 TEXPSHEADERS.context =     .;$TEXMF/{etex,tex,pdftex,dvips,fonts/{type1,pfb}}//
262 TEXPSHEADERS =             .;$TEXMF/{dvips,fonts/{type1,pfb},pdftex}//
263
264 % PostScript Type 1 outline fonts.

```

```

265 T1FONTS =                .;$TEXMF/fonts/{type1,pfb};//;$TEXMF/fonts/misc/hbf//
266
267 % PostScript AFM metric files.
268 AFMFONTS =                .;$TEXMF/fonts/afm//
269
270 % TrueType outline fonts.
271 TTFONTS =                  .;$TEXMF/fonts/{truetype,ttf}//
272 TTF2TFMINPUTS =            .;$TEXMF/ttf2pk//
273
274 % Type 42 outline fonts.
275 T42FONTS =                  .;$TEXMF/fonts/type42//
276
277 % A place to puth everything that doesn't fit the other font categories.
278 MISCFONTS =                .;$TEXMF/fonts/misc//
279
280 % Dvips' config.* files (this name should not start with 'TEX'!).
281 TEXCONFIG =                .;$TEXMF/dvips//
282
283 % Makeindex style (.ist) files.
284 INDEXSTYLE =                .;$TEXMF/makeindex//;$TEXMF/tex//
285
286 % Used by DMP (ditroff-to-mpx), called by makempx -troff.
287 TRFONTS =                  /usr/lib/font/devpost
288 MPSUPPORT =                .;$TEXMF/metapost/support
289
290 % For xdvi to find mime.types and .mailcap, if they do not exist in
291 % $HOME. These are single directories, not paths.
292 % (But the default mime.types, at least, may well suffice.)
293 MIMELIBDIR =                $SELFAUTOPARENT/etc
294 MAILCAPLIBDIR =            $SELFAUTOPARENT/etc
295
296 % TeX documentation and source files, for use with kpsewhich.
297 TEXDOCS =                  .;$TEXMF/doc//
298 TEXSOURCES =                .;$TEXMF/source//
299
300 % allow for compressed files, and various extenions
301 TEXDOCSSUFFIX =              :.dvi:.pdf:.ps:.html:.txt:.tex
302 TEXDOCSCOMPRESS =            :.gz:.bz2:.zip:.Z
303 TEXDOCEXT =                  {$TEXDOCSSUFFIX}{$TEXDOCSCOMPRESS}
304
305 % Omega-related fonts and other files. The odd construction for OFMFONTS
306 % makes it behave in the face of a definition of TFMFONTS. Unfortunately
307 % no default substitution would take place for TFMFONTS, so an explicit
308 % path is retained.
309 OFMFONTS =                  .;{$TEXMF/fonts,$VARTEXFONTS}/{ofm,tfm};//;$TFMFONTS
310 OPLFONTS =                  .;{$TEXMF/fonts,$VARTEXFONTS}/opl//
311 OVFFONTS =                  .;{$TEXMF/fonts,$VARTEXFONTS}/ovf//
312 OVPFONTS =                  .;{$TEXMF/fonts,$VARTEXFONTS}/ovp//
313 OTPINPUTS =                  .;$TEXMF/omega/otp//
314 OCPINPUTS =                  .;$TEXMF/omega/ocp//
315
316 %dvipdfm
317 DVIPDFMINPUTS =              .;$TEXMF/dvipdfm//
318
319 %% t4ht utility, sharing files with TeX4ht
320 TEX4HTFONTSET=alias,iso8859
321 TEX4HTINPUTS =                .;$TEXMF/tex4ht/base//;$TEXMF/tex4ht/ht-fonts/{$TEX4HTFONTSET}//
322 T4HTINPUTS=                  .;$TEXMF/tex4ht/base//
323 %% The mktex* scripts rely on KPSE_DOT. Do not set it in the environment.
324
325 XDVIINPUTS=.$TEXMF/{xdvi,dvips}//
326 KPSE_DOT =                    .
327
328 % This definition isn't used from this .cnf file itself (that would be
329 % paradoxical), but the compile-time default in paths.h is built from it.
330 % The SELFAUTO* variables are set automatically from the location of
331 % argv[0], in kpse_set_prognam.
332 %

```

```

333 % About the /. construction:
334 % 1) if the variable is undefined, we'd otherwise have an empty path
335 % element in the compile-time path. This is not meaningful.
336 % 2) if we used /$VARIABLE, we'd end up with // if VARIABLE is defined,
337 % which would search the entire world.
338 %
339 % The TETEXDIR stuff isn't likely to be relevant unless you're using teTeX,
340 % but it doesn't hurt.
341 %
342 TEXMFCNF = .;$VARTEXMF/web2c;{$SELFAUTOLOC,$SELFAUTODIR,$SELFAUTOPARENT}\
343 {,{/share,}/texmf{.local,}/web2c};c:/TeX/texmf/web2c
344
345
346 % Suggestions for editor settings under Windows. Uncomment your
347 % preferred option. The corresponding MFEDIT can also be set for use with
348 % Metafont.
349 %
350 % Winedt:
351 % TEXEDIT=C:\WinEdt\WinEdt.exe "[Open('%s');SellLine(%d,7)]
352 % Textpad:
353 % TEXEDIT = c:\Progra~1\TextPad\System\Ddeopn32 TextPad %s(%d)
354 % UltraEdit (newer Win32 versions):
355 % TEXEDIT = uedit32 %s/%d/1
356 % WinTeXShell32:
357 % TEXEDIT = texshell.exe /l=%d %s
358 % vi, vim, gvim. here we show Windows gvim.exe:
359 % TEXEDIT = gvim.exe %s +%d
360 % PFE:
361 % TEXEDIT=pfe32/g%d %s
362 % MED:
363 % TEXEDIT=med.exe "%s" %d
364 % TSE:
365 % TEXEDIT=e32.exe "%s" -n%d
366 % Epsilon (Lugaru) http://www.lugaru.com/
367 % TEXEDIT="c:\Program Files\eps90\bin\e32.exe" +%d %s
368 % WinShell
369 % TEXEDIT=C:\Progra~1\WinShell\WinShell.exe -c %s -l %d
370
371 % For unix
372 %
373 % vi, vim, NEdit, (X)Emacs, pico, jed
374 % TEXEDIT = vi +%d %s
375 % TEXEDIT = vim +%d %s
376 % TEXEDIT = nedit +%d %s
377 % TEXEDIT = xemacs +%d %s
378
379 %(x)fte:
380 % TEXEDIT = xfte -l%d %s
381
382
383 %-----
384 % Write .log/.dvi/etc. files here, if the current directory is unwritable.
385 % TEXMFOUTPUT = /tmp
386
387 % If a dynamic file creation fails, log the command to this file, in
388 % either the current directory or TEXMFOUTPUT. Set to the
389 % empty string or 0 to avoid logging.
390 MISSFONT_LOG = missfont.log
391
392 % Set to a colon-separated list of words specifying warnings to suppress.
393 % To suppress everything, use TEX_HUSH = all; this is equivalent to
394 % TEX_HUSH = checksum:lostchar:readable:special
395 TEX_HUSH = none
396
397 % Enable system commands via \write18{...}?
398 shell_escape = f
399
400 % Allow TeX \openout/\openin on filenames starting with '.' (e.g., .rhosts)?

```



```

401 % a (any)      : any file can be opened.
402 % r (restricted) : disallow opening "dotfiles".
403 % p (paranoid)  : as 'r' and disallow going to parent directories, and
404 %                restrict absolute paths to be under $TEXMFOUTPUT.
405 openout_any =      p
406 openin_any =       a
407 % Allow TeX, MF, and MP to parse the first line of an input file for
408 % the %&format construct.
409 parse_first_line =      t
410
411 % Allow TeX, eTeX, Omega to include 'src:' specials in the dvi file.
412 % These specials are used by viewers to jump from the viewer into
413 % the editor at the right page/lineno.
414 % Possible values : none auto cr display hbox math par parend vbox
415 src_specials =      none
416
417 % Disable search on multiple suffixes filenames. In many case, when 'foo.bar'
418 % is looked for, you do not want to look for 'foo.bar.tex' before. This flag
419 % disables searching for standard suffixes if the file name has already an
420 % extension of 3 characters. Default value is true (old behaviour).
421 % allow_multiple_suffixes =      f
422
423 % Enable the mktex... scripts by default? These must be set to 0 or 1.
424 % Particular programs can and do override these settings, for example
425 % dvips's -M option. Your first chance to specify whether the scripts
426 % are invoked by default is at configure time.
427 %
428 % These values are ignored if the script names are changed; e.g., if you
429 % set DVIPSMMAKEPK to 'foo', what counts is the value of the environment
430 % variable/config value 'FOO', not the 'MKTEXPK' value.
431 %
432 % MKTEXTEX =      0
433 % MKTEXPK =      0
434 % MKTEXMF =      0
435 % MKTEXTFM =      0
436 % MKOCP =      0
437 % MKOFM =      0
438
439 % What MetaPost runs to make MPX files. This is passed an option -troff
440 % if MP is in troff mode. Set to '0' to disable this feature.
441 MPXCOMMAND =      makempx
442
443
444 % Part 3: Array and other sizes for TeX (and Metafont and MetaPost).
445 %
446 % If you want to change some of these sizes only for a certain TeX
447 % variant, the usual dot notation works, e.g.,
448 % main_memory.hugetex =      20000000
449 %
450 % If a change here appears to be ignored, try redumping the format file.
451 %
452
453 %-----
454 % Memory. Must be less than 8,000,000 total.
455 %
456 % main_memory is relevant only to initex, extra_mem_* only to non-ini.
457 % Thus, have to redump the .fmt file after changing main_memory; to add
458 % to existing fmt files, increase extra_mem_*. (To get an idea of how
459 % much, try \tracingstats=2 in your TeX source file;
460 % web2c/tests/memtest.tex might also be interesting.)
461 %
462 % To increase space for boxes (as might be needed by, e.g., PiCTeX),
463 % increase extra_mem_bot.
464 %
465 %
466 main_memory =      1500000 % words of inmemory available; also applies to inimg&mp
467 extra_mem_top =      0 % extra high memory for chars, tokens, etc.
468 extra_mem_bot =      0 % extra low memory for boxes, glue, breakpoints, etc.

```

```

469
470 % Words of font info for TeX (total size of all TFM files, approximately).
471 font_mem_size =                200000
472
473 % Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
474 font_max =                    1000
475
476 % Extra space for the hash table of control sequences (which allows 10K
477 % names as distributed).
478 hash_extra =                  15000
479
480 % Max number of characters in all strings, including all error messages,
481 % help texts, font names, file names, control sequences.
482 % These values apply to TeX and MP.
483
484 % Minimum pool space after TeX/MP's own strings; must be at least
485 % 25000 less than pool_size, but doesn't need to be nearly that large.
486 string_vacancies =            45000
487 max_strings =                  65000                % max number of strings
488 pool_free =                    47500                % min pool space left after loading .fmt
489 pool_size =                    125000
490 % Hyphenation trie. As distributed, the maximum is 65535; this should
491 % work unless 'unsigned short' is not supported or is smaller than 16
492 % bits. This value should suffice for UK English, US English, French,
493 % and German (for example). To increase, you must change
494 % 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
495 % the trie will then consume four bytes per entry, instead of two.
496 %
497 % US English, German, and Portuguese: 30000.
498 % German: 14000.
499 % US English: 10000.
500 %
501 trie_size =                    262000
502
503 % Buffer size. TeX uses the buffer to contain input lines, but macro
504 % expansion works by writing material into the buffer and reparsing the
505 % line. As a consequence, certain constructs require the buffer to be
506 % very large. As distributed, the size is 50000; most documents can be
507 % handled within a tenth of this size.
508 buf_size =                     200000
509
510 % Parameter specific to MetaPost.
511 % Maximum number of knots between breakpoints of a path.
512 % Set to 2000 by default.
513 % path_size.mpost =            10000
514
515 % These are pdftex-specific.
516 obj_tab_size =                  200000 % PDF objects
517 dest_names_size=300000 % destinations
518
519 % These are Omega-specific.
520 ocp_buf_size =                  500000                % character buffers for ocp filters.
521 ocp_stack_size =                10000                % stacks for ocp computations.
522 ocp_list_size =                 1000                 % control for multiple ocps.
523
524 % These work best if they are the same as the I/O buffer size, but it
525 % doesn't matter much. Must be a multiple of 8.
526 dvi_buf_size =                  16384                % TeX
527 gf_buf_size =                   16384                % MF
528
529 % It's probably inadvisable to change these. At any rate, we must have:
530 % 45 < error_line < 255;
531 % 30 < half_error_line < error_line - 15;
532 % 60 <= max_print_line;
533 % These apply to Metafont and MetaPost as well.
534 error_line =                    79
535 half_error_line =               50
536 max_print_line =               79

```

```

537 stack_size =          1500 % simultaneous input sources
538 save_size =          5000 % for saving values outside current group
539 param_size =          1500 % simultaneous macro parameters
540 max_in_open =          15 % simultaneous input files and error insertions
541 hyph_size =          1000 % number of hyphenation exceptions, >610 and <32767
542 nest_size =           500 % simultaneous semantic levels (e.g., groups)
543
544
545 %-----
546 %
547 main_memory.hugetex =          1500000
548 param_size.hugetex =           1500
549 stack_size.hugetex =           1500
550 hash_extra.hugetex =          15000
551 string_vacancies.hugetex =      45000
552 pool_free.hugetex =           47500
553 nest_size.hugetex =            500
554 save_size.hugetex =           5000
555 pool_size.hugetex =          1250000
556 max_strings.hugetex =           65000
557
558 main_memory.mf =              800000
559 main_memory.mpost =           1000000
560 pool_size.mpost =             500000
561
562 buf_size.context =            200000 % needed for omega bug
563 extra_mem_bot.context =       4000000
564 extra_mem_top.context =       2000000
565 font_mem_size.context =       500000
566 hash_extra.context =          50000
567 main_memory.context =         1500000
568 max_strings.context =         100000
569 nest_size.context =            500
570 obj_tab_size.context =        300000
571 param_size.context =           5000
572 pool_free.context =           47500
573 pool_size.context =          1250000
574 save_size.context =           50000
575 stack_size.context =           5000
576 string_vacancies.context =     90000
577 % Context's metafun
578 main_memory.metafun =         2500000
579 pool_size.metafun =           1000000
580
581 % cslatex
582 main_memory.cslatex =          1500000
583 param_size.cslatex =           1500
584 stack_size.cslatex =           1500
585 hash_extra.cslatex =          15000
586 string_vacancies.cslatex =      45000
587 pool_free.cslatex =           47500
588 nest_size.cslatex =            500
589 save_size.cslatex =           5000
590 pool_size.cslatex =          1250000
591 max_strings.cslatex =           55000
592 font_mem_size.cslatex=         400000
593
594 main_memory.lambda =           1500000
595
596 % redundant. all LaTeX should be huge
597 main_memory.hugelatex =        1500000
598 param_size.hugelatex =          1500
599 stack_size.hugelatex =          1500
600 hash_extra.hugelatex =         15000
601 string_vacancies.hugelatex =    45000
602 pool_free.hugelatex =           47500
603 nest_size.hugelatex =            500
604 save_size.hugelatex =           5000

```

```

605 pool_size.hugelatex = 1250000
606 max_strings.hugelatex = 55000
607 font_mem_size.hugelatex= 400000
608
609 % standard LaTeX is itself huge
610
611 main_memory.latex = 1500000
612 param_size.latex = 1500
613 stack_size.latex = 1500
614 hash_extra.latex = 15000
615 string_vacancies.latex = 45000
616 pool_free.latex = 47500
617 nest_size.latex = 500
618 save_size.latex = 5000
619 pool_size.latex = 1250000
620 max_strings.latex = 55000
621 font_mem_size.latex= 400000
622
623 main_memory.jadetex = 1500000
624 param_size.jadetex = 1500
625 stack_size.jadetex = 1500
626 hash_extra.jadetex = 15000
627 string_vacancies.jadetex = 45000
628 pool_free.jadetex = 47500
629 nest_size.jadetex = 500
630 save_size.jadetex = 5000
631 pool_size.jadetex = 1250000
632 max_strings.jadetex = 55000
633 font_mem_size.jadetex= 400000
634
635
636 main_memory.pdfjadetex = 2500000
637 param_size.pdfjadetex = 1500
638 stack_size.pdfjadetex = 1500
639 hash_extra.pdfjadetex = 50000
640 string_vacancies.pdfjadetex = 55000
641 pool_free.pdfjadetex = 47500
642 nest_size.pdfjadetex = 500
643 save_size.pdfjadetex = 5000
644 pool_size.pdfjadetex = 1250000
645 max_strings.pdfjadetex = 55000
646
647 main_memory.xmltex = 1500000
648 param_size.xmltex = 1500
649 stack_size.xmltex = 1500
650 hash_extra.xmltex = 50000
651 string_vacancies.xmltex = 45000
652 pool_free.xmltex = 47500
653 nest_size.xmltex = 500
654 save_size.xmltex = 50000
655 pool_size.xmltex = 1250000
656 max_strings.xmltex = 55000
657
658 main_memory.pdfxmltex = 2500000
659 param_size.pdfxmltex = 1500
660 stack_size.pdfxmltex = 1500
661 hash_extra.pdfxmltex = 50000
662 string_vacancies.pdfxmltex = 45000
663 pool_free.pdfxmltex = 47500
664 nest_size.pdfxmltex = 500
665 save_size.pdfxmltex = 50000
666 pool_size.pdfxmltex = 1250000
667 max_strings.pdfxmltex = 55000
668
669 font_mem_size.pdflatex = 210000
670 main_memory.pdflatex = 1500000
671 param_size.pdflatex = 3000
672 stack_size.pdflatex = 3000

```

```

673 hash_extra.pdflatex = 15000
674 string_vacancies.pdflatex = 55000
675 pool_free.pdflatex = 47500
676 nest_size.pdflatex = 500
677 pool_size.pdflatex = 1250000
678 save_size.pdflatex = 50000
679 max_strings.pdflatex = 55000
680
681 main_memory.pdfelatex = 1500000
682 param_size.pdfelatex = 1500
683 stack_size.pdfelatex = 1500
684 hash_extra.pdfelatex = 15000
685 string_vacancies.pdfelatex = 45000
686 pool_free.pdfelatex = 47500
687 nest_size.pdfelatex = 500
688 pool_size.pdfelatex = 1250000
689 save_size.pdfelatex = 50000
690 max_strings.pdfelatex = 55000
691
692 main_memory.pdfetex = 1500000 % 1000000 bot/top
693 hash_extra.pdfetex = 50000
694 pool_size.pdfetex = 1250000
695 string_vacancies.pdfetex = 90000
696 max_strings.pdfetex = 100000
697 pool_free.pdfetex = 47500
698 nest_size.pdfetex = 500
699 param_size.pdfetex = 5000
700 save_size.pdfetex = 50000
701 stack_size.pdfetex = 5000
702 obj_tab_size.pdfetex = 256000

```