MiKTeX Manual

Revision 2.0 (MiKTeX 2.0)
December 2000

Christian Schenk <cs@miktex.org>

1 What is MiKTeX?

1.1 MiKTeX Features

MiKTeX is a TeX distribution for Windows (95/98/NT/2000). Its main features include:

- Native Windows implementation with support for long file names.
- On-the-fly generation of missing fonts.
- TDS (TeX directory structure) compliant.
- Open Source.
- Advanced TeX compiler features:
  - TeX can insert source file information (aka source specials) into the DVI file. This feature improves Editor/Previewer interaction.
  - TeX is able to read compressed (gzipped) input files.
  - The input encoding can be changed via TCX tables.
- Previewer features:
  - Supports graphics (PostScript, BMP, WMF, TPIC, . . .)
  - Supports colored text (through color specials)
  - Supports PostScript fonts
  - Supports TrueType fonts
  - Understands HyperTeX (`html:`) specials
  - Understands source (`src:`) specials
  - Customizable magnifying glasses
- MiKTeX is network friendly:
  - integrates into a heterogeneous TeX environment
  - supports UNC file names
  - supports multiple TEXMF directory trees
  - uses a file name database for efficient file access
  - Setup Wizard can be run unattended

The MiKTeX distribution consists of the following components:

- TeX: The traditional TeX compiler.
- e-TeX: A feature-extended version of TeX.
- Yap: DVI previewer.
- pdfTeX: A variant of TeX that creates PDF files.
- dvipdfm: Converts DVI (TeX output) files into PDF.
- Omega: An enhanced version of TeX with support for 16-bit character sets.
- METAFONT: Converts font specifications into raster fonts.
- MetaPost: Converts picture specifications into PostScript commands.

- dvips: Converts DVI (TeX output) files into PostScript.
- MakeIndex: Composes indexes.
- BibTeX: Composes bibliographies.
- Standard LaTeX Packages: AMS-LaTeX, Babel, PSNFSS, ...
- TeXinfo, PSutils, . . .: Lots of utilities.

1.2 How to get MiKTeX

MiKTeX

The official (i.e. most recent & stable) MiKTeX release is archived in the CTAN[1] directory

`'systems/win32/miktex/'`.

Visit the Project Page (see Section 1.3 [Project Page], page 2) for detailed download instructions.

Other Packages

Here is a list of other packages you should take into consideration:

Aladdin Ghostscript (http://www.cs.wisc.edu/~ghost/aladdin/index.html)
: Ghostscript is an interpreter for the PostScript language. Yap uses Ghostscript to display embedded EPS graphics.

Adobe Acrobat Reader (http://www.adobe.com/prodindex/acrobat/readstep.html)
: A PDF viewer.

WinEdt (http://www.winedt.com)
: WinEdt is a shareware TeX editor/shell. It cooperates with MiKTeX with respect to forward and inverse DVI search (see Section 5.2 [Source Specials], page 25).

ActivePerl (http://www.activestate.com)
: ActivePerl is an implementation of Perl for the Windows platform. A few MiKTeX utilities (e.g. `psmerge`) are Perl scripts. You should install Perl if you want to use these utilities.

1.3 The MiKTeX Project Page

You can visit the MiKTeX Project Page for information about new releases, patches and so on.

[1] CTAN: Comprehensive TeX Archive Network

1.4 The MiKTeX Mailing List

MiKTeX Mailing List

There is a discussion list for MiKTeX. To join this list, send an e-mail to <miktex-request@dsts.dk> which contains the word **subscribe** as the first line in the message body.

This list is archived at www.egroups.com.

1.5 Documentation

The MiKTeX Manual (which you are reading right now) concentrates on documenting MiKTeX specific features.

Other MiKTeX related documentation includes:

Frequently Asked Questions
: Lists answers to frequently asked questions.

Tips & Tricks
: Lists useful tips.

Shortcuts to these documents can be found in the MiKTeX program folder (see Section 3.2 [Start Menu], page 16).

2 What's new in MiKTEX 2.0?

New Applications

- MiKTeX Options: A graphical front-end to the MiKTEX configuration utility (`initexmf`).
- Remove MiKTEX! Wizard: Assists in removing MiKTEX from the computer.
- pdfeTEX 0.14f/2.1: A pdfTEX/e-TEX merger.

Updated Applications

- Dvipdfm 0.13.2
- LaTEX 2000/06/01
- Omega 1.11
- pdfTEX 0.14f

Setup Wizard Changes

- You can specify the order by which the TEXMF root directories are searched.
- The MiKTEX bin directory is added to the `PATH` environment variable.

File Searching Changes

- It is now possible to place the local root directory (usually '`C:\Local TeXMF`') in the front of the search path.
- On-the-fly generated files (e.g. PK fonts) are automatically added to the file name database.

TEX Compiler Changes

New Options

- '`--include-directory=`*dir*' prepends *dir* to the search path.
- '`--output-directory=`*dir*' sets the destination directory for all output files.
- '`--enable-write18`' enables the `\write18` primitive (see below).
- `\write18` primitive: starts a command interpreter to carry out the specified command

New Features

- TEX automatically renews the format file, if it is unacceptable. This should eliminate the '`(Fatal format file error; I'm stymied)`' problem.

Texify Changes

- New option '`--max-iterations=`*n*' limits the number of iterations. This prevents endless processing. The default for *n* is 5.

Yap Changes

- New button '`Double-Page`' to turn on double-page view (view two pages side-by-side).
- Yap supports EEPIC drawing primitives (also known as *tpic* specials).
- Presentation (full-screen) mode.

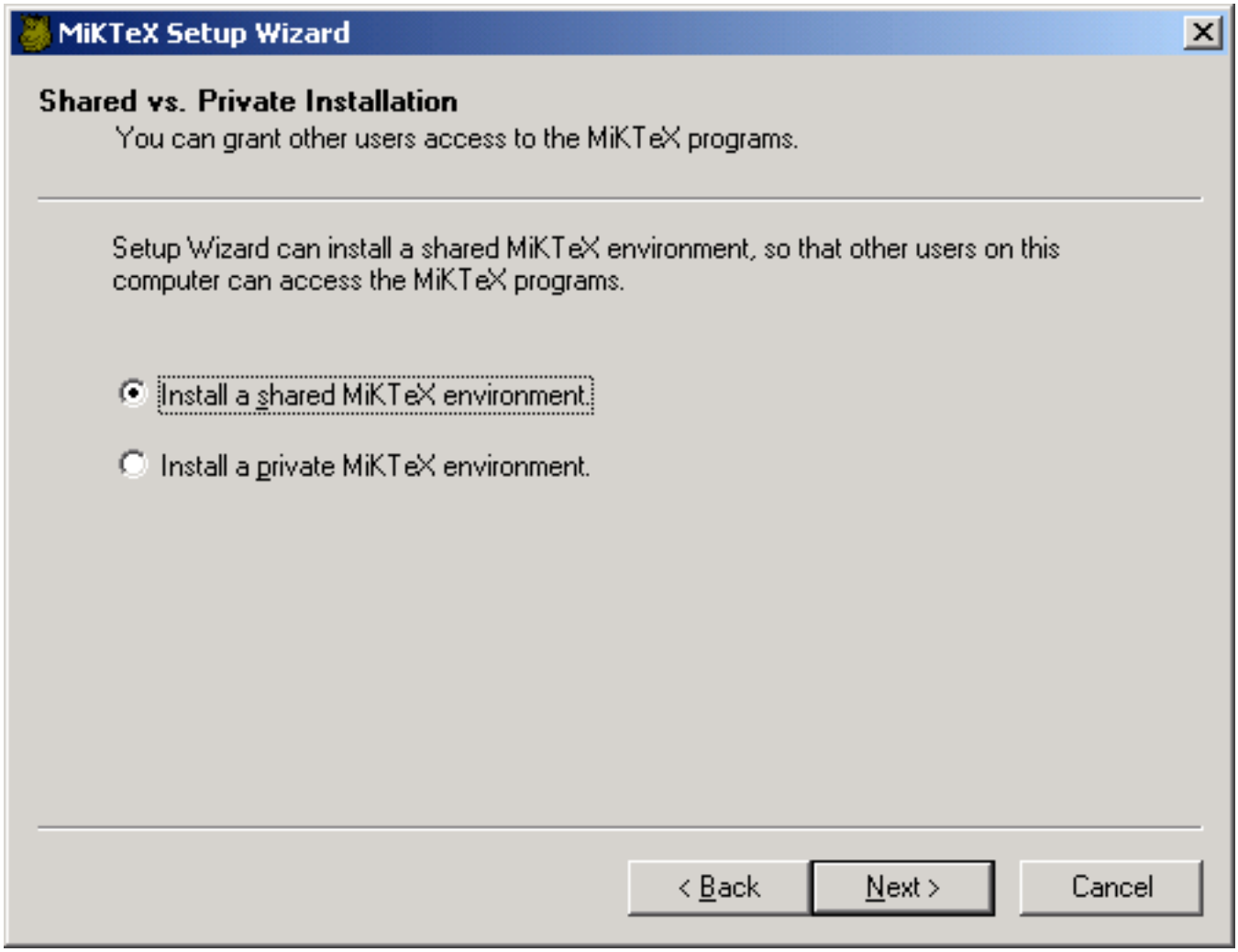
3 Installing MiKTEX

3.1 Running Setup Wizard

1. You should first remove any previous MiKTEX installation:
   - MiKTEX 1.11 users can use the uninstall program.
   - MiKTEX 1.20 users must remove MiKTEX by hand. See the Local Guide of version 1.20, for more information.
   - MiKTEX 2.0 users can use the uninstall program.
2. Make sure that you have enough disk space. A complete MiKTEX installation consumes approximately 70MB of disk space.
3. It is highly recommended that you login as Administrator, if you're installing MiKTEX on a Windows NT/2000 computer.
4. Choose a location for the installation folder (e.g. `C:\Program Files\MiKTeX`). This folder receives the files of the MiKTEX distribution.
5. You can cause MiKTEX to deposit newly created files (fonts, format files, file name databases) in a separate hierarchy of folders, called the *Local TEXMF tree*. If you decide to use a local TEXMF tree, then you must choose a location for its root folder (e.g. `C:\Local TeXMF`).

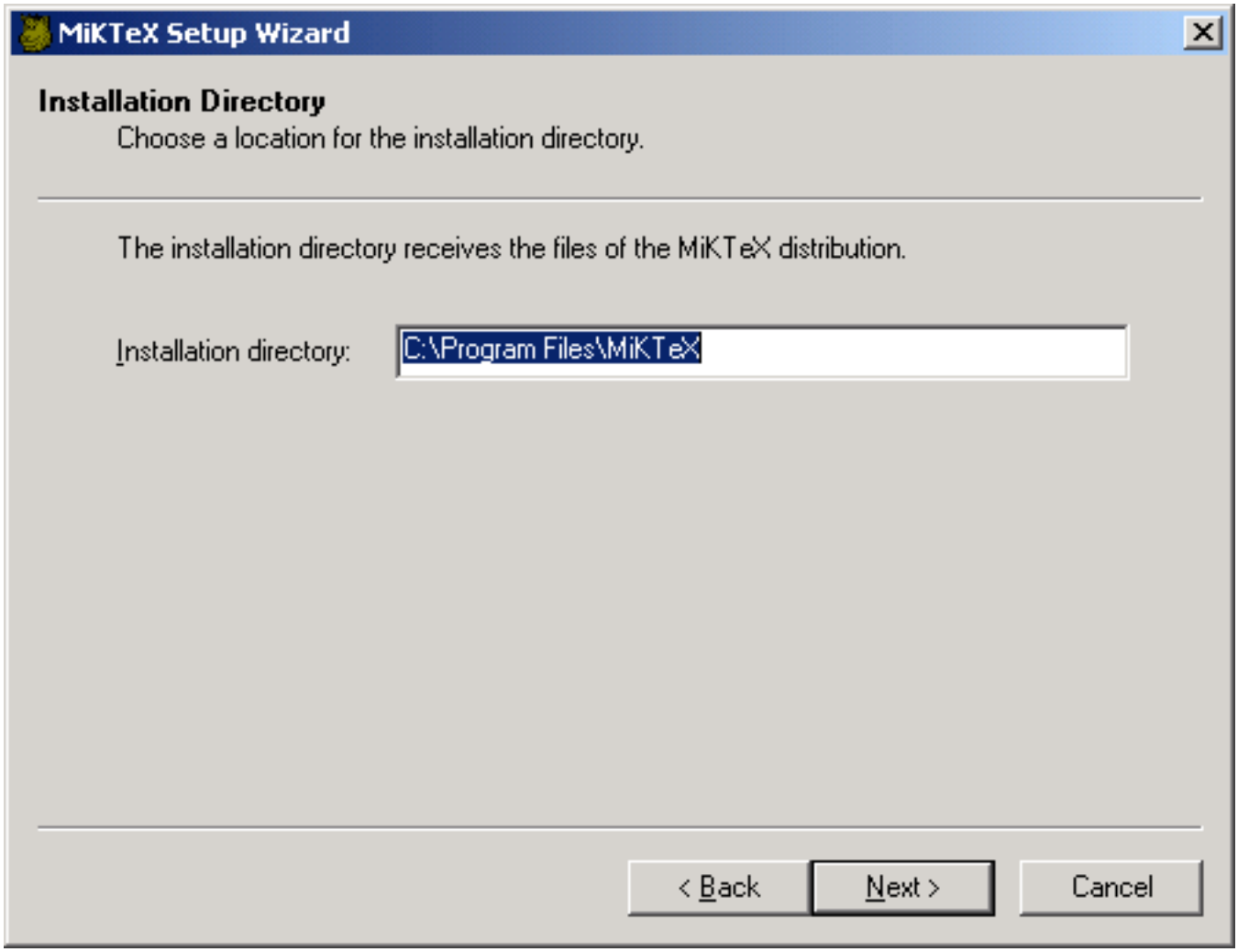
   Benefits that a local TEXMF tree provides include the following:
   - You can use it for your own additions (macros, fonts).
   - Easier updates: You don't have to worry about future MiKTEX updates, since the local TEXMF tree wont be touched by the setup program.
6. Decide whether you want to incorporate an existing TEXMF tree. For example, if you have a TEXLive CD inserted in your CD-ROM drive `E:`, then it would be possible to include `E:\texmf` (the root of the TEXLive TEXMF tree) in the MiKTEX search procedure.
7. Start MiKTEX Setup Wizard (`setupwiz.exe`). You will presented with the welcome page:

8. Click '`Next>`' to go to the next page:



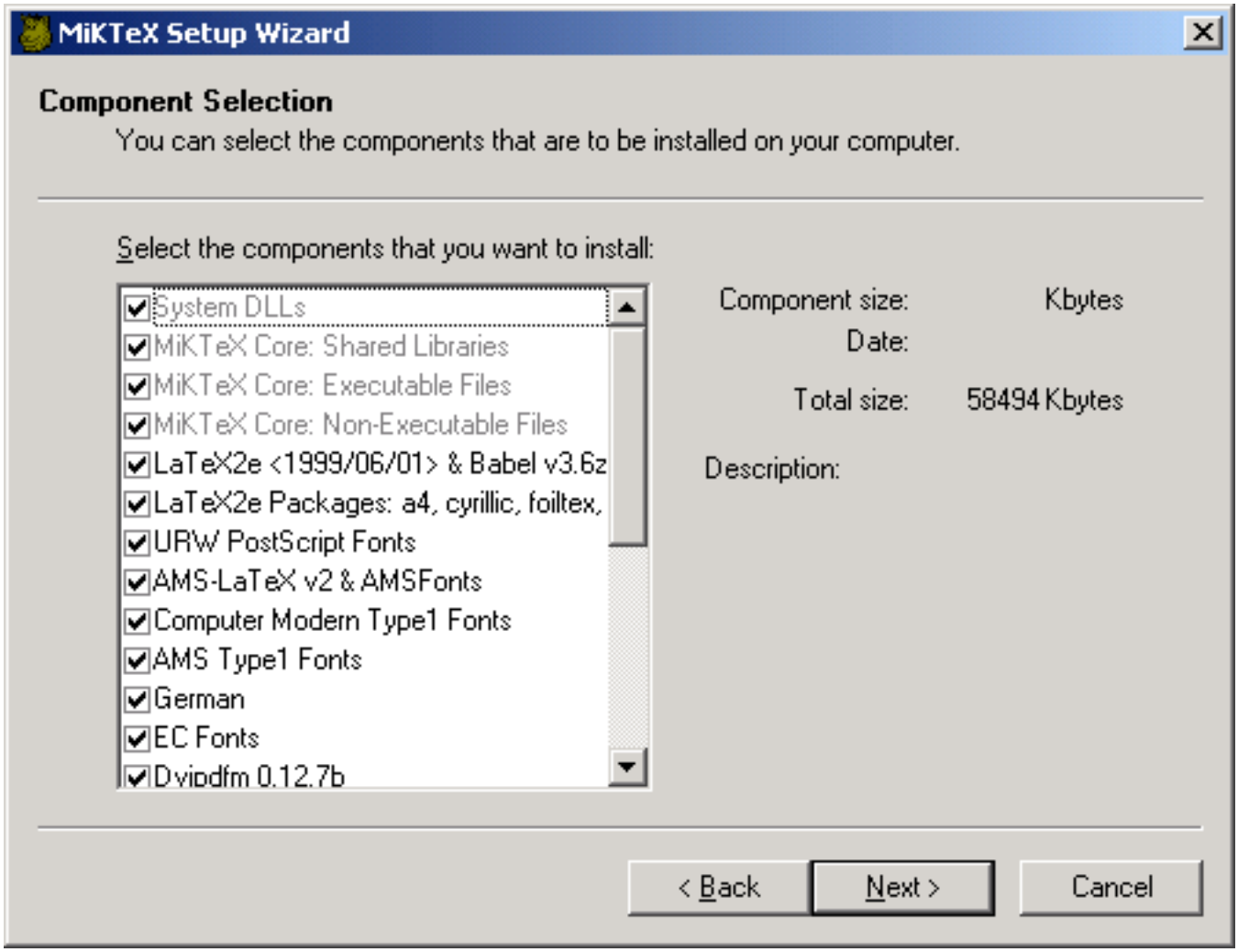
You have the option to create a shared installation, so that other users are able to use MiKTEX. Under Windows NT/2000, this option requires administrator privileges.

9. Click '`Next>`' to go to the next page:



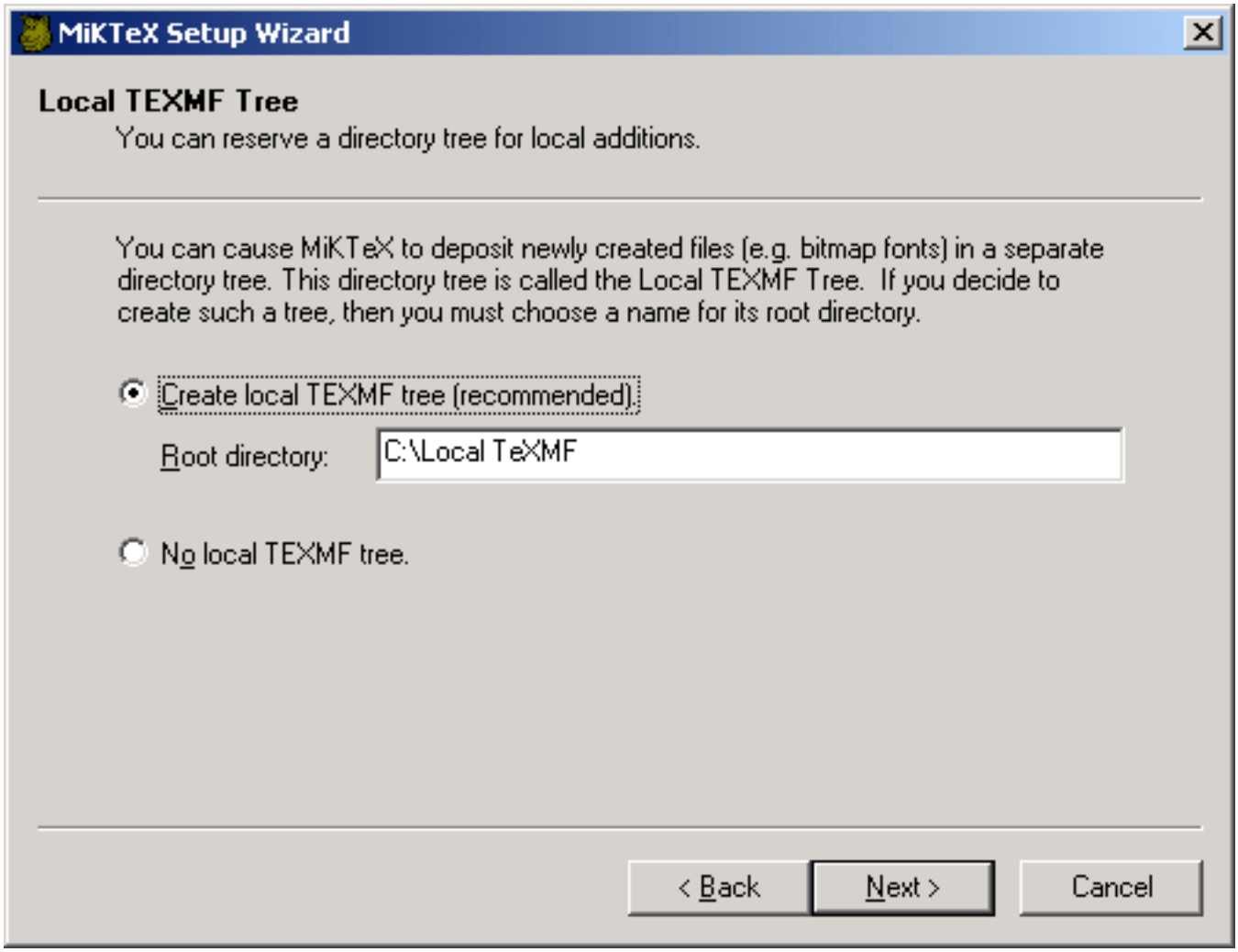
Enter the full path to the desired installation folder (see step 4).

10. Click '`Next>`' to go to the next page:



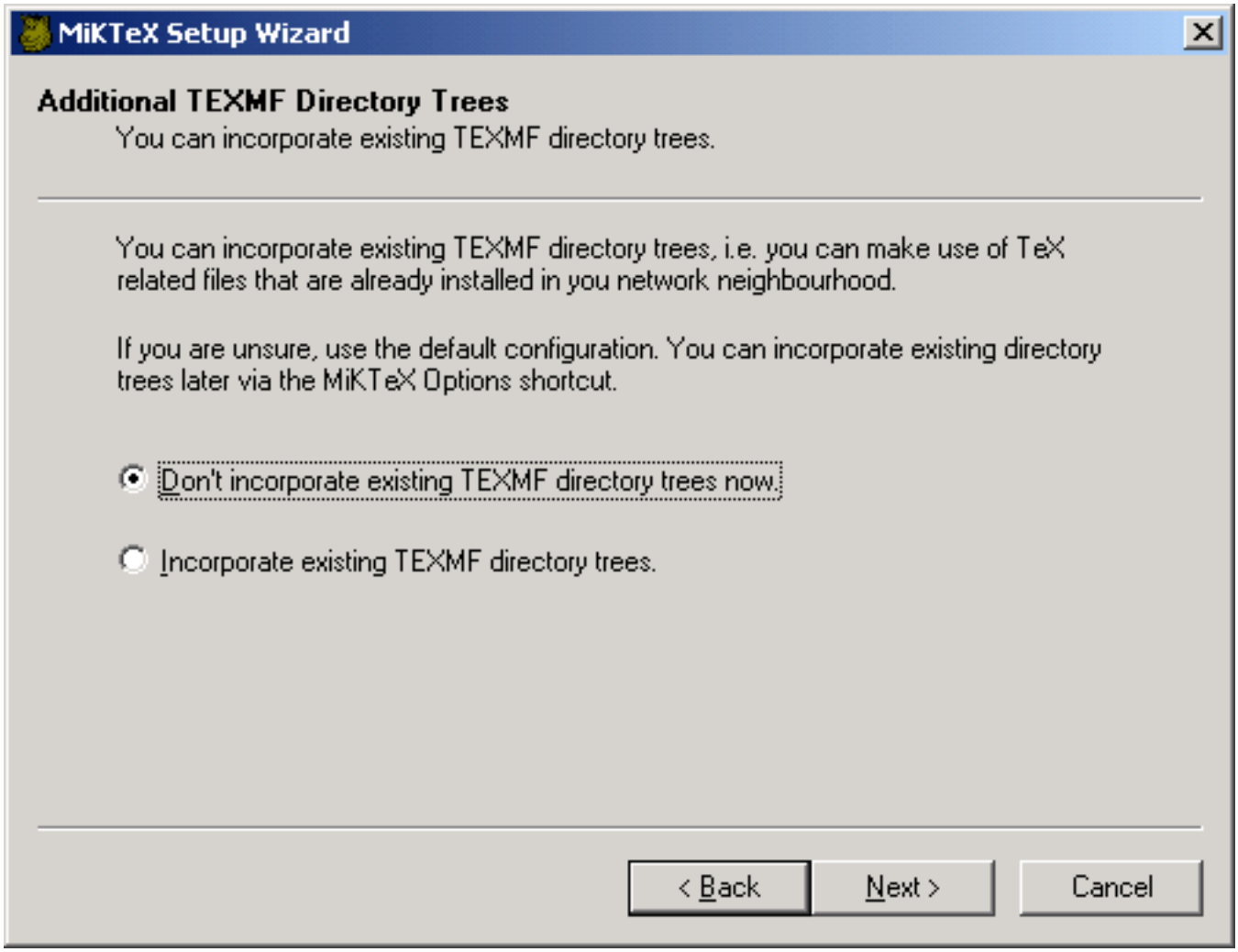
Choose the components that you want to install.

11. Click '`Next>`' to go to the next page:



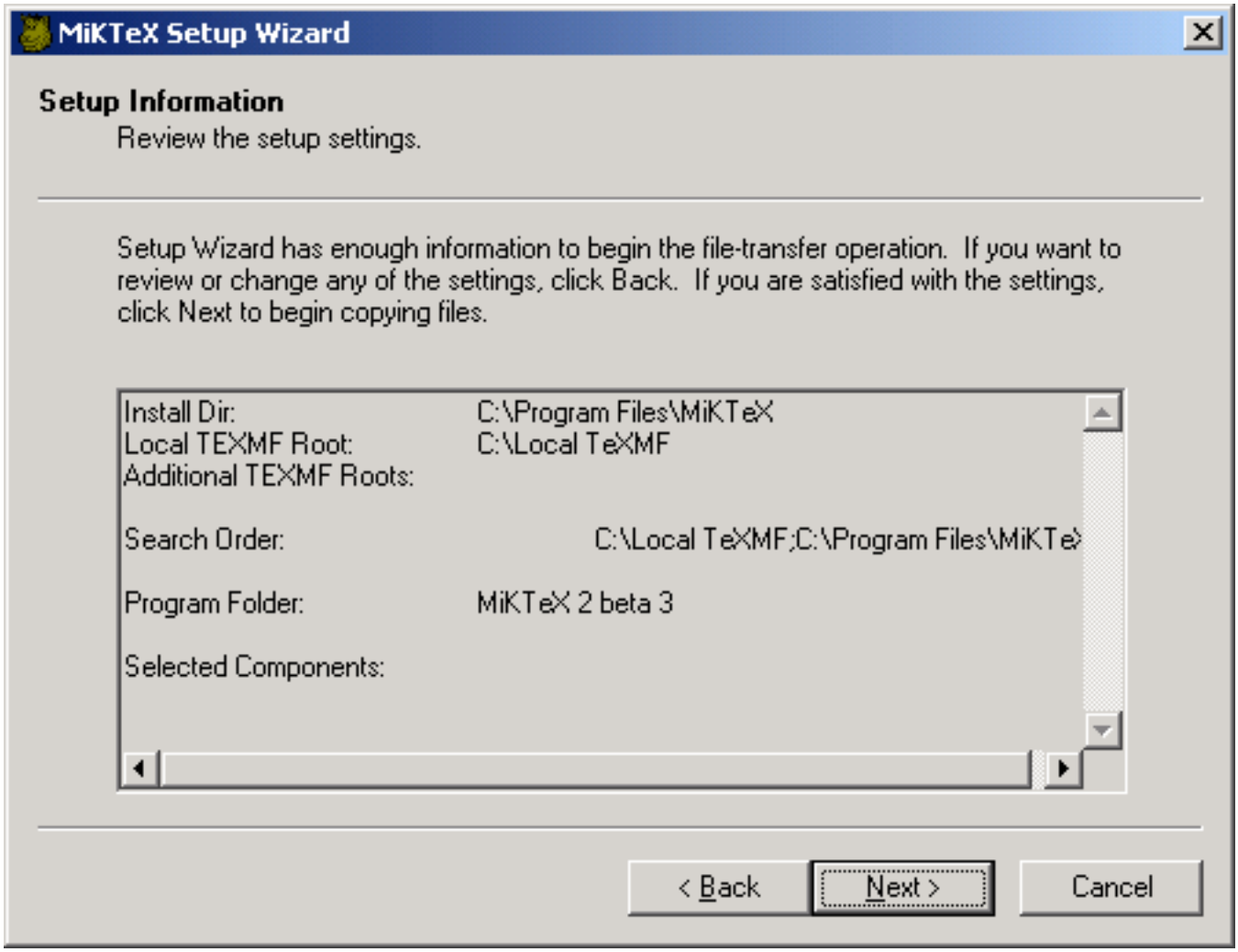
Enter the full path to the desired local TEXMF folder (see step 5). Check the button '`No local TEXMF tree`', if you don't want to use a local TEXMF tree.

12. Click '`Next>`' to go to the next page:



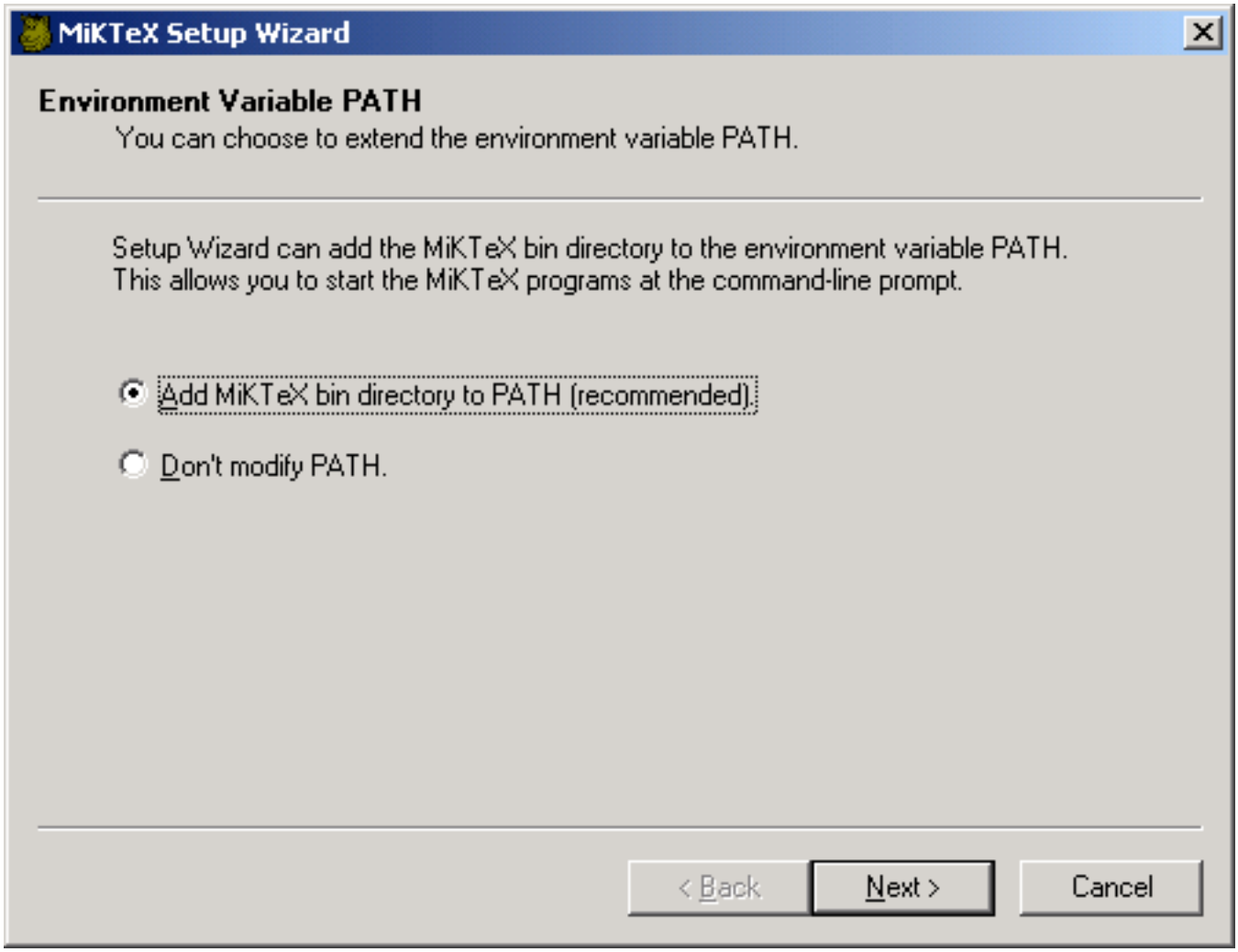
You are asked about whether you want to incorporate existing TEXMF trees. If you check the button '`Incorporate existing TEXMF directory trees`', then you must specify the locations of those trees on a subsequent page.

13. Click '`Next>`' to go to the next page:

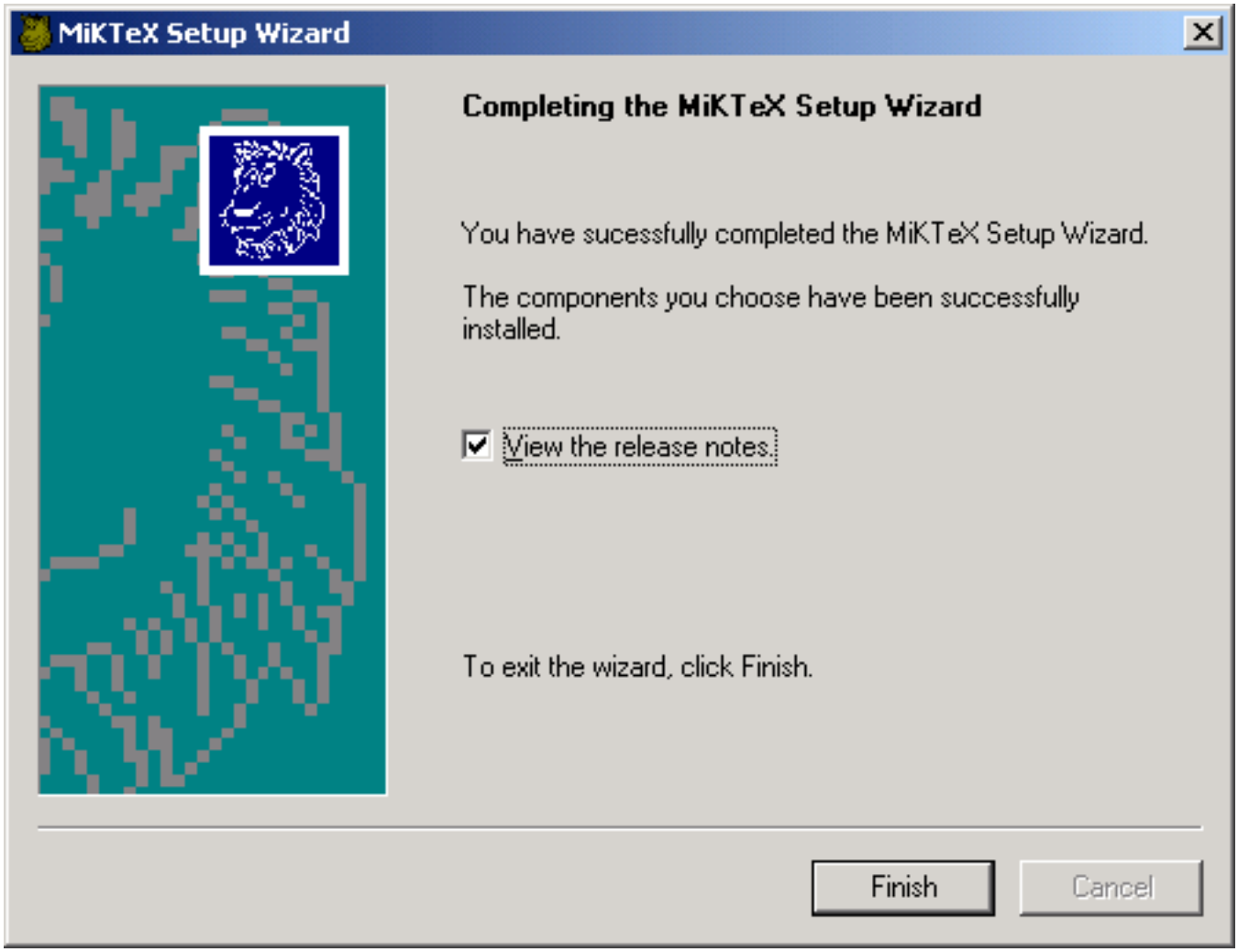


You should now review the installation options.

14. Click '`Next>`' to start the installation process. The selected components will be copied to the installation folder chosen in step 9.
15. When the files have been copied to your computer, you have the option to register the path to the MiKTEX bin directory:



16. Click '`Next>`' to go to the final page:



Click '`Finish`' to close Setup Wizard.

3.1.1 Setup Options

For the purpose of an unattended setup, some settings can be specified on the command-line and/or in a separate text file named `setupwiz.opt`.

You can invoke the MiKTEX Setup Wizard with the following command-line options:

`--additional-directory-trees` *DIRS*
: Specify additional TEXMF directories.

`--allow-unattended-reboot`
: Allow a reboot in unattended mode.

`--dry-run`
: Simulate the installation process. No files will be installed. The log file will be written to the temporary directory.

`--installation-directory` *DIR*
: Specify the installation directory.

`--help` Show available options and exit.

`--no-additional-directory-trees`
: Prevent MiKTEX from using additional directory trees.

`--no-local-directory`
: Prevent MiKTEX from using a local directory.

`--program-folder` *FOLDER*
: Specify the MiKTEX program folder.

`--unattended`
: Run Setup Wizard in unattended mode. No user input is required.

Command-line options can also be specified in a text file named `setupwiz.opt`. This file must be in the same directory as `setupwiz.exe`.

3.2 Items in the Start Menu

The Setup Wizard inserts the following menu items into the start menu:

Shortcuts to help files

`MiKTeX 2.0 | Help | Frequently Asked Questions`
: Answers to frequently asked questions.

`MiKTeX 2.0 | Help | LaTeX2e Reference`
: LaTEX2e reference manual.

`MiKTeX 2.0 | Help | Local Guide`
: The MiKTEX Manual.

`MiKTeX 2.0 | Help | Release Notes`
: Last-minute release notes.

`MiKTeX 2.0 | Help | Tips and Tricks`
: Useful tips.

Program links

`MiKTeX 2.0 | DVI viewer`
: A shortcut to the DVI viewer.

`MiKTeX 2.0 | MiKTeX Options`
: A shortcut to the configuration utility.

3.3 The TEXMF Folder Hierarchy

Setup Wizard creates a TDS-compliant folder hierarchy:

1. The *installation folder* (usually '`C:\Program Files\MiKTeX`') contains the subfolder and files of the MiKTEX distribution. The contents of the installation folder is meant to be read-only, i.e. no files should be added, removed or changed.

2. The *local TEXMF folder* (usually '`C:\Local TeXMF`') receives all files that are created on-the-fly by certain utilities.

You can incorporate additional TEXMF folders, if the need arises. For example, you could create a TDS-compliant folder hierarchy rooted at `C:\My TeXMF Files`. This would serve as a repository for your own TEX files. See Section 4.1 [Defining TEXMF Root Directories], page 19, for more information.

3.3.1 Installation Folder

The installation folder (usually `C:\Program Files\MiKTeX`) is the root of a TDS-compliant folder hierarchy. If you have installed the complete distribution, then the installation folder contains the following subfolders:

`bibtex`, `dvips`, `makeindex`, ...
: These subfolders contain application related input files.

`doc`
: This subfolder contains all user documentation.

`fonts`
: This subfolder contains fonts in various formats.

`miktex`
: The `miktex` subfolder is reserved for MiKTEX related files:

  `miktex\bin`
  : Contains all executables.

  `miktex\config`
  : Contains the global configuration file `miktex.ini` and the MiKTEX font mapping file `miktex.map`. The MiKTEX Setup Wizard deposits its own log file in this subfolder.

  `miktex\base`
  : Contains the METAFONT string pool file `mf.pool`.

  `miktex\fmt`
  : Contains TEX string pool files: `etex.pool`, `pdftex.pool`, `omega.pool`, `tex.pool`.

  `miktex\mem`
  : Contains the MetaPost string pool file `mp.pool`.

3.3.2 Local TEXMF Folder

The local TEXMF folder (usually `C:\Local TeXMF`) receives files that are generated on-the-fly. For example, if the TEX compiler needs a TEX Font Metric (TFM) file that is not available yet, then it creates that file (if possible) and installs it in an appropriate subfolder of the local TEXMF folder.

Typically, the local TEXMF folder contains the following subfolders:

`fonts`
: Contains font files that are not part of the MiKTEX distribution, but that were created on-the-fly.

`miktex\config`
: This subfolder contains the file name database files.

`miktex\base`
: This subfolder contains METAFONT format files.

`miktex\fmt`
: This subfolder contains TEX format files.

`miktex\mem`
: This subfolder contains MetaPost format files.

3.4 Removing MiKTEX

MiKTEX can be removed with the help of the *Remove MiKTeX! Wizard*:

1. Open Control Panel (usually via `Start | Settings | Control Panel`).
2. Click on the `Software` (or `Add/Remve Programs`) icon.
3. Select `MiKTeX 2.0`.

4 Configuring MiKTeX

4.1 Managing TEXMF Folder Hierarchies

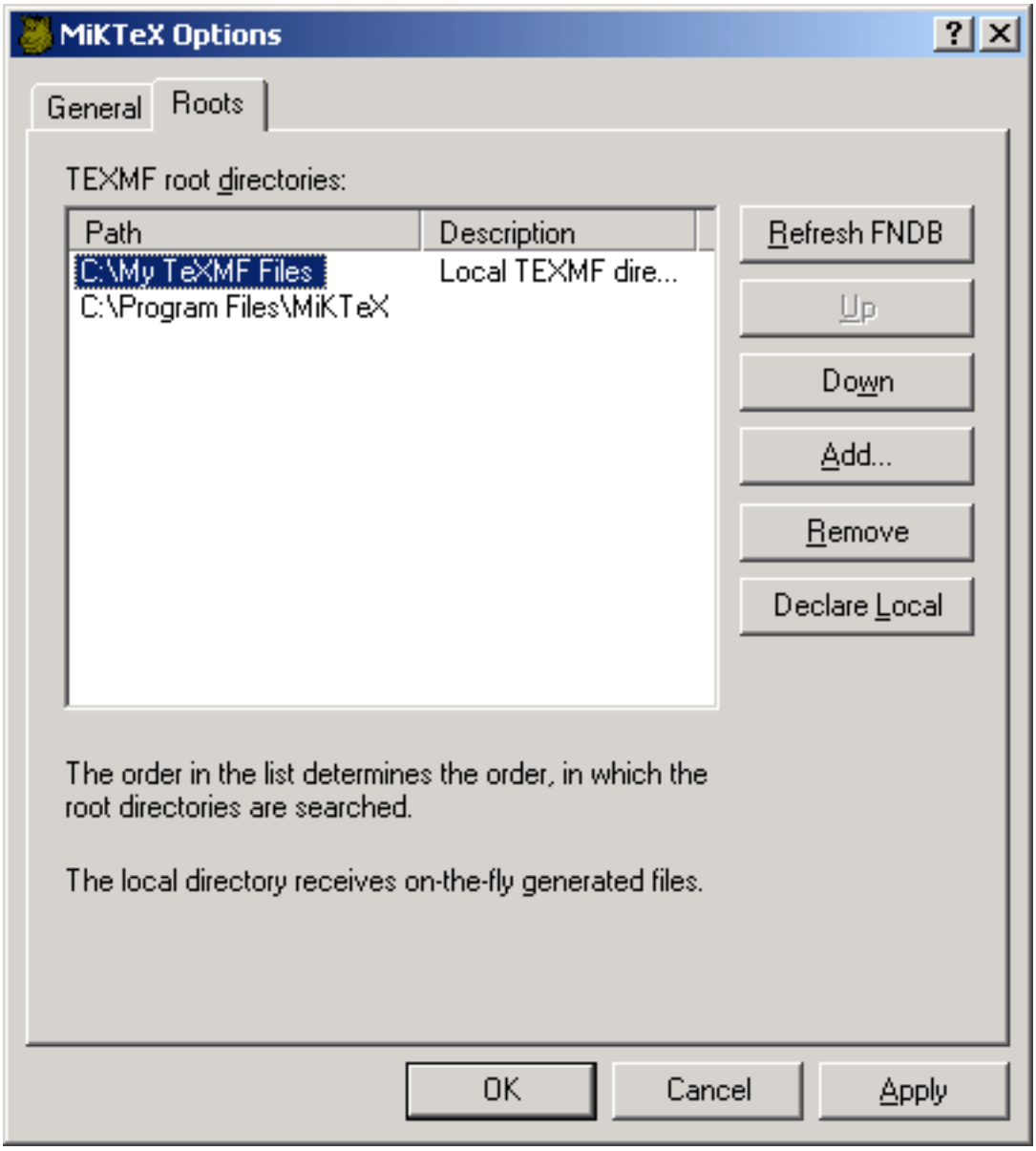
The standard setup process creates two TEXMF root folders:

- `C:\Program Files\MiKTeX`: the installation folder (see Section 3.3.1 [Installation Folder], page 17).
- `C:\Local TeXMF`: the local TEXMF folder (see Section 3.3.2 [Local TEXMF Folder], page 17).

You can incorporate additional TEXMF folder hierarchies with the help of *MiKTeX Options*:

1. Click `Start | Programs | MiKTeX 2.0 | MiKTeX Options`.
2. The *MiKTeX Options* window opens. Click on the `Roots` tab.

The `Roots` window displays a list of TEXMF root folders that are used by MiKTeX:



Click `Up` and `Down` to change the order by which the TEXMF folders are searched for input files.

Click `Add...` to append a new TEXMF root folder to the list.

Click `Refresh FNDB` to scan the selected folders for new files and update the file name database (see Section 4.2 [FNDB Refresh], page 20) accordingly.

Click `Remove` to remove selected folders from the list.

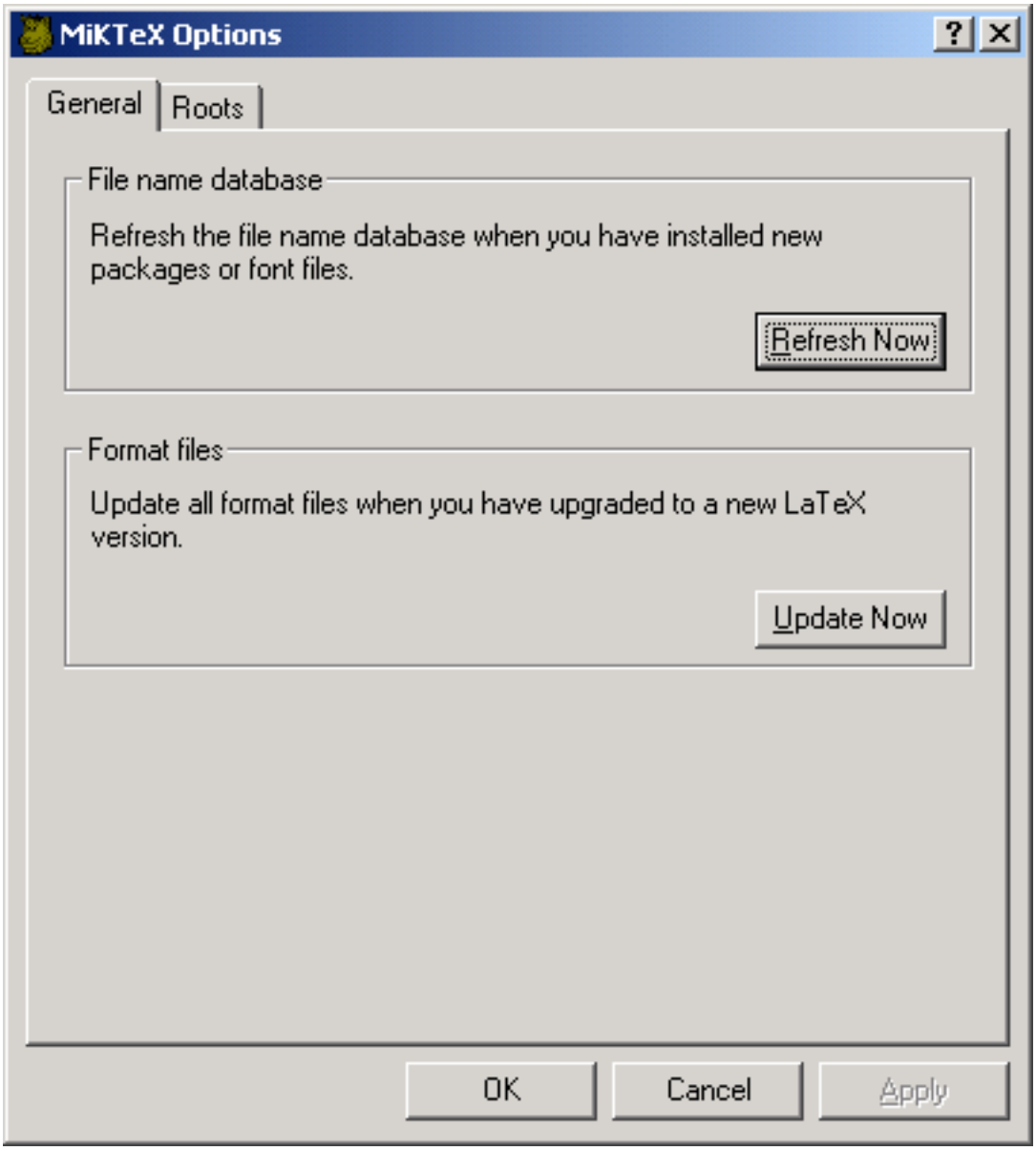
Click `Declare Local` to declare the selected folder as the local TEXMF folder (see Section 3.3.2 [Local TEXMF Folder], page 17).

4.2 Refreshing the File Name Database

To speed up file search, MiKTEX makes use of a list of known file names. This list is called the file name database (FNDB).

It is strongly recommended that you update the file name database whenever files are added to or removed from one of the TEXMF directories. You can update the file name database with the help of *MiKTeX Options*:

1. Click `Start | Programs | MiKTeX 2.0 | MiKTeX Options`.
2. The *MiKTeX Options* window opens:



3. Click `Refresh Now`

4.3 Managing Format Files

Some programs initialize itself by reading parts of the memory from an external file. For the TEX family of programs, such a file is called a *Format File*.

4.3.1 Building Format Files

MiKTEX Setup Wizard has created standard format files in course of the installation process. You can refresh those format files with the help of *MiKTeX Options*:

1. Click `Start | Programs | MiKTeX 2.0 | MiKTeX Options`.
2. The *MiKTeX Options* window opens.
3. Click `Update Now`.

4.3.2 Defining New Format Files

1. Click `Start | Programs | MiKTeX 2.0 | MiKTeX Options`.
2. The *MiKTeX Options* window opens.
3. Click on the `TeX Formats` tab.

You are presented with a list of well known TEX formats. To add an entry to this list, click `New...`. You have to supply the following information:

Format Name
: The name of the format.

Compiler The TEX compiler variant which must be used to create the format file.

Input File The name of the main input file.

Output File
: The name of the output file.

Preloaded Format
: Optional: The name of another format, which must be loaded before creating the actual format file.

Description
: A one-line comment which describes the new format.

A new executable with the the name of the format will be created.

4.4 Selecting Hyphenation Rules

You can control the loading of hyphenation patterns with the help of MiKTEX Options:

1. Click `Start | Programs | MiKTeX 2.0 | MiKTeX Options`.
2. The *MiKTeX Options* window opens.
3. Click on the `Languages` tab.

You can choose which hyphenation patterns are to be loaded by TEX.

4.5 Expert Topics

4.5.1 Maintaining the PostScript resource database

The PostScript resource database (PSres) is used by some utilities in order to locate PostScript resources (font outlines/metrics/encodings).

The database is located in the local MiKTEX config folder (usually '`C:\Local TeXMF\miktex\config`'). The name of the database file is '`dpres.dpr`'. It is a text file, i.e. you can view it with a conventional text editor.

It is strongly recommended that you update the database whenever PostScript resources ('`*.pfb;*.afm;*.enc`') are added to or removed from one of the TEXMF trees.

You update the database files by invoking `initexmf` with the command line switch `--mkpsres`:

```
C:\> initexmf --mkpsres
```

4.5.1.1 Incorporating External Font Directories

It is possible to add non-MiKTEX font directories to the PostScript resource database. The `--mkpsres` switch accepts as an optional argument the name of an external font directory. You can use several `--mkpsres` switches with one invocation of `initexmf`.

By specifying the command line flag `--search`, you can cause `initexmf` to automatically search your workstation for third party PostScript resource files (e.g. Acrobat Reader fonts):

```
C:\> initexmf --mkpsres --search
```

5 Non-standard TEX Features

This chapter describes features, that were added to the MiKTEX implementation of Donald Knuth's TEX.

5.1 Suppressing Screen Output

The command-line option '`--quiet`' suppresses all diagnostic messages. You will see no screen output, unless there are errors. These are shown in a "C style form" and do not stop the compilation process. For example, the input file

foo.tex:
```
\documentclass{article}
\begin{document}
What's \This?
\end{documnt}
```

would cause TEX to print one error message, as in

```
C:\> latex --quiet foo.tex
foo.tex:3: Undefined control sequence
C:\>
```

5.2 Auto-insertion of Source Specials

What are source specials?

Source specials are pieces of information embedded in a DVI file, which make a connection between the source file location (e.g. line 100 in `foo.tex`) and the DVI location (e.g. page 2 in `foo.dvi`). Source specials can improve the Edit-TEX-View-Edit cycle:

1. You edit your source file.
2. You compile the source file to get a DVI file.
3. You execute a special editor command to open Yap, going directly to the DVI page that corresponds to the cursor location in your source file.
4. You navigate inside the DVI file (e.g. PgUp/PgDn).
5. You double-click somewhere on the DVI view; this causes Yap to bring the editor window to the front, moving the text cursor directly to the line that corresponds to the view location.

How to insert source specials

The TEX compiler option '`--src`' inserts source specials into the DVI file. You would say

```
C:\> latex --src foo.tex
```

to create the DVI file `foo.dvi` with embedded source specials.

5.3 Quoted File Names

The TEX compiler can handle quoted file names. This makes it possible to specify long file names that contain spaces.

For example, to compile the input file '`long file name.tex`', you start TEX as follows:

```
C:\> latex "long file name"
```

This produces the DVI file '`long file name.dvi`'. The log file is named '`long file name.log`'.

You can, to some extent, use quoted file names inside the TEX document. For example,

```
\input{"extra long file name"}
```

would cause TEX to read the file '`extra long file name.tex`'.

Things get a little bit complicated if you want to use the LaTEX primitive '`\include`'. You have to write

```
\include{"extra\space long\space file\space name"}
```

in order to get the expected result.

5.4 Specifying Additional Input Directories

The command-line option '`--include-directory=`*dir*' allows you to extend the input search path for one invocation of TEX.

For example,

```
tex --include-directory="C:\My TeX" foo.tex
```

prepends '`C:\My TeX`' to the input search path, i.e. '`C:\My TeX`' will be searched first, when TEX opens any input file (including `foo.tex`).

You can specify either absolute paths (as in the example above) or relative paths.

5.5 Specifying the Output Directory

The command-line option '`--output-directory=`*dir*' causes TEX to create all output files in another directory.

Example:

```
mkdir C:\texoutput
latex --output-directory=C:\texoutput foo.tex
```

All output files (`foo.dvi`, `foo.log`, ...) will be created in `C:\texoutput`.

5.6 Running Programs

TEX handles output stream 18 in a special way: the token list is interpreted as a system command. If the `\write18` feature is enabled (see below), then `\write18{`*toklist*`}` starts the command interpreter (usually `command.com`) to carry out the command specified by *toklist*. For example:

```
\write{dir}
```

lists the files and subdirectories of the current directory.

TEX ignores `\write18` by default. You enable it by editing `miktex.ini` (search for `write18`) or by using the TEX compiler switch `--enable-write18`.

5.7 TCX files: Character translations

[This section is "borrowed" from the Web2C manual].

TCX (TEX character translation) files help TEX support direct input of 8-bit international characters if fonts containing those characters are being used. Specifically, they map an input (keyboard) character code to the internal TEX character code (a superset of ASCII).

Of the various proposals for handling more than one input encoding, TCX files were chosen because they follow Knuth's original ideas for the use of the '`xchr`' and '`xord`' tables. He ventured that these would be changed in the WEB source in order to adjust the actual version to a given environment. It turned out, however, that recompiling the WEB sources is not as simple task as Knuth predicted; therefore, TCX files, providing the possibility of changing of the conversion tables on on-the-fly, has been implemented instead.

This approach limits the portability of TEX documents, as some implementations do not support it (or use a different method for input-internal reencoding). It may also be problematic to determine the encoding to use for a TEX document of unknown provenance; in the worst case, failure to do so correctly may result in subtle errors in the typeset output.

While TCX files can be used with any format, using them breaks the LaTEX '`inputenc`' package. This is why you should either use *tcxfile* or '`inputenc`' in LaTEX files, but never both.

Specifying TCX files:

- You can specify a TCX file to be used for a particular TEX run by specifying the command-line option '`-translate-file=`*tcxfile*' or (preferably) specifying it explicitly in the first line of the main document '`%& -translate-file=`*tcxfile*'.
- TCX files are searched for along the `TCXPath` path.
- `INITEX` ignores TCX files.

The MiKTEX distribution comes with at least two TCX files, '`il1-t1.tcx`' and '`il2-t1.tcx`'. These support ISO Latin 1 and ISO Latin 2, respectively, with Cork-encoded fonts (a.k.a. the T1 encoding). TCX files for Czech, Polish, and Slovak are also provided.

Syntax of TCX files:

1. Line-oriented. Blank lines are ignored.
2. Whitespace is ignored except as a separator.
3. Comments start with '`%`' and continue to the end of the line.
4. Otherwise, a line consists of one or two character codes:

   src `[`*dest*`]`
5. Each character code may be specified in octal with a leading '`0`', hexadecimal with a leading '`0x`', or decimal otherwise. Values must be between 0 and 255, inclusive (decimal).
6. If the *dest* code is not specified, it is taken to be the same as *src*.
7. If the same *src* code is specified more than once, it is the last definition that counts.

Finally, here's what happens: when TEX sees an input character with code *src*, it 1) changes *src* to *dest*; and 2) makes code the *dest* "printable", i.e., printed as-is in diagnostics and the log file instead of in '`^^`' notation.

By default, no characters are translated, and character codes between 32 and 126 inclusive (decimal) are printable. It is not possible to make these (or any) characters unprintable.

Specifying translations for the printable ASCII characters (codes 32–127) will yield unpredictable results. Additionally you shouldn't make the following characters printable: `^^I` (TAB), `^^J` (line feed), `^^M` (carriage return), and `^^?` (delete), since TEX uses them in various ways.

Thus, the idea is to specify the input (keyboard) character code for *src*, and the output (font) character code for *dest*.

6 TEXify: The MiKTEX Compiler Driver

`texify` is a command-line utility that simplifies the creation of DVI (PDF) documents: `texify` automatically runs LaTEX (pdfLaTEX), MakeIndex and BibTEX as many times as necessary to produce a DVI (PDF) file with sorted indices and all cross-references resolved.

To run `texify` on an input file '`foo.tex`', do this:

```
C:\> texify foo.tex
```

As shown in this example, the input file names to `texify` must include any extension ('`.tex`', '`.ltx`', etc.).

There are several command-line optione you can use to control `texify` (see Section A.9 [texify], page 38). Here are some examples:

'`texify --clean foo.tex`'
: All auxiliary files will be removed, i.e. only the output '`foo.dvi`' file will be left in the current directory.

'`texify --tex-opt=--src foo.tex`'
: Passes the option '`--src`' to the TEX compiler.

'`texify --run-viewer foo.tex`'
: Opens the output file '`foo.dvi`' (unless there are compile erros).

'`texify --tex-opt=--src --viewer-opt="-1 -s\"200 foo.tex\"" --run-viewer foo.tex`'
: Compiles '`foo.tex`' with source file information ('`--src`') and then initiates forward DVI search to open '`foo.dvi`' at the source special location '`200 foo.tex`'. The viewer option '`-1`' activates the current viewer task (if there is already one running).

Appendix A Manual Pages

This chapter includes manual pages for some few programs. Only programs with non-standard command-line options are documented here. You should browse the document folder (usually `C:\Program Files\MiKTeX\doc`) if you are searching for 'real' documentation.

A.1 Common Compiler Options

The following command-line switches are commonly supported by all variants of the TEX compiler.

`--alias=`*app*
: Pretend to be *app*.

`--buf-size=`*n*
: Set the internal `buf_size` to *n*. `buf_size` is the maximum number of characters simultaneously present in current lines of open files and in control sequences between `\csname` and `\endcsname`; must not exceed 1073741823.

`--c-style-errors`
: Show C/C++ style error messages. This switch implies `\scrollmode`.

`--error-line=`*n*
: Set the internal `error_line` to *n*. `error_line` us the width of context lines on terminal error messages.

`--half-error-line=`*n*
: Set the internal `half_error_line` to *n*. `half_error_line` is the width of first lines of contexts in terminal error messages; should be between 30 and (`error_line` - 15).

`--halt-on-error`
: Quit after the first error.

`--initialize`
: Initialize internal tables; these tables can be `\dump`ed to a dump file.

`--include-directory=`*dir*
: Prepend *dir* to the search path.

`--interaction=`*mode*
: Set TEX's interaction mode (one of: `batchmode`, `nonstopmode`, `scrollmode`, `errorstopmode`).

`--job-name=`*name*
: Specify the name of the job. This also sets the name of all output files.

`--job-time=`*filename*
: Set the time of all output files to the time of *filename*.

`--help` Show a help screen and exit.

`--max-in-open=`*n*
: Set the internal `max_in_open` to *n*. `max_in_open` is the maximum number of input files and error insertions that can be going on simultaneously.

`--max-print-line=`*n*
: Set the internal `max-print-line` to *n*. `max-print-line` is the width of longest text lines output; should be at least 60.

`--max-strings=`*n*
: Set the internal `max_strings` to *n*. `max_strings` is the maximum number of strings; must not exceed 1073741823.

`--mem-bot=`*n*
: Set the internal `mem_bot` to *n*. `mem_bot` is the smallest index in the `code` array dumped by `INITEX` (`INIOMEGA`, `INIPDFTEX`); must not be less than `mem_min`.

`--mem-max=`*n*
: Set the internal `mem_max` to *n*. `mem_max` is the greatest index in the internal `mem` array; must be strictly less than 1073741823.

`--mem-min=`*n*
: Set the internal `mem_min` to *n*. `mem_min` is the smallest index in the internal `mem` array; must be 0 or more; must be equal to `mem_bot` in `INITEX` (`INIOMEGA`, `INIPDFTEX`), otherwise <= `mem_bot`.

`--mem-top=`*n*
: Set the internal `mem_top` to *n*. `mem_top` is the largest index in the `mem` array dumped by `INITEX` (`INIOMEGA`, `INIPDFTEX`); must be substantially larger than 0 and not greater than `mem_max`.

`--nest-size=`*n*
: Set the internal `nest_size` to *n*. `nest_size` is the maximum number of semantic levels simultaneously active.

`--output-directory=`*dir*
: Sets the output directory.

`--param-size=`*n*
: Set the internal `param_size` to *n*. `param_size` is the maximum number of simultaneous macro parameters.

`--pool-size=`*n*
: Set the internal `pool-size` to *n*. `pool_size` is the maximum number of characters in strings, including all error messages and help texts, and the names of all fonts and control sequences; must exceed `string_vacancies` by the total length of the program's own strings, which is currently about 30000.

`--quiet`
: Suppress all output (except errors).

`--save-size=`*n*
: Set the internal `save_size` to *n*. `save_size` is the amount of space for saving values outside of current group; must be at most 1073741823.

`--shell-escape`
: Same as `--enable-write18`.

`--silent` Same as `--quiet`.

`--src-specials`
: Insert source file information into the DVI file.

`--stack-size=`*n*
: Set the internal `stack_size` to *n*. `stack_size` is the maximum number of simultaneous input sources.

`--string-vacancies=`*n*
: Set the internal `string_vacancies` to *n*. `string_vacancies` is the minimum number of characters that should be available for the user's control sequences and font names, after the program's own error messages are stored.

`--tcx=`*name*
`--translate-file=`*name*
: Causes TEX to process the TCX table *name*.

`--terminal=oem`
: Causes TEX to use the current DOS codepage (e.g. cp850) for console output.

`--trace=`*traceflags*
: Set trace flags.

`--trie-size=`*n*
: Set the internal `trie_size` to *n*. `trie_size` is the amount of space for hyphenation patterns; should be larger for `INITEX` (`INIOMEGA`, `INIPDFTEX`) than it is in production versions of the program.

`--trie-op-size=`*n*
: Set the internal `trie_op_size` to *n*. `trie_op_size` is the amount of space for "opcodes" in the hyphenation patterns.

`--try-gz` Try *file*`.tex.gz` if *file*`.tex` cannot be found.

`--undump=`*name*
: Causes TEX to read the format file *name*.

`--version`
: Print version information and exit.

`--enable-write18`
: Enable the `\\write18` construct (see Section 5.6 [write18], page 26).

A.2 bibtex

BibTEX is a preprocessor for the LaTEX document-preparation system. It handles most of the formatting decisions required to produce a reference list, outputting a `.bbl` file; with this file LaTEX actually produces the reference list.

Synopsis

`bibtex [`*option*`...] `*name*

Reads the file '*name*`.aux`' and outputs the file '*name*`.bbl`'.

Options

`--help` Shows a help screen and exits successfully.

`--min-crossrefs=`*N*
Sets the internal `min_crossrefs` parameter to *N*.

`--version`
Shows version information and exits successfully.

Documentation

See *BibTEXing*, available as file '`btxdoc.dvi`'.

A.3 `dvicopy`

`dvicopy` is a utility program that allows one to take a DVI file that references composite fonts (VF) and convert it into a DVI file that does not contain such references.

Synopsis

`dvicopy [`*option*`...] `*old new*

Converts DVI file *old* into *new*.

Options

`--help` Shows a help screen and exits successfully.

`--mag=`*MAG*
Sets magnification to *MAG*.

`--select=`*range*
Selects a range of pages to be copied.

`--version`
Prints version information and exits successfully.

A.4 `initexmf` (MiKTEX Configuration Utility)

`initexmf` is the MiKTEX Configuration Utility.

Synopsis

`initexmf [`*option*`...]`

Options

`--dump` Refresh all format files (`*.base`;`*.efmt`;`*.fmt`;`*.mem`).

`--dump=`*program*
: Refresh the format files related to a specific program. *program* must be one of: `elatex`, `etex`, `lambda`, `latex`, `metafont`, `metapost`, `omega`, `pdflatex`, `pdftex`, `tex`.

`--find-elatex-input` *FILE*
: Find e-LaTeX input file.

`--find-etex-input` *FILE*
: Find e-TeX input file.

`--find-executable` *FILE*
: Find a MiKTeX executable.

`--find-lambda-input` *FILE*
: Find Lambda input file.

`--find-latex-input` *FILE*
: Find LaTeX input file.

`--find-metafont-input` *FILE*
: Find METAFONT input file.

`--find-metapost-input` *FILE*
: Find MetaPost input file.

`--find-omega-input` *FILE*
: Find Omega input file.

`--find-other-executable` *FILE*
: Find an executable.

`--find-pdflatex-input` *FILE*
: Find pdfLaTeX input file.

`--find-pdftex-input` *FILE*
: Find pdfTeX input file.

`--find-tex-input` *FILE*
: Find TeX input file.

`--list-modes`
: List all known METAFONT modes.

`--local-root` *root*
: Specify the local TEXMF root.

`--mkpsres`
: Update the PostScript resource database ‘`psres.dpr`’. You can use this option in conjunction with `--search` (see below).

`--mkpsres=‘dir’`
: Add a new font directory to the PostScript resource database ‘`psres.dpr`’.

`--personal`
`-p` Do not use a personal configuration file.

`--personal=`*filename*
`-p`*filename*
Define the location of the personal configuration file.

`--print-only`
`-n` Print what would be done. Nothing is changed.

`--quiet` Suppress screen output.

`--reconfigure`
Reconfigure MiKTEX.

`--report` Create a configuration report.

`--root-directories` *dirlist*
`-r` *dirlist* Specify the list of TEXMF root directories.

`--search` Search for PS resource files (requires `--mkpsres`).

`--update-fndb`
`-u` Refresh the whole file name database.

`--update-fndb=`*root*
`-u`*root* Refresh the file name database for a specific TEXMF root.

`--verbose`
`-v` Print information on what is being done.

`--version`
`-V` Print the version number and exit.

A.5 `mp` (MetaPost)

MetaPost (installed as `mp`) reads a series of pictures specified in the MetaPost programming language, and outputs corresponding PostScript code.

Synopsis

```
mp [option...] [name[.mp]] [command...]
mp [option...] "&format" [command...]
```

Options

`--c-style-errors`
Show C/C++ style error messages. This switch implies `\scrollmode`.

`--initialize`
Initializes MetaPost's internal tables so that they can be dumped.

`--help` Shows a short help screen and exits successfully.

`--tex=`*texprogram*
: Uses *texprogram* instead of `tex` when compiling text labels. This flag overrides the environment variable `TEX`.

`--version`
: Prints version information and exits successfully.

Aliases

`inimp` Equivalent to '`mp --ini`'.

`mpost` Equivalent to '`mp`'.

`virmp` Equivalent to '`mp`'.

Environment Variables

`TEX` Specifies the TEX compiler which should be used when compiling text labels.

Documentation

For a complete description of the MetaPost language, see AT&T technical report CSTR-162, available as the file '`mpman.ps`'.

A.6 `omega`

Omega is a 16-bit enhanced version of TEX.

Synposis

`omega [`*option*`...] [`*name*`[.tex]] [`*command*`...]`

Options

Omega supports the common compiler options (see Section A.1 [Common Compiler Options], page 30).

Aliases

`iniomega` Equivalent to `omega --ini`.

`viromega` Equivalent to `omega`.

`lambda` Equivalent to `omega "&lambda"`.

Documentation

For a complete description of Omega, see the Omega manual, available as the file '`omega-manual.dvi`'.

A.7 pdftex

pdfTEX is a special version of TEX that outputs PDF.

Synopsis

```
pdftex [option...] [name[.tex]] [command...]
pdftex [option...] "&format" [command...]
```

Options

Besides the common command-line switches (see Section A.1 [Common Compiler Options], page 30), pdfTEX supports these options:

`--font-max=`*n*
: Sets the internal `font_max` to *n*. `font_max` is the maximum internal font number; must not exceed 5000.

Aliases

`inipdftex`
: Equivalent to `pdftex --ini`.

`virpdftex`
: Equivalent to `pdftex`.

`pdflatex` Equivalent to `pdftex "&pdflatex"`.

Documentation

For a complete description of pdfTEX, see the the pdfTEX User Manual, available as file '`pdftexman.pdf`'.

A.8 tex

`tex` is Donald Knuth's TEX compiler.

Synopsis

```
tex [option...] [name[.tex]] [command...]
tex [option...] "&format" [command...]
```

Options

Besides the common command-line switches (see Section A.1 [Common Compiler Options], page 30), `tex` supports the following options:

`--font-max=`*n*
: Sets the internal `font_max` to *n*. `font_max` is the maximum internal font number; must not exceed 5000.

Aliases

`latex` Equivalent to `tex "&latex"`.

`initex` Equivalent to `tex --ini`.

`virtex`
Equivalent to `tex`.

See Also

See Section A.9 [texify], page 38, for an alternative way to invoke TEX.

Documentation

For a complete description of TEX, see *The TEXbook* by Donald E. Knuth.

A.9 `texify`

`texify` runs Texinfo or LaTEX input files through TEX (pdfTEX) in turn until all cross-references are resolved, building all indices.

Synopsis

`texify [`*option*`]...` *file*`...`

The directory containing each *file* is searched for included files. The suffix of *file* is used to determine its language (LaTEX or Texinfo).

Makeinfo is used to perform Texinfo macro expansion before running TEX when needed.

Options

`-@` Use `@input` instead of `\input`; for preloaded Texinfo.

`-b`
`--batch` No interaction.

`-c`
`--clean` Remove all auxiliary files.

`-e`
`--expand` Force macro expansion using makeinfo.

`-I` *dir* Search *dir* for input files.

`-h`
`--help` Display this help and exit successfully.

`-l` *lang*
`--language=`*lang*
Specify the *lang* of *file*: `latex` or `texinfo`.

`--max-iterations=`*n*
: Limits the number of iterations to prevent endless processing. The default for *n* is 5.

`--mkidx-option=`*option*
: Pass *option* to the index generator.

`-p`
`--pdf`
: Use pdfTeX or pdfLaTeX for processing.

`-q`
`--quiet`
: No output unless errors (implies `--batch`).

`--run-viewer`
: Run a viewer on the resulting DVI/PDF file.

`-s`
`--silent`
: Same as `--quiet`.

`-t` *cmd*
`--texinfo=`*cmd*
: Insert *cmd* after `@setfilename` in copy of input file. Multiple values accumulate.

`--tex-option=`*option*
: Pass *option* to (La)(pdf)TeX.

`-v`
`--version`
: Display version information and exit successfully.

`--viewer-option=`*option*
: Pass *option* to the viewer.

Environment Variables

The values of the `BIBTEX`, `LATEX` (or `PDFLATEX`), `MAKEINDEX`, `MAKEINFO`, `TEX` (or `PDFTEX`), and `TEXINDEX` environment variables are used to run those commands, if they are set.

Aliases

`texi2dvi` Equivalent to `texify`.

Appendix B The MiKTEX Configuration File

This chapter describes the contents of the MiKTEX configuration file (`miktex.ini`).

B.1 Specifying search paths

Search paths are used by MiKTEX to find special files (such as TEX input files) within a comprehensive directory hierarchy.

A search path is a semicolon-separated list of directory paths. This list is traversed from left to right, i.e. the first directory is searched first.

In a directory path, the following character sequences have a special meaning:

`%R` A placeholder for the list of TEXMF root directories.

`//` A flag, which causes MiKTEX to search recursively.

Example

Assuming that `C:\Program Files\MiKTeX;\\myserver\texmf` is the list of TEXMF root directories, the search path `.;%R\tex\latex//;%R\tex\generic//` causes LaTEX to search its input files in the following locations:

1. In the current directory (`.`).
2. In the directory `C:\Program Files\MiKTeX\tex\latex` and in all directories below it.
3. In the directory `\\myserver\texmf\tex\latex` and in all directories below it.
4. In the directory `C:\Program Files\MiKTeX\tex\generic` and in all directories below it
5. In the directory `\\myserver\texmf\tex\generic` and in all directories below it.

Testing a new search path

You can use the configuration utility `initexmf` to test whether an input file can be found via the current search path. For example, the command

```
C:\> initexmf --find-latex-input a4.sty
```

searches for the LaTEX input file `a4.sty`. The full path name is printed if the file was found.

B.2 Contents of a MiKTEX Configuration File

A MiKTEX configuration file is divided into several named sections. Each section contains configuration settings for a specific application or feature.

B.2.1 [BibTeX]: BibTEX Configuration Settings

The section `[BibTeX]` contains BibTEX related configuration settings.

`Input Dirs`
: Search path (see Section B.1 [Search Paths], page 40) for BibTEX input files (both databases and style files).

`min_crossrefs`
: Minimum number of cross-refs required for automatic `cite_list` inclusion.

B.2.2 [Dvips]: Dvips Configuration Settings

The section `[Dvips]` contains Dvips related configuration settings.

`CONFIGPath`
: Where Dvips searches its configuration files (e.g. `config.ps`).

`ENCPath` Where Dvips searches for `.enc` files.

`GraphicsPath`
: Where Dvips searches for `.eps` files.

`MAPPath` Where Dvips searches for `.map` files.

`PSPath` Where Dvips searches for PS header files.

B.2.3 [Graphics]: Graphics Conversion Rules

The section `[Graphics]` contains graphics conversion rules. Each rule has the syntax

.fromext.toext=commandline

fromext is the file name extension of the source file. *toext* is the file name extension of the destination file. *commandline* is the command-line which does the conversion. The command-line may include the following placeholders:

`%i` The name of the input file.

`%o` The name of the output file.

The standard MiKTEX configuration file contains the following rules:

```
.gif.bmp=giftopnm %i | ppmtobmp -windows > %o
.pcx.bmp=pcxtoppm %i | ppmtobmp -windows > %o
.png.bmp=pngtopnm %i | ppmtobmp -windows > %o
.tga.bmp=tgatoppm %i | ppmtobmp -windows > %o
.tif.bmp=tifftopnm %i | ppmtobmp -windows > %o
.tiff.bmp=tifftopnm %i | ppmtobmp -windows > %o
```

B.2.4 [Magic]: Memory Settings for TEX & Friends

The section `[Magic]` contains memory related configuration settings. The values are used by TEX, pdfTEX and Omega for the dynamic allocation of certain data structures.

Format-Independent Values

The following parameters can be changed at run time to extend or reduce TEX's capacity. They may have different values in `INITEX` and in production versions of TEX.

`mem_min` Smallest index in TEX's internal `mem` array; must be 0 or more; must be equal to `mem_bot` in `INITEX`, otherwise `<=mem_bot`.

`mem_max` Greatest index in TEX's internal `mem` array; must be strictly less than 1073741823.

`buf_size` Maximum number of characters simultaneously present in current lines of open files and in control sequences between `\csname` and `\endcsname`; must not exceed 1073741823.

`error_line`
Width of context lines on terminal error messages.

`half_error_line`
Width of first lines of contexts in terminal error messages; should be between 30 and (`error_line` - 15).

`max_print_line`
Width of longest text lines output; should be at least 60.

`stack_size`
Maximum number of simultaneous input sources.

`max_in_open`
Maximum number of input files and error insertions that can be going on simultaneously.

`font_max` Maximum internal font number; must not exceed 5000.

`font_mem_size`
Number of words of `font_info` for all fonts.

`param_size`
Maximum number of simultaneous macro parameters.

`nest_size`
Maximum number of semantic levels simultaneously active.

`max_strings`
Maximum number of strings; must not exceed 1073741823.

`string_vacancies`
The minimum number of characters that should be available for the user's control sequences and font names, after TEX's own error messages are stored.

`pool_size`
Maximum number of characters in strings, including all error messages and help texts, and the names of all fonts and control sequences; must exceed `string_vacancies` by the total length of TEX's own strings, which is currently about 23000.

`save_size`
: Space for saving values outside of current group; must be at most 1073741823.

`trie_size`
: Space for hyphenation patterns; should be larger for `INITEX` than it is in production versions of TEX.

`trie_op_size`
: Space for "opcodes" in the hyphenation patterns.

Format-Dependent Values

Like the preceding parameters, the following quantities can be changed at run time to extend or reduce TEX's capacity. But if they are changed, it is necessary to rerun the initialization program `INITEX` to generate new tables for the production TEX program. One can't simply make helter-skelter changes to the following constants, since certain rather complex initialization numbers are computed from them.

`mem_bot` Smallest index in the mem array dumped by `INITEX`; must not be less than `mem_min`.

`mem_top` Largest index in the mem array dumped by `INITEX`; must be substantially larger than 0 and not greater than `mem_max`.

B.2.5 [MakeIndex]: MakeIndex Configuration Settings

The section `[MakeIndex]` contains MakeIndex related configuration settings.

`INDEXSTYLE`
: Search path (see Section B.1 [Search Paths], page 40) for MakeIndex style files.

B.2.6 [MakePK]: MakePK Configuration Settings

The section `[MakePK]` contains configuration settings that are related to the auto-creation of packed raster fonts.

`DestDir` The specification of a directory where newly created PK (Packed Raster Font) files are to be installed.

The specifiation may include special character sequences which will be replaced at search-time:

`%m`	The current METAFONT mode.
`%d`	The horizontal resolution (in dots per inch).
`%s`	The font supplier (e.g. `public`).
`%t`	The typeface name (e.g. `cm`).

Admin note: All MiKTEX users must have permission to create files in the specified directory.

B.2.7 [MakeTFM]: MakeTFM Configuration Settings

`DestDir` Where new `.tfm` files are to be installed.
The specifiation may contain special character sequences which are replaced at search-time:

`%s` The font supplier (e.g. `public`).

`%t` The typeface name (e.g. `cm`).

Admin note: MiKTeX users must have permission to add files to the specified directory.

B.2.8 [METAFONT]: METAFONT Configuration Settings

The section `[METAFONT]` contains METAFONT related configuration settings.

`Input Dirs`
Search path (see Section B.1 [Search Paths], page 40) for METAFONT input files.

B.2.9 [MetaPost]: MetaPost Related Configuration Settings

The section `[MetaPost]` contains MetaPost related configuration settings.

`Input Dirs`
Where MetaPost searches for input files.

B.2.10 [MiKTeX]: General Configuration Settings

The section `[MiKTeX]` contains general configuration settings and search path specifications.

General Configuration Settings

`Trace` This is a comma separated list of trace options:

`notrace` Inhibits trace output to the console.

`fndb` Traces the file name database.

`filesearch`
Traces the find-file machinery.

`access` Traces file accesses.

`process` Traces secondary processes.

`tcx` Traces TCX tables.

`error` Traces error conditions.

`time` Traces execution time.

`TraceFile`
The name of the trace file.

Search Path Specifications

`AFMPath` Used to locate Adobe font metric files (`*.afm`).

`BASEPath` Used to locate METAFONT base files (`*.base`).

`ENCPath` Used to locate `*.enc` files.

`EXEPath` Used to locate executables.

`FMTPath` Used to locate TEX format files (`.fmt`). Also used to locate e-TEX format files (`.efmt`).

`GraphicsPath`
Used to locate graphics files (`*.eps;*.bmp;...`).

`MAPPath` Used to locate font map files (`*.map`).

`MEMPath` Used to locate MetaPost memory files (`.mem`).

`OFMPath` Used to locate Omega font metric files (`*.ofm`).

`OVFPath` Used to locate Omega virtual fonts (`*.ovf`).

`PKPath` Used to locate packed font raster files (`*.pk`).

`PSPath` Used to locate PostScript header files (`*.enc;*.map`);

`TCXPath` Used to locate character translation files (`.tcx`).

`TFMPath` Used to locate TEX font metric files (`*.tfm`).

`TTFPath` Used to locate TrueType fonts (`*.ttc;*.ttf`).

`Type1Path`
Used to locate Type1 fonts (`*.pfa;*.pfb`).

`VFPath` Used to locate virtual fonts (`*.vf`).

B.2.11 [Omega]: Omega Configuration Settings

The section `[Omega]` contains Omega related configuration settings:

`Input Dirs`
The search path (see Section B.1 [Search Paths], page 40) for Omega input files.

`OCPPath` Where Omega searches for OCP files.

B.2.12 [otp2ocp]: `otp2ocp` Configuration Settings

`Input Dirs`
Used by `otp2ocp` to locate OTP files (`.otp`).

B.2.13 [pdfTeX]: pdfTeX Configuration Settings

The section `[pdfTeX]` contains pdfTeX related configuration settings.

`Input Dirs`
: Where pdfTeX searches for input files.

`PSPath`
: Where pdfTeX searches for font mapping files.

B.2.14 [ps2pk]: `ps2pk` Configuration Settings

The section `[ps2pk]` contains configuration settings for the `ps2pk` utility:

`PSResPath`
: Where `ps2pk` searches for PS resource files.

B.2.15 [TeX]: TeX Configuration Settings

The section `[TeX]` contains TeX-related configuration settings.

`Editor`
: The command to be started when you press `e` in the error menu.
You can use the following placeholders:

  `%f` Will be replaced by the name of the input file that caused the error.

  `%h` Will be replaced by a help text.

  `%l` Will be replaced by the line number.

  `%m` Will be replaced by the error message.

  `%t` Will be replaced by the name of the transcript file.

  For example, a suitable value for WinEdt would be '`winedt %f -G(1,%l,0) -S(12,+1,0)`'.
  For NT Emacs, set `Editor` to '`gnulientw -F +%l %f`'.

`Input Dirs`
: Used by TeX to locate input files.

B.2.16 [Yap]: Yap Configuration Settings

`Input Dirs`
: Used by Yap to locate DVI files (`*.dvi`).

Appendix C Index

%

-

.

/

^

\

8

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Table of Contents