TEX Implementations

Piet van Oostrum

November 21, 2002

Contents

Original TEX implementation

- Knuth's implementation: **Web = Pascal+TEX**
  - Pascal part for code
  - TEX part for documentation
  - mixed together for literate Programming
  - program `tangle` to extract Pascal code
  - program `weave` to extract TEX code

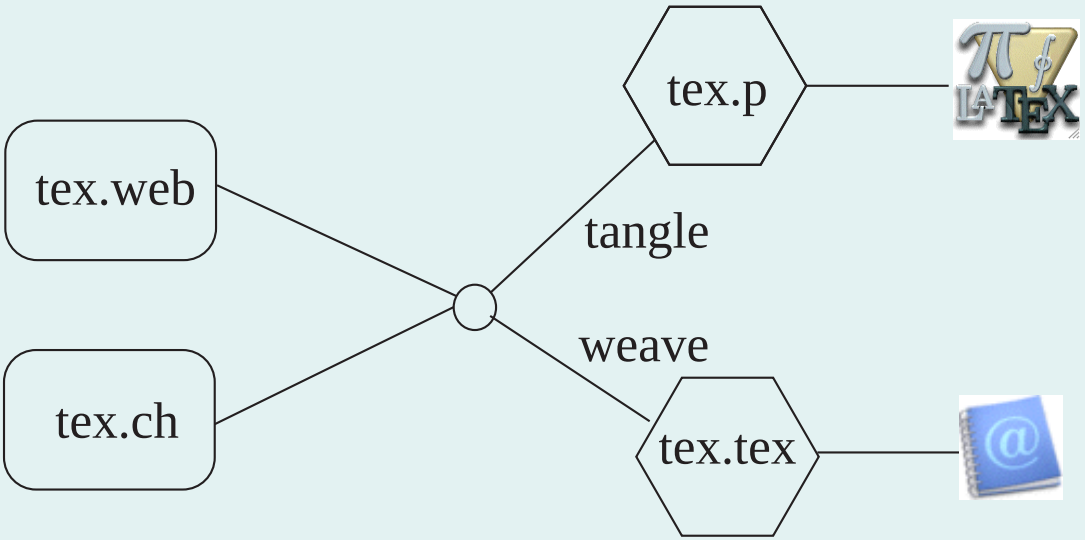
Example Web code

```
@ The following subroutine compares string |s| with another string of the
same length that appears in |buffer| starting at position |k|;
the result is |true| if and only if the strings are equal.
Empirical tests indicate that |str_eq_buf| is used in such a way that
it tends to return |true| about 80 percent of the time.

@p function str_eq_buf(@!s:str_number;@!k:integer):boolean;
  {test equality of strings}
label not_found; {loop exit}
var j: pool_pointer; {running index}
@!result: boolean; {result of comparison}
begin j:=str_start[s];
while j<str_start[s+1] do
  begin if so(str_pool[j])<>buffer[k] then
    begin result:=false; goto not_found;
    end;
  incr(j); incr(k);
  end;
result:=true;
not_found: str_eq_buf:=result;
end;
```

Tangle and Weave

- Tangle and weave take a `.web` file and a `.ch` file
- The `.ch` file contains modifications to the `.web` file
- E.g. for small corrections or system-dependent changes
- This isolates local changes from the normal TEX updating process


tex.web
tex.ch
tangle
weave
tex.p
tex.tex

Web2C

- Knuth used only simple Pascal
- Many features of Pascal were not portable
- Or had buggy implementations
  - Memory management
  - I/O
- Advent of Unix and C:
  - Hand translation of Pascal to C (Common TEX):
    Problem with keeping in sync with Knuth
  - Web2c:
    Automatic translation from Web-Pascal to C
    System-specific parts (allocation of arrays, I/O) written in C

Modern implementations

- Most modern implementation are probably based on Web2c
- Unix-implementations:
  - Web2c – bare-bones implementations with the Basic TEX and Metafont stuff, and some dvi-drivers.
  - Karl Berry added path-searching:
    Directory structure got too big
    Too slow on Network File Systems
    Solution: Add a filename database (ls-R)
  - TEX Directory Structure (TDS): Standardized directory structure for all implementations.
  - **tetex**: Packaging by Tomas Esser
    Contains web2c with many packages, doc etc.
    Is now the standard Unix implementation

TDS Example

```
 bibtex
  bib
  bst
    ams
      amsalpha.bst
      amsplain.bst
    base
      plain.bst
      siam.bst
      unsrt.bst
doc
dvips
fonts
  pk
  tfm
  type1
  vf
```

TDS Example – continued

```
metafont
metapost
tex
  context
  generic
  latex
    amsmath
    base
    carlisle
    context
    custom-bib
    fancyhdr
  plain
web2c
```

How dows TEX find its files?

- Original TEX: (probably)
  - Look only in current directory
    Or specify full path
- First Web2C implementations:
  - Specify `TEXINPUTS` environment variable:
    `TEXINPUTS=.:/usr/local/tex:$HOME/tex`
    Input files are searches in these 3 directories
  - Similar variables for bibtex (`BIBINPUTS`) etc.
  - Problem: No distinction between plain TEX, LATEX, context etc.

Refinement

- Use different environment variables for different TEXprograms:
- For LATEX: `TEXINPUTS.latex`
- For context: `TEXINPUTS.context`
- For others: `TEXINPUTS`
- The TEX program looks at the name with which it is invoked and chooses the corresponding variable, if present, otherwise the default.

Search method

- Just searching through all directories is expensive, especially:
  - when a large directory tree is used (LaTeX)
  - when the files are on the network
- Solution (Karl Berry):
  `kpathsearch` a.k.a `kpathsea`
  this uses the `ls-R` files that contains the locations of the files
- A special file `texmf.cnf` contains
  - the values of the environment variables
  - other parameters, e.g. array sizes
- This file can specify several TeX trees, e.g.
  - `texmf` for the distribution
  - `texmf.os` for additions for the operating system
  - `texmf-local` for local additions
  - `$HOME/texmf` for user files

Example texmf.cnf

```
% The main tree, which must be mentioned in $TEXMF, below:
TEXMFMAIN = $SELFAUTOPARENT/texmf
% A place for local additions to a "standard" texmf tree.
TEXMFLOCAL = $SELFAUTOPARENT/texmf-local

% User texmf trees can be catered for like this...
HOMETEXMF=$HOME/texmf

% Now, list all the texmf trees. If you have multiple trees,
% use shell brace notation, like this:
TEXMF = {$HOMETEXMF,$TEXMFLOCAL,!!$TEXMFMAIN}
% The braces are necessary.

% LaTeX-specific macros are stored in latex.
TEXINPUTS.latex = .;$TEXMF/tex/{latex,generic,}//

% Plain TeX.  Have the command tex check all directories as a last
% resort, we may have plain-compatible stuff anywhere.
TEXINPUTS.tex = .;$TEXMF/tex/{plain,generic,}//

% Context macros by Hans Hagen:
TEXINPUTS.context   = .;$TEXMF/{pdfetex,pdftex,etex,tex}/
                                      {context,plain,generic,}//
```

Notes

- `$SELFAUTOPARENT` is the directory above the one where the TEX programs are found
- `//` means search also subdirectories
- `!!` means that (only) the `ls-R` file is used for searching, otherwise also a normal directory search is done
- For `!!` directories the `ls-R` file **must be updated** if something is added or removed
- This is done by running the program `texhash` or `mktexlsr` or choosing a similar command form the MikTEX of `fptex` menu.
- You can also use `TEXINPUTS` by putting an empty entry where the standard path should be used, e.g.
  `TEXINPUTS=$HOME/mytex:`

TEXMF

- Where does `kpathsearch` find its `texmf.cnf`?
  - If environment variable `TEXMFCNF` is set, this is a path where `texmf.cnf` files are searched.
  - All files found are used, one after the other
  - In this way you only have to put modifications in your own `texmf.cnf`
  - This makes updating the installation easier
- The default location is usually in `web2c` in the standard `texmf` directory
- This can also be changed by the `TEXMF` environment variable

MS Windows implementations 1

- Free implementations
  - MikTEX: Lean and mean implementation by Christian Schenk (based upon web2c)
    Divided in 3 levels: small, large, total
    More packages can be incrementally installed
    From Internet, disk or CDROM

MS Windows implementations 1

- Free implementations
  - MikTEX: Lean and mean implementation by Christian Schenk (based upon web2c)
    Divided in 3 levels: small, large, total
    More packages can be incrementally installed
    From Internet, disk or CDROM
  - fpTEX by François Popineau:
    Adaption of tetex to MS Windows
    Quite big and complete
    Base for TEX-live CDROMs

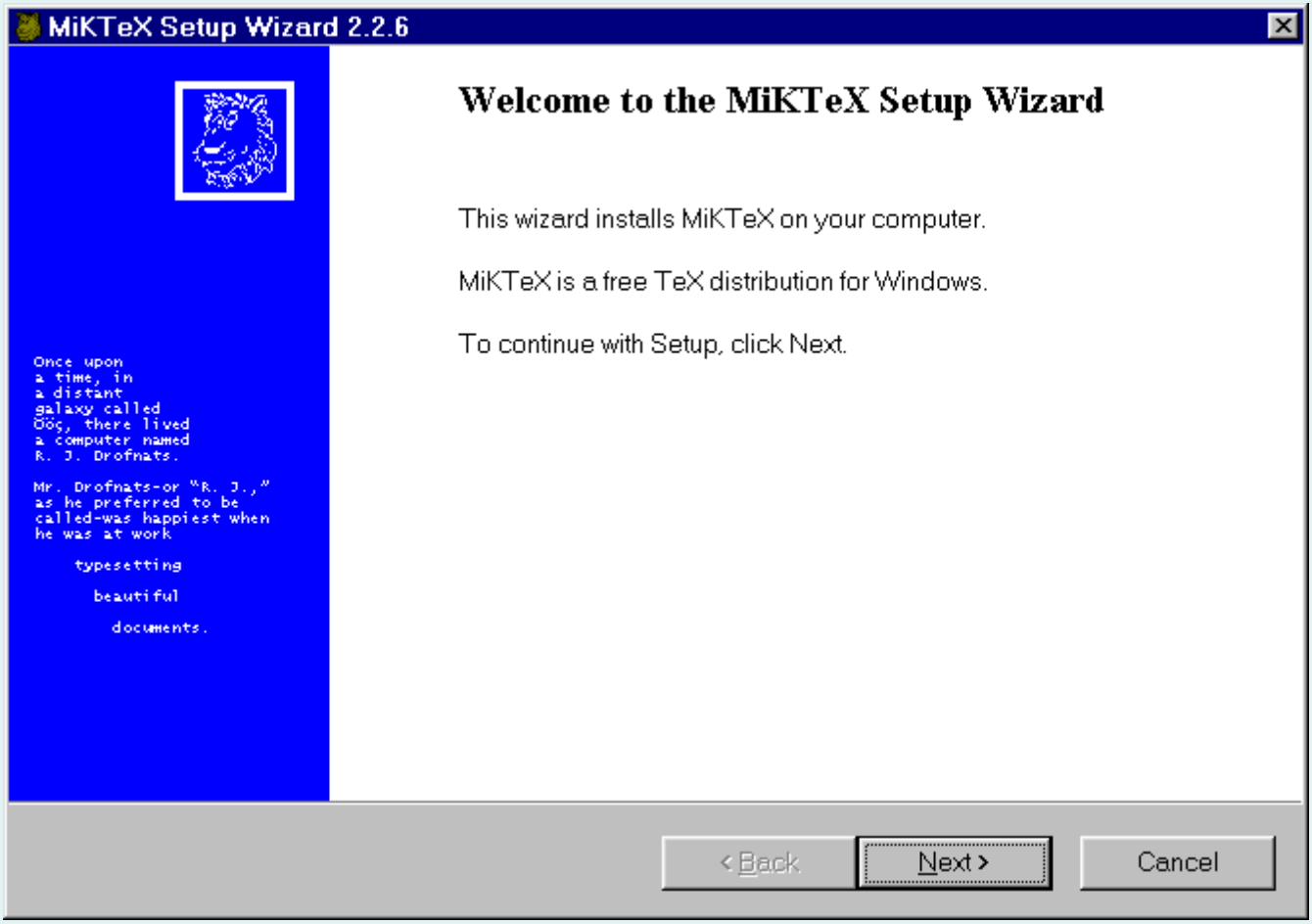
MS Windows implementations 1

- Free implementations
  - MikTEX: Lean and mean implementation by Christian Schenk (based upon web2c)
    Divided in 3 levels: small, large, total
    More packages can be incrementally installed
    From Internet, disk or CDROM
  - fpTEX by François Popineau:
    Adaption of tetex to MS Windows
    Quite big and complete
    Base for TEX-live CDROMs
- Several editors/IDEs (Integrated Development Environments) are available to use these.

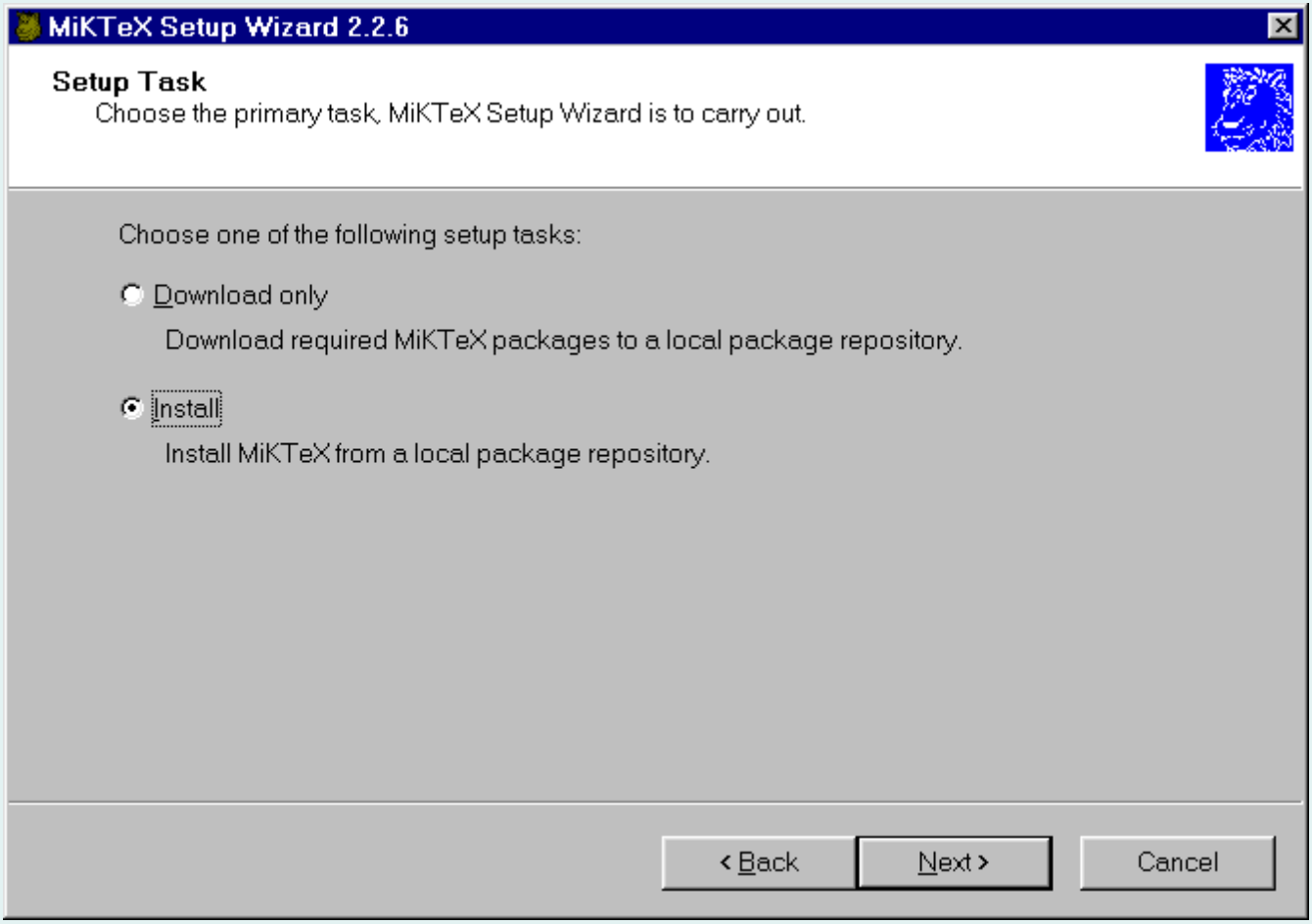
MS Windows implementations 2

- Coomercial implementations
  - Y & Y TEX
  - VTEX
  - Bakoma TEX (shareware)
- Usually have a GUI
- Good interaction with rest of Windows
  - Graphics
  - Drag and drop/cut and paste
  - Fonts

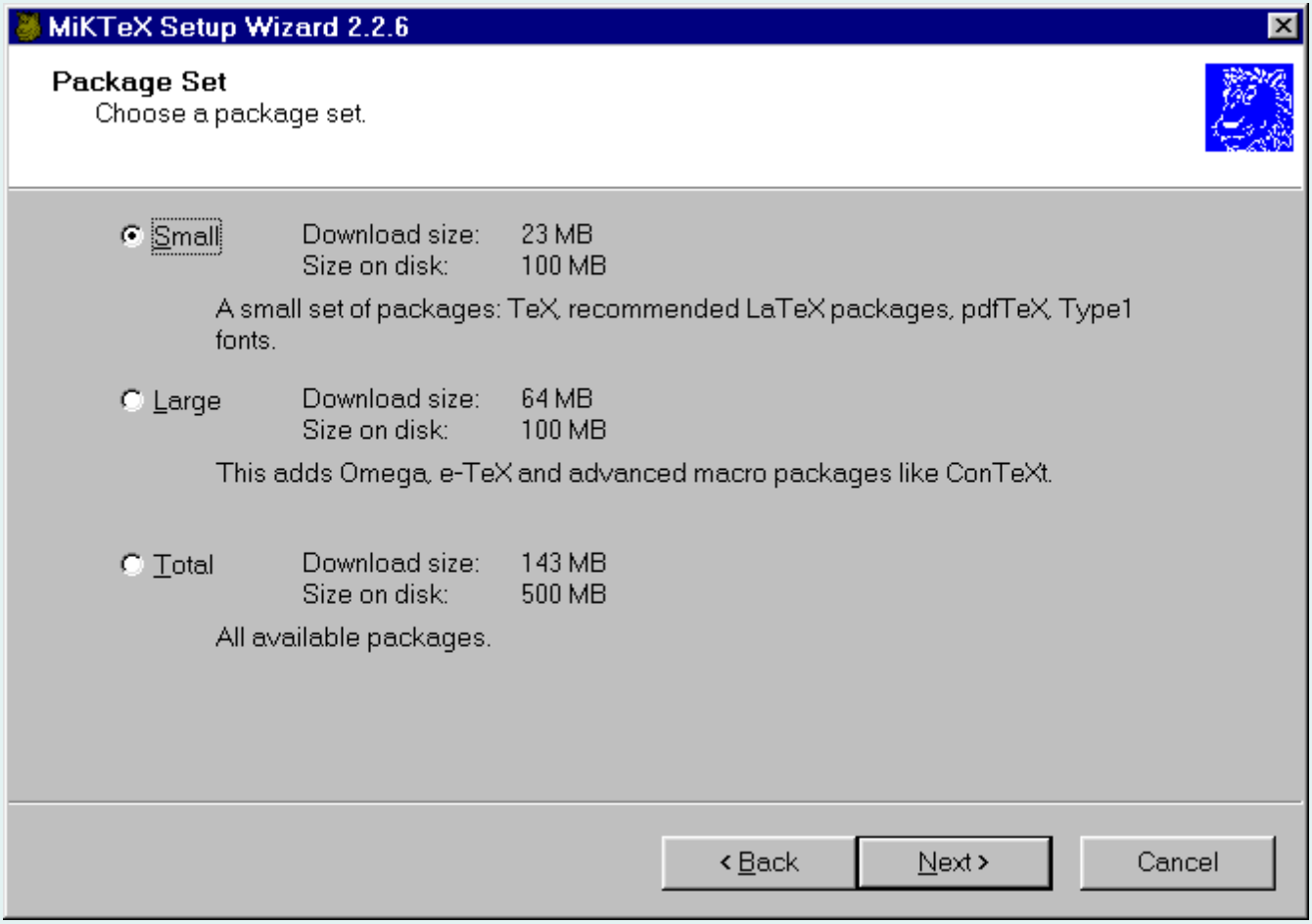
MikTeX Installation 1



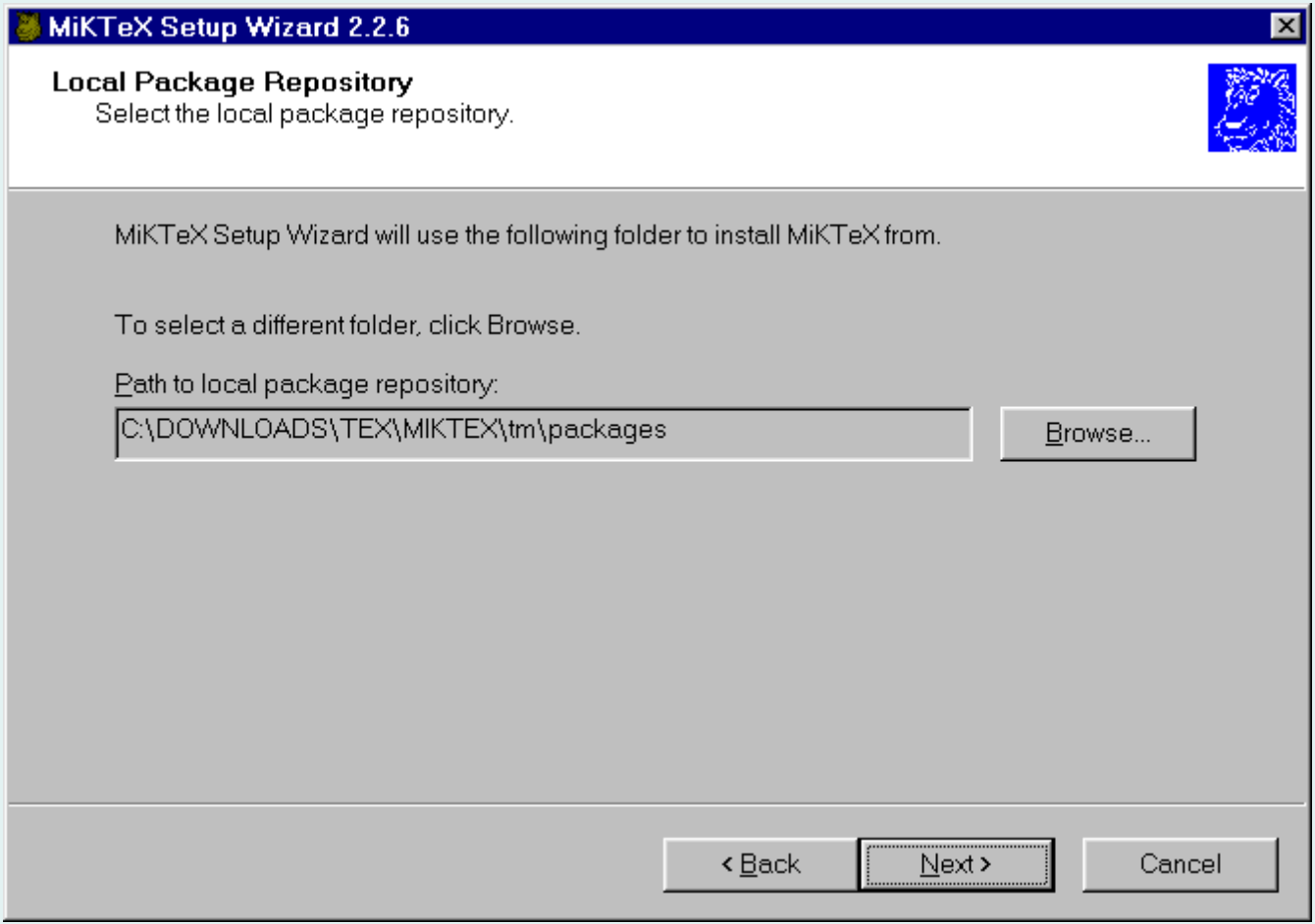
MikTeX Installation 2



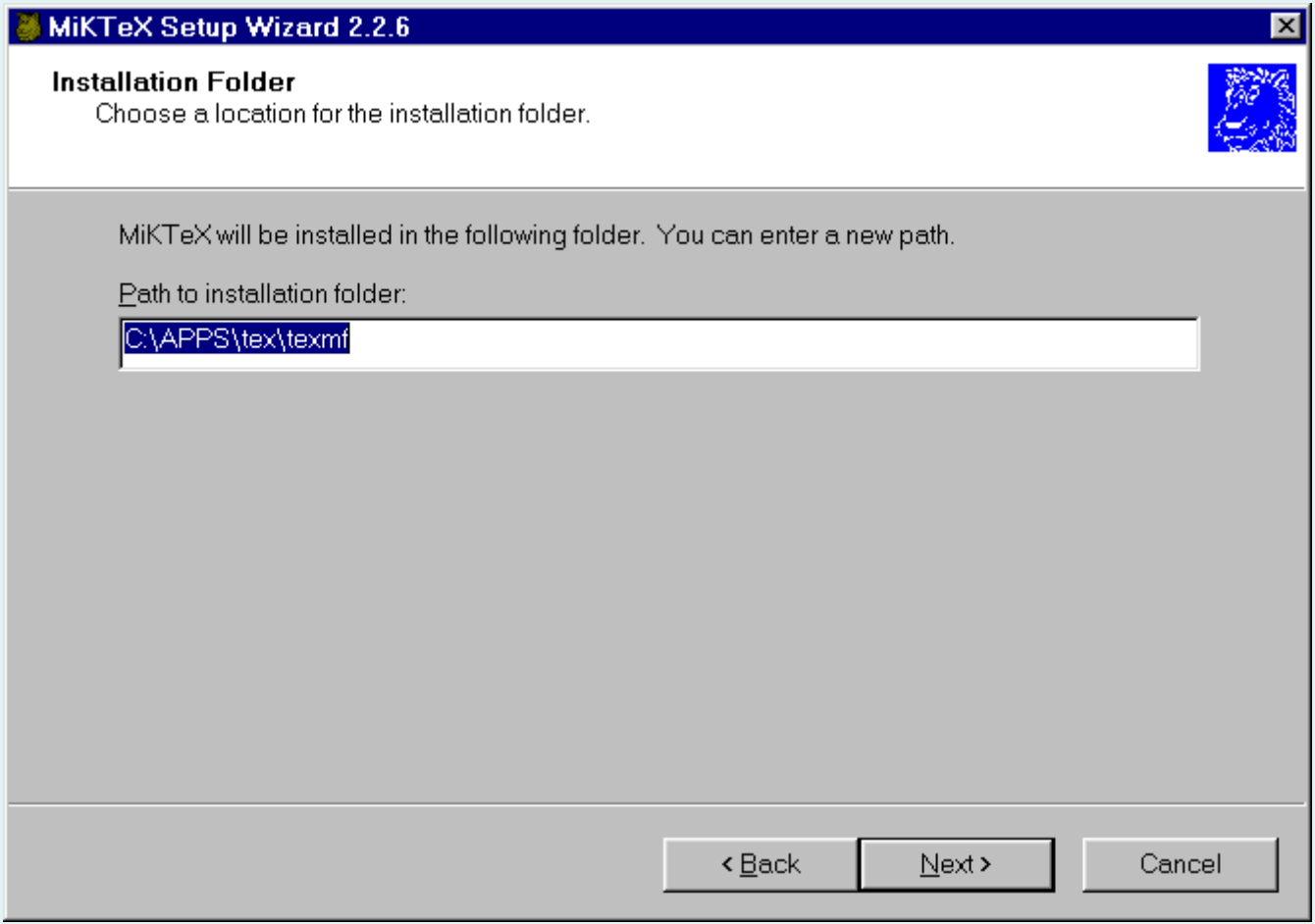
MikTeX Installation 3



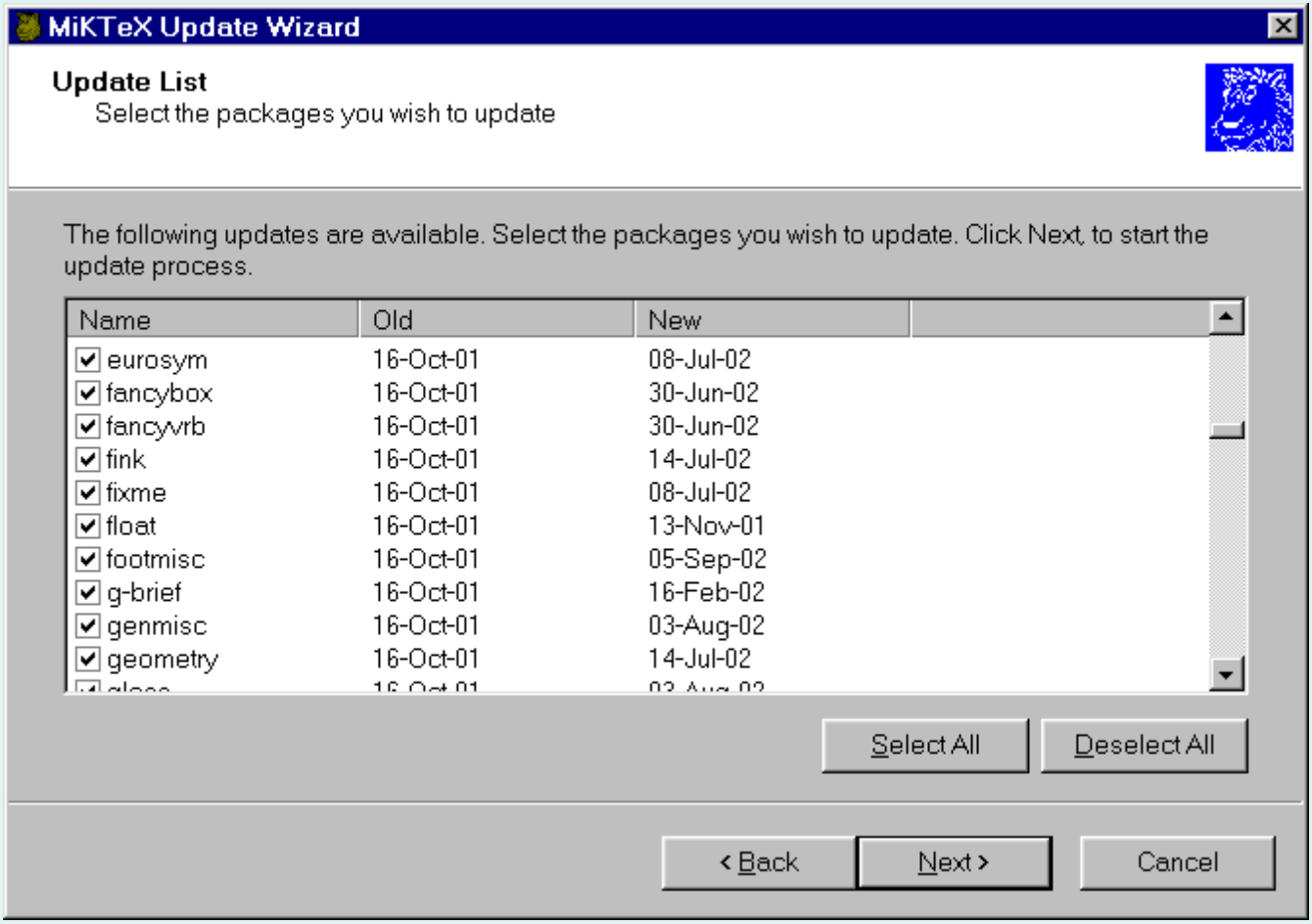
MikTeX Installation 4



MikTeX Installation 5



MikTeX Update Wizard



Mac

- Macintosh implementations:
  - OzTeX – Specific Macintosh implementation
    Uses Mac paradigms and GUI
  - CMacTeX – Web2C based
    Is more command-line oriented
  - Both are shareware with a modest fee
  - Nowadays not often updated
- MacOS X (Unix-like kernel):
  - teTEXimplementation (Gerben Wierda)
  - simple IDE (TEXshop or iMacTEX) added

Mac OSX i-Installer

DEMO

How to install TEX

- MS Windows
  - If you want a small installation:
    use MikTEX
  - If you want a complete (and large) installation:
    Use MikTEX or the TEX-live CDROM

teTEX installation

- On Linux:
  - Most Linux systems have teTEX installed
  - Sometimes they have outdated or even bad installations
  - Most systems have an easy installation system (like RPM's) with complete binaries
  - Before installing a new version it is usually wise to uninstall the old version.

teTEX installation

- On other Unix systems
  - Sometimes you can get binary packages
  - Otherwise you have to compile yourself
  - download some tar-archives and unpack these
  - run `./configure`
  - run `make`
  - run `make install`
  - Hope you get no errors

TEX-live installation

- Put the CDROM in your computer
- Start the install program if it does not autostart
- If you don't want everything you have to choose which parts you want installed
  - This may be confusing

Adaption for Dutch language

- All current implementations have provisions for multi-language typesetting
  - Multiple hyphenation patterns
  - Babel support
- Support for Dutch is usually not enabled by default
- To enable it you have to edit the file `language.dat`
- remove the % before the line 'dutch nehyph.tex' (or similar)
- Rebuild the formats (this depends on the implementation)
- mikTEX and fpTEX have Start menu items for this
- Also possible from the command line
- on teTeX you can run the `texconfig` utility