



# OzT<sub>E</sub>X User Guide

Version 6.0, July 2008

Andrew Trevorrow

[andrew@trevorrow.com](mailto:andrew@trevorrow.com)

[www.trevorrow.com](http://www.trevorrow.com)

OzT<sub>E</sub>X is a Macintosh implementation of the T<sub>E</sub>X typesetting system. Version 6.0 requires an Intel Mac running OS 10.4 or later. This document assumes you know how to use a Macintosh. It also assumes you know how to create a T<sub>E</sub>X (or L<sup>A</sup>T<sub>E</sub>X) input file.

OzT<sub>E</sub>X aims to provide a standard T<sub>E</sub>X environment for the Macintosh that is easy to extend or customize to suit your particular needs. If you have used T<sub>E</sub>X on some other computer then the way OzT<sub>E</sub>X works shouldn't be too surprising.

OzT<sub>E</sub>X is shareware, so you are welcome to try it out before paying the registration fee (see the "Shareware Fee" item in OzT<sub>E</sub>X's Help menu for details about how to pay). Comments, bug reports and suggestions are all welcome. I'm happy to help sort out any initial installation problems but please note that if you want continued email support then you'll have to pay the shareware fee.

## Contents

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                   | <b>1</b>  |
| <b>2</b> | <b>A guided tour of OzTeX</b>         | <b>3</b>  |
| 2.1      | Starting OzTeX                        | 3         |
| 2.2      | Choosing an editor                    | 3         |
| 2.3      | Editing a TeX input file              | 3         |
| 2.4      | Running TeX                           | 4         |
| 2.5      | Viewing a DVI file                    | 5         |
| 2.5.1    | Setting the default view              | 5         |
| 2.5.2    | Paper coordinates                     | 5         |
| 2.5.3    | Zooming in and out                    | 6         |
| 2.5.4    | Scrolling around                      | 6         |
| 2.5.5    | Checking for errors                   | 6         |
| 2.5.6    | Selecting pages                       | 7         |
| 2.5.7    | Mouse shortcuts                       | 7         |
| 2.5.8    | Keyboard shortcuts                    | 7         |
| 2.5.9    | Switching to your editor              | 8         |
| 2.6      | Printing a DVI file                   | 9         |
| 2.6.1    | Non-PostScript printing               | 9         |
| 2.6.2    | Low-level PostScript printing         | 9         |
| 2.6.3    | Standard PostScript printing          | 10        |
| 2.6.4    | Forcing PostScript printing           | 10        |
| 2.7      | Positioning pages correctly           | 11        |
| 2.8      | Continuing from where you left off    | 11        |
| <b>3</b> | <b>OzTeX's menus</b>                  | <b>12</b> |
| 3.1      | The File menu                         | 12        |
| 3.2      | The Edit menu                         | 13        |
| 3.3      | The TeX menu                          | 14        |
| 3.4      | The View menu                         | 16        |
| 3.5      | The Tools menu                        | 19        |
| 3.6      | The Config menu                       | 20        |
| 3.7      | The Help menu                         | 21        |
| <b>4</b> | <b>Configuration files</b>            | <b>22</b> |
| 4.1      | The required syntax                   | 22        |
| 4.2      | Path names                            | 22        |
| 4.2.1    | Unix path syntax                      | 23        |
| 4.2.2    | Mac path syntax                       | 23        |
| 4.2.3    | Symbolic links and aliases            | 23        |
| 4.2.4    | Path names in input files             | 24        |
| 4.3      | Search paths and subfolder expansion  | 25        |
| 4.4      | The Default configuration files       | 26        |
| 4.5      | The Local configuration file          | 26        |
| 4.6      | Creating your own configuration files | 26        |
| 4.7      | Changing keyboard shortcuts           | 27        |
| 4.8      | Adding extra tools                    | 28        |

# CONTENTS

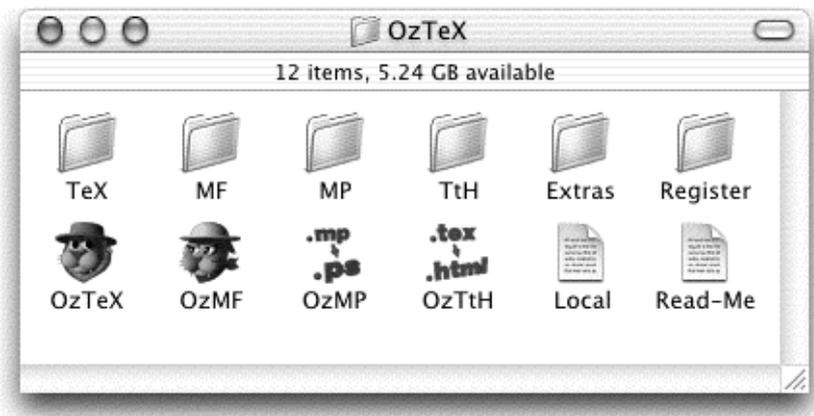
|           |  |           |
|-----------|--|-----------|
| <b>5</b>  | <b>Calling Unix commands</b>                           | <b>28</b> |
| 5.1       | Changing the shell . . . . .                           | 30        |
| 5.2       | Changing PATH . . . . .                                | 30        |
| <b>6</b>  | <b>OzTeX on OS X</b>                                   | <b>31</b> |
| 6.1       | Advantages . . . . .                                   | 31        |
| 6.2       | Known problems . . . . .                               | 31        |
| 6.3       | Using teTeX's texmf tree . . . . .                     | 31        |
| 6.4       | Tips for Unix beginners . . . . .                      | 32        |
| <b>7</b>  | <b>Task lists: a simple scripting facility</b>         | <b>33</b> |
| 7.1       | Waiting for an application to finish . . . . .         | 34        |
| 7.2       | File-specific tasks . . . . .                          | 34        |
| 7.3       | Using menu items to “drop” files . . . . .             | 34        |
| <b>8</b>  | <b>L<sup>A</sup>TeX</b>                                | <b>35</b> |
| 8.1       | Installing a new package . . . . .                     | 35        |
| 8.2       | Using L <sup>A</sup> TeX 2.09 . . . . .                | 35        |
| <b>9</b>  | <b>Building format files</b>                           | <b>36</b> |
| 9.1       | Building big formats . . . . .                         | 36        |
| 9.2       | Building custom formats . . . . .                      | 36        |
| <b>10</b> | <b>Fonts</b>   | <b>37</b> |
| 10.1      | TFM files . . . . .                                    | 37        |
| 10.2      | Creating new TFM files . . . . .                       | 37        |
| 10.3      | PK files . . . . .                                     | 38        |
| 10.4      | PostScript fonts . . . . .                             | 39        |
| 10.4.1    | Reencoding PostScript fonts . . . . .                  | 39        |
| 10.4.2    | Downloading PostScript fonts . . . . .                 | 40        |
| 10.4.3    | PostScript fonts on a non-PostScript printer . . . . . | 41        |
| 10.4.4    | PostScript or Computer Modern? . . . . .               | 41        |
| 10.5      | Macintosh screen fonts . . . . .                       | 42        |
| 10.6      | Encoding files . . . . .                               | 42        |
| 10.7      | Virtual fonts . . . . .                                | 42        |
| <b>11</b> | <b>OzMF</b>  | <b>43</b> |
| 11.1      | Making PK fonts on demand . . . . .                    | 44        |
| 11.2      | Making TFM files . . . . .                             | 44        |
| 11.3      | Setting the Metafont mode . . . . .                    | 45        |
| <b>12</b> | <b>OzMP</b>  | <b>45</b> |
| <b>13</b> | <b>OzTtH</b>   | <b>46</b> |
| <b>14</b> | <b>DVIPS</b>   | <b>46</b> |
| 14.1      | Downloading fonts used in EPS files . . . . .          | 47        |
| <b>15</b> | <b>Creating PDF files</b>                              | <b>47</b> |
| 15.1      | PDF via dvips . . . . .                                | 47        |
| 15.2      | Ghostscript or Distiller? . . . . .                    | 48        |
| 15.3      | PDF via pdfTeX or dvipdfm . . . . .                    | 48        |

## CONTENTS

|  |           |
|--|-----------|
| <b>16 Graphics</b>                                       | <b>48</b> |
| 16.1 Including Macintosh graphic files . . . . .         | 49        |
| 16.2 Previewing EPS files . . . . .                      | 51        |
| 16.3 Compatibility with other DVI translators . . . . .  | 52        |
| 16.4 Including PostScript files . . . . .                | 52        |
| 16.5 Rotating $\TeX$ material . . . . .                  | 53        |
| 16.6 Color . . . . .                                     | 53        |
| 16.7 Hyper $\TeX$ . . . . .                              | 53        |
| <b>17 Recovering from errors</b>                         | <b>54</b> |
| 17.1 Errors while loading a configuration file . . . . . | 54        |
| 17.2 Errors while loading an encoding file . . . . .     | 54        |
| 17.3 Errors while running $\TeX$ . . . . .               | 54        |
| 17.4 Errors while viewing a DVI file . . . . .           | 55        |
| 17.5 Errors while printing a DVI file . . . . .          | 56        |
| <b>18 Miscellany</b>                                     | <b>57</b> |
| 18.1 Transferring TFM/VF/PK/DVI files . . . . .          | 57        |
| 18.2 Oz $\TeX$ CD . . . . .                              | 57        |
| 18.3 Getting more information . . . . .                  | 57        |
| <b>References</b>  | <b>58</b> |
| <b>Index</b>   | <b>59</b> |

## 1 Introduction

After carrying out the installation instructions in the `Read-Me` file supplied with OzTeX you should have a folder that looks something like this:



The OzTeX folder can be called anything you like, and can be moved anywhere on your disk. It is strongly recommended that you do not change *any* of the supplied folders. Instead, create new folders for your own macros, fonts, formats, etc., and make suitable changes to the search paths in the `Local` configuration file. This will make upgrading to future versions much easier.

Here's a brief description of what's inside the supplied folders. (This document uses paths like "`TeX/Configs/`" to refer to various folders within the OzTeX folder. If you're not familiar with this notation you might want to skip ahead and have a quick look at section 4.2.)

- The `TeX` folder has subfolders with lots of files needed by OzTeX:
  - `TeX/Configs/` contains various configuration files, including the `Default` file which is loaded by OzTeX when it starts up. Section 4 discusses configuration files.
  - `TeX/Help/` contains a number of text files. The information in these files can be displayed by selecting their names from OzTeX's Help menu.
  - `TeX/Docs/Plain/` contains examples of Plain TeX files, including `nasty.tex` which is used in the guided tour.
  - `TeX/Docs/LaTeX/` contains `ozuser.tex`, the source file for this user guide, plus `hyper.tex` which shows how to use the `hyperref` and `color` packages.
  - `TeX/Inputs/` contains a large number of TeX input files stored in various subfolders. These files are needed to use Plain TeX, L<sup>A</sup>TeX, AMS-L<sup>A</sup>TeX, AMS-TeX and REVTeX.
  - `TeX/Formats/` contains format files read by TeX and the `TeX.pool` file read by INITEX. The three most popular formats are supplied: `Plain.fmt`, `LaTeX.fmt` and `AMSTeX.fmt`. Section 9 describes how to create format files.
  - `TeX/TFM/` contains many TeX font metric files stored in various subfolders. These TFM files are required by TeX to carry out typesetting; see section 10.1.
  - `TeX/VF/` contains virtual font files; see section 10.7.
  - `TeX/PK/` contains PK font files stored in various subfolders. See section 10.3 for more information about PK files. OzTeX is distributed with a fairly small set of PK files (sufficient to print out the documentation on most printers). New PK files at any desired size can be built on demand; see section 11.
  - `TeX/Encodings/` contains screen font encoding files; see section 10.6.

- `TeX/PS/` contains some PostScript files that can be sent to a PostScript printer.
- `TeX/Type1/` contains some PostScript Type 1 fonts that `dvips` can download.
- `TeX/DVIPS/` contains various files needed by `dvips`, Tom Rokicki’s popular DVI-to-PostScript translator. `OzTeX` contains an enhanced version of `dvips`. See section 14 for more information.
- `TeX/Unix/` contains some Unix scripts and programs that can be used when calling Unix commands; see section 5.
- The `MF` folder contains files needed by `OzMF`, a Mac implementation of Donald Knuth’s `METAFONT` program and other font related tools. Unless you’re a `METAFONT` programmer you probably won’t need to run `OzMF` yourself. `OzTeX` can automatically call `OzMF` when a new PK file is needed. See section 11 for more information about `OzMF`.
- The `MP` folder contains files needed by `OzMP`, a Mac implementation of John Hobby’s `MetaPost` program. See section 12 for more information about `OzMP`.
- The `TtH` folder contains files needed by `OzTtH`, a Mac implementation of Ian Hutchinson’s `TtH` program for converting `TeX` files to HTML. See section 13 for more information.
- The `Extras` folder contains miscellaneous items, mostly contributions from other people:
  - The `Docs` folder has additional documentation kindly provided by a number of different people. In the `Gentle Introduction` subfolder you’ll find `gentle.tex`, the source file for Michael Doob’s *A Gentle Introduction to TeX* and a precursor to his book *TeX Starting from 1* [3]. The `LaTeX Short Course` subfolder contains the source files for George Grätzer’s book *Math into L<sup>A</sup>T<sub>E</sub>X — An Introduction to L<sup>A</sup>T<sub>E</sub>X and AMS-L<sup>A</sup>T<sub>E</sub>X* [6]. Also included are a couple of documents written by Patrick Cousot and P. Alejandro López-Valencia on how to use fonts.
  - The `Y&Y` folder has support files for the fonts supplied by Y&Y.
  - The `PStoTTF` folder contains a `FontForge` script for converting PostScript fonts into TrueType format; see section 6.2 for why you might want to do this.
- The `Register` folder contains information on how to pay the shareware fee.

## 2 A guided tour of OzTeX

The aim of this section is to get you acquainted with OzTeX's major features.

### 2.1 Starting OzTeX

Assuming you've installed everything, double-click on the OzTeX application file to get the program started. After a brief pause you should see a window with the title "OzTeX" and a couple of messages showing the current version of OzTeX and its web site. OzTeX will also load the `Default` and `Local` configuration files, plus an encoding file called `Mac8r.enc` (referenced in the `Default` file). The OzTeX window is always present; it provides a terminal-like interface for displaying output and (less often) accepting user input. Text in the window can be copied to the Clipboard or saved to a file.

If you made a mistake during installation then you might get one or more error messages. The messages should give you some idea of how to fix the problem. If you made changes to the `Local` file then read section 4 for a detailed discussion of configuration files.

The production of a TeX document typically involves a number of iterations through the following cycle: edit, typeset, preview and print. (People concerned about our dwindling forests will hopefully preview many more times than they print.) Let's go through this cycle step by step.

### 2.2 Choosing an editor

OzTeX does not have an integrated text editor, but this isn't really a problem. There are plenty of TeX-savvy editors available for the Mac and OzTeX makes it easy to switch back to your preferred editor.

OzTeX and the other Oz\* applications always create text files with a creator code specified by the `text_editor` parameter. All the supplied `Default` config files set `text_editor` to `ALFA` (for Alpha). If you use a different editor then put a line like this near the top of your `Local` config file:

```
text_editor = R*ch      (for BEdit)
```

OzTeX also uses the specified editor as the target of an "open file" event which is sent when you select "Edit foo.tex" from the Edit menu. This item always shows the name of the most recently typeset file. See section 3.2 for more details about the items in the Edit menu.

### 2.3 Editing a TeX input file

A TeX input file is a standard Macintosh text file. We won't create an input file from scratch, instead we'll use one of the sample files provided in the `TeX/Docs/Plain/` folder. There is nothing special about this folder, or `TeX/Docs/LaTeX/`. I simply like to keep my TeX and LaTeX input files in separate folders. You can keep your input files anywhere you like. A good place would be in a new folder in the OzTeX folder (but not inside any of the supplied folders!).

The file we'll use is called `nasty.tex`. As the name suggests, it is not a typical TeX input file, but it does illustrate most of the things you'll encounter when using OzTeX in the future.

Switch to your text editor and open `nasty.tex` from the `TeX/Docs/Plain/` folder. The only change we'll make to `nasty.tex` is to add a deliberate error: insert the illegal command `\xxx` at the start of the file. Save this change before switching back to OzTeX.

## 2.4 Running TeX

Let's now typeset `nasty.tex`. Before starting TeX we need to make sure that the correct format will be used. All the available formats appear at the end of the TeX menu. Check this menu and make sure the Plain format is ticked; this is the format required by `nasty.tex`. (This check is not actually needed because the first line of `nasty.tex` contains “`%&plain`” which forces OzTeX to use the Plain format.)

Choose “TeX...” from the TeX menu and open `nasty.tex`. TeX will start up, load the Plain format, and begin reading the given file. When it sees the unknown command, TeX will display a suitable error message, beep, and wait for you to type something. Note that a solid block cursor sits next to TeX's “?” prompt. This block cursor always appears when TeX is waiting for you to type something. The OzTeX window should contain the following:

```
This is TeX, Version 3.14159 (no format preloaded)
**&plain nasty.tex
(nasty.tex
! Undefined control sequence.
1.5 \xxx

? █
```

Chapter 6 in *The TeXbook* [8] explains what you can do in such a situation. (If you type “e” then OzTeX will switch to your editor and open the file containing the error, possibly with the error line selected if your editor supports that option.) In this case we can simply ignore the error and continue by hitting the return key. Although `nasty.tex` is not very nice, you shouldn't see any more TeX errors. A 19-page DVI file called `nasty.dvi` should be created.

Have a look at the second item in both the File and View menus. Whenever a new DVI file is created, its name will automatically appear in these items so you can quickly print or view the DVI file without going through the open-file dialog. The second item in the TeX menu has also been updated with the name of the most recent input file (`nasty.tex` in this case), so you can easily typeset the same file again by pressing command-T. The file name also appears in the bottom item of the Edit menu so you can quickly switch back to your editor by pressing command-E.

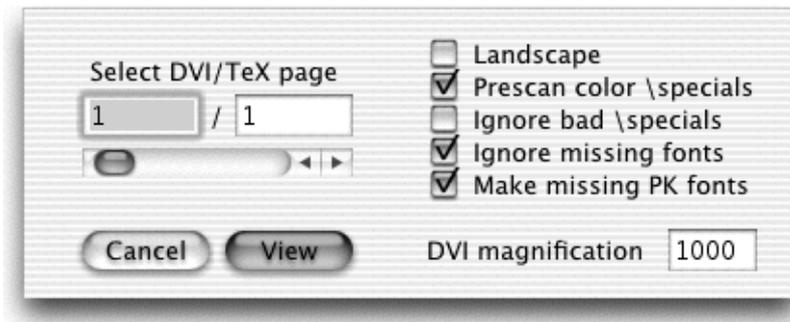
Another way to repeat a TeX run is to type “t” at any time during the first run. This is very handy for L<sup>A</sup>TeX users because two runs are often needed to resolve all cross references. Type “tt” to get two extra runs (any more than two “t”s will be ignored). Type “-t” to cancel the extra run(s). A TeX error or user interruption will also cancel any further runs. If you type “o” while TeX is running then OzTeX will view the DVI file at the end of the run. Type “-o” to cancel the view. Note that “t” and “o” can be typed in any order. The DVI file is only viewed after the last run, and only if that run is successful.

OzTeX supports background typesetting. At any time while TeX is running you can switch to another application and TeX will continue to run in the background.

At the end of every TeX run, OzTeX displays the time taken and the amount of memory allocated by TeX for its large arrays. The total number of bytes depends on the current TeX parameters. The values in the `Default` config file are suitable for most documents. For very large or complex documents you might need to increase one or more TeX parameters; you'll know this is necessary if TeX quits with a “capacity exceeded” error; see section 17.3. Depending on which parameters have to be increased you might also need to rebuild your formats; see section 9.

## 2.5 Viewing a DVI file

The next step is to preview the DVI file and check for problems that TeX might have missed, like missing fonts, bad page breaks, spelling mistakes, etc. Let's have a look at `nasty.dvi` by choosing the second item in the View menu. OzTeX creates a new window, called the “view” window, and sets its title to the selected DVI file. A dialog box has also appeared:



The scroll bar allows you to select any page in the DVI file. There are also two edit boxes in which you can type a DVI page number or TeX page number. (The dialog box also allows you to change some other options, but we'll ignore them at this stage; see the description of “View DVI...” in section 3.4 for details.) Play around with the scroll bar if you like but make sure the 2nd DVI page is selected before clicking the View button.

The 2nd page in `nasty.dvi` contains a single rule with some text below it. Now is a good time to adjust the size and location of the OzTeX and view windows to suit your screen. When you quit, OzTeX remembers the current window settings for use the next time it starts up.

Note that for DVI files opened from the Finder (or via any other application that can send OzTeX a suitable “open file” event) you can bypass the above view dialog by setting `bypass_view_dialog = true` in your Local file.

### 2.5.1 Setting the default view

OzTeX displays the page at a scale approximating the document's actual size (see below for more about scaling). This is the initial default view, but it might not suit the size or dimensions of your particular monitor. To change the default view, you just set up the desired scale and location using the various commands described below and then select “Save as Default View” from the View menu. OzTeX remembers your new default view until you decide to change it.

### 2.5.2 Paper coordinates

The coordinate system used by OzTeX to view (or print) a DVI page is based on units called “paper pixels”. The `viewing_resolution` parameter defines the number of paper pixels per inch when previewing. The `printing_resolution` parameter is used when printing a DVI file.

The paper pixel at (0,0) is exactly one inch in from the top and left edges of the paper. This position is referred to as the “TeX origin” because it is also the origin of the coordinate system used in DVI files. Every character, rule or `\special` has a specific location defined by a pair of paper pixels ( $h,v$ ). Vertical coordinates increase down the paper and horizontal coordinates increase to the right.

If the view window is frontmost then the cursor is changed to a cross whenever it moves over the contents region. The current position of the cross is shown in a box at the lower left corner of the view window. You can click in this box to change units; for example, set the units to `px` to see paper pixels.

### 2.5.3 Zooming in and out

A “scale factor” is used to display a DVI page at a particular size. It defines the number of paper pixels in each Macintosh screen pixel (both horizontally and vertically) and always has an integer value greater than or equal to 1. Certain View menu items can change the current scale factor:

- “Default View” restores the default view.
- “Full View” sets the scale factor to its maximum value and displays the entire page and paper edges in the middle of the view window.
- “Actual Size” sets the scale factor so that the new view will show the page at roughly the right size (it’s only approximate because the scale factor is an integer value).
- “Zoom In” decrements the current scale factor.
- “Zoom Out” increments the current scale factor.

You can also change the scale factor by clicking in the view window in a variety of ways:

- A simple click will temporarily magnify the area under the cursor until the mouse button is released. The scale factor in the magnified area is 1. The magnified area can be moved around and if the mouse moves outside the viewing area then scrolling will occur.
- Command-click to zoom in by decrementing the scale factor.
- Option-click to zoom out by incrementing the scale factor.
- Shift-click to zoom out by doubling the scale factor.
- Option-command-click to zoom in by halving the scale factor. However, if the mouse is dragged then OzTeX will zoom in to the selected rectangle (to cancel this operation, make the rectangle very thin).
- A beep occurs if you can’t zoom in or out any further.

When you zoom in or out by clicking in the page, OzTeX will try to make the point you clicked the middle of the new view. When you choose a menu item, OzTeX tries to keep the current mid-point in the middle of the new view. In both cases it is possible that the view may unexpectedly shift so that it remains within the scrolling limits set by OzTeX. Experiment with the various ways of zooming to see which methods you prefer.

The most precise display occurs when the scale factor is 1 because each screen pixel corresponds to exactly one paper pixel. OzTeX doesn’t allow you to zoom in any further than this.

### 2.5.4 Scrolling around

The view window has scroll bars that allow you to move over the page in the standard Macintosh manner. Note that the arrow keys can be used instead of clicking in the scroll arrows. OzTeX won’t let you get too far away from the page/paper boundaries. If you do manage to get lost just choose “Full View” or “Default View”.

### 2.5.5 Checking for errors

It is a good idea to select the “Page Info” item at least once while viewing a DVI file. This item displays its results in the OzTeX window. The display includes a list of all the fonts used in the DVI file and clearly indicates any that are missing. (If virtual fonts are used then “Page Info” won’t necessarily detect all missing PK fonts. A missing PK font referenced in a virtual font

will only be detected if a character requiring that PK font is used somewhere on the current page.) OzTeX also lists any `\special` commands on the current page, showing their locations and arguments.

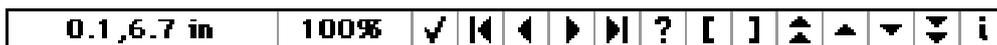
### 2.5.6 Selecting pages

There are a number of interesting pages in `nasty.dvi`. Use “Previous Page”, “Next Page” or “Go to Page...” to have a look at some of them. Note that the current DVI/TeX page numbers are displayed in the view window’s title bar, along with the total number of pages. Here are some points of interest:

- The current scale factor and page location will only change if the selected page is off the paper, in which case OzTeX will beep and display a full view. (You can prevent the switch to full view by setting `auto_full_view = false`.) Pages 13 to 15 illustrate this behavior. When you move from such a page to a normal page, OzTeX switches back to the default view.
- Page 3 has examples of `\special` commands that include graphic files. OzTeX is able to preview included files of type EPSF/PICT/PNTG; see section 16 for more information. The location of each `\special` is indicated by a small green marker. Note that the size of this marker does not change as you zoom in or out.
- Page 9 uses a couple of PostScript fonts; see section 10.4.
- All the fonts used on page 12 are deliberately missing.
- Page 16 has examples of bad `\special` commands.

### 2.5.7 Mouse shortcuts

The bottom left corner of the view window has a thin bar with small clickable icons so you can use the mouse to perform various actions:



From left to right you can: Change units; Change the scale; Restore the default view; Go to the first/previous/next/last page; Bring up the page selection dialog; Go back/forward one link; Scroll backwards rapidly/slowly; Scroll forwards slowly/rapidly; Display page information.

### 2.5.8 Keyboard shortcuts

OzTeX provides a number of keyboard shortcuts to make it easier to preview a DVI file:

- Type “o” to bypass the view dialog and open the most recently viewed DVI page (or page 1 if you haven’t viewed anything yet). OzTeX will also restore the scale factor and location you last used.
- Type “i” to see page info.
- Type “g” to bring up the page selection dialog.
- Type “<num>g” to go to DVI page <num>. “-1g” will go to the last page, “-2g” to the 2nd last page, etc. “99g” will also go to the last page if there are fewer than 99 pages.

- Type “<num>t” to go to TeX page <num>. “99t” will only go to TeX page 99 if it exists; if not, OzTeX beeps and the page selection dialog appears.
- Type “b” to move to the previous page.
- Type “n” to move to the next page.
- Type “[” to go back one link.
- Type “]” to go forward one link.
- Type “r” to restore the default view.
- Type “f” to change the scale to full view.
- Type “a” to change the scale to actual size.
- Type “h” or the home key to move to the first page.
- Type “e” or the end key to move to the last page.
- Hit the space bar or page down key to scroll forwards through a document.
- Hit the delete key or page up key to scroll backwards.
- Hit the return key to bring either the view window or the OzTeX window to the front.
- Type “w” to close the view window.

All the command-key items in the View menu have matching shortcuts. If you don’t like these keyboard assignments then it’s easy to change them; see section 4.7.

### 2.5.9 Switching to your editor

As has been mentioned, you can switch back to your editor by pressing command-E or by selecting the item at the bottom of the Edit menu. However, if your DVI file contains `\special` commands of the form “`src:<line><file>`” then you can control-click in the view window and OzTeX will locate the nearest `src \special` and tell your editor to select the specified line in the named file (normally your input file). The two editors most commonly used with OzTeX, Alpha and BBEedit, both support this feature.

One way to get `src \specials` included in your DVI file is to put this command in your `LaTeX` input file:

```
\usepackage{srcltx}
```

Another way is to run `TeX`’s `tex` or `latex` with the `-src-specials` option. By default, OzTeX’s previewer shows the location of each `src \special` with a small blue marker. The color and visibility of these markers can be changed via the `src_specials` parameter.

Note that OzTeX recognizes when a currently viewed DVI file has been changed by another application and will automatically update the display. This is handy for people who use OzTeX mainly as a DVI viewer and prefer to use a different front-end to run TeX.

When you have finished previewing, just close the view window. The view window is automatically closed whenever you run TeX or print a DVI file. If you load a configuration file when the view window is in front then it will close while the file is loaded and then reopen.

## 2.6 Printing a DVI file

Before printing a DVI file it is a good idea to select “Page Setup” from the File menu and make sure the paper size and orientation are correct. To print `nasty.dvi` choose either of the top two items in the File menu. The second item shows the current DVI file (if there is one).

OzTeX tries to support all Macintosh printers, but it also tries to take advantage of the sophisticated features available in PostScript printers. This has often meant going against Apple’s guidelines for device-independent printing, so OzTeX’s printing code has had a long and interesting history! All this is half-explanation and half-apology for what might seem to be an unnecessarily complicated printing strategy. OzTeX supports three printing methods:

### 2.6.1 Non-PostScript printing

This printing method is no longer needed on OS X.

### 2.6.2 Low-level PostScript printing

Low-level PostScript printing is only done if “Use Standard PostScript” is *not* ticked. It’s the fastest and most reliable way to print. The `send_ps` parameter specifies a Unix command to send PostScript data to a printer. The default `send_ps` value is `lpr` which tells OzTeX to send `.ps` files and `dvips` output to your default printer. To send output to a non-default printer, add a line like this to your Local file’s OzTeX-only section:

```
send_ps = "lpr -P printername"
```

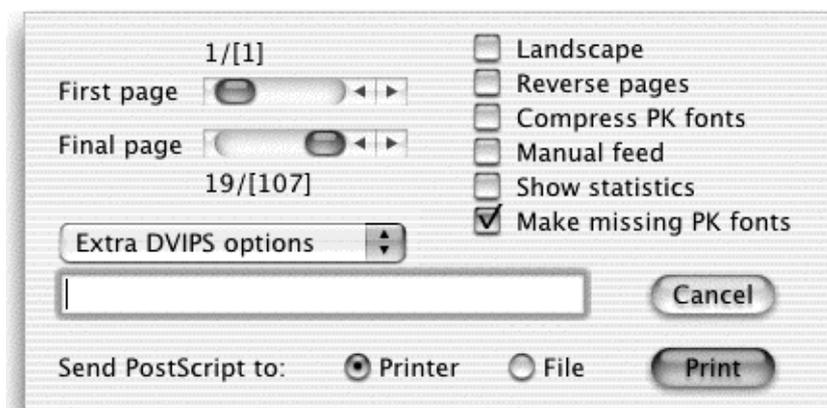
Substitute *printername* with the name of your printer. If you don’t know which printer is the default, or if you’re unsure about the name of your printer, type “`lpstat -d -p`” in Terminal.

If `lpr` doesn’t work for some reason, and your printer is connected via AppleTalk, then try a command like this:

```
send_ps = "atprint \"printername:LaserWriter\""
```

You can use the `atlookup` command in Terminal to generate a list of all known AppleTalk printers.

Having set `send_ps` to a suitable Unix command, OzTeX can print a DVI file by calling `dvips` and piping the PostScript output to the supplied command. OzTeX will start up `dvips` and display the following dialog:



This dialog box lets you change a variety of print options. The option you'll want to change most often is the page range. The two scroll bars control the first and final pages. OzTEX prevents you from choosing a first page greater than the final page. Play around with the scroll bars but restore their values so that every page will be printed.

A number of check boxes let you change the paper orientation (landscape or portrait), the page order (reverse or normal), whether or not PK fonts should be compressed (useful at high resolutions or when very large fonts are used), the paper source (manual feed or normal input tray), whether or not to show statistics (equivalent to `-d7`), and whether or not to make missing PK fonts. Any change to a check box option will be remembered the next time you print a DVI file, unless you decide to Cancel the dialog.

The pop-up menu makes it easy to add extra `dvips` options. Selecting an option from the menu will automatically append it to the current options. If an option requires a parameter then OzTEX selects the parameter template so you can type in the desired number, string, or whatever. You can of course ignore the pop-up menu and just type in your options. Details on all `dvips` options can be found in the DVIPS item in the Help menu.

Two radio buttons let you send the output to the printer or to a file. The initial destination is determined by the `to_printer` parameter. The `Default` file sets this flag to `true`, so if you prefer to send output to a file you can set `to_printer = false` in your `Local` file. For the purposes of this tour we want to use the printer, so click on the Print button.

You can't actually print every page in `nasty.dvi` because there is a deliberate error that makes this impossible. The first 11 pages should not cause any problems (assuming you're using A4 paper) but the rest will generate nearly every type of error message you're ever likely to encounter. You should eventually get up to page 17 where a deliberate PostScript error in a `\special` file will prevent any further pages being printed.

Note that you can bypass the above print dialog by setting `bypass_print_dialog = true` in your `Local` file, but this only works for DVI files selected for printing from the Finder (or from any other application that can send OzTEX a suitable "print file" event).

### 2.6.3 Standard PostScript printing

Note that this printing method is not supported on Leopard (OS 10.5).

If "Use Standard PostScript" is ticked then OzTEX will start up `dvips` and display the standard dialog for that printer. OzTEX-specific options are available, including an edit box for entering `dvips` options. When you're ready, click on the Print button to start printing.

Note that `dvips` has to go through all selected DVI pages first and determine font and character usage. All the PK character bitmaps and PostScript font files are then downloaded at the beginning, so the first page in a long document might take a while to be printed.

Although there are some advantages to using standard PostScript, such as being able to select printer-specific options like which input tray to use, there are some disadvantages. Standard PostScript code takes longer to print and Apple's print drivers are not all that reliable. My advice is to leave "Use Standard PostScript" unticked and use OzTEX's more efficient low-level printing method.

WARNING: If "Use Standard PostScript" is ticked then the `printing_resolution` value must match the resolution used by `dvips`, otherwise the printed output will be the wrong size. Yet another reason to avoid this method of printing.

### 2.6.4 Forcing PostScript printing

The "Force PostScript Printing" option is now always ticked and the item is disabled so you can't change it. This is done because CUPS allows PostScript to be printed on non-PostScript

devices, providing a couple of advantages:

1. Your DVI files can use all of the lovely PostScript `\special` tricks supported by `dvips` (rotation, etc). Any included `.eps` files will be properly rendered — no need to rely on the PICT 256 image.
2. It is very easy to switch between multiple printers — just create separate config files with different `send_ps` commands.

## 2.7 Positioning pages correctly

For your DVI output to appear in precisely the right location on the paper, the TeX origin must be exactly one inch in from the top and left paper edges. Page 2 in `nasty.dvi` contains a thick rule with its top left corner exactly at the TeX origin. Another good test file is `testpage.tex` in `TeX/Inputs/LaTeX/base/`.

To adjust the origin used by `dvips` you could add something like this to your `Local` file:

```
dvips_options = "$c -03mm,-2mm"
```

Note that `$c` is replaced by the current value of `dvips_options`; this is a handy technique for extending the `dvips` options. The `$c` is only special at the *start* of the `dvips_options` string.

If your printer is set up for US Letter paper then before adjusting the offsets you should first set the correct `paper_width` and `paper_height` values by doing this in your `Local` file:

```
load_config = "US Letter"
```

Note that the current paper dimensions *are* passed on to `dvips` via the `-T` option, but only if the `add_T_option` parameter is `true` (which it is in the `Default` file).

If you use more than one printer then you might need to create a separate config file for each printer, or you might need another config file for landscape printing. The best solution is to create a new folder within the OzTeX folder called, say, `myconfigs` and put all your own config files in that folder. Then in the `Local` file you can add these config files to OzTeX's `Config` menu by doing something like this:

```
config_folders = myconfigs/ $c
config_files   = "Printer A" "Printer B" $c
load_config    = "Printer A"           (preferred printer)
```

In the above example `$c` is replaced by the current list. `$c` can be used with any parameter that accepts a list of strings, usually as the first or last string.

## 2.8 Continuing from where you left off

That's the end of the guided tour. One more useful feature to note is that OzTeX remembers most of your settings in a preferences file called `OzTeX Prefs`. Some of the important settings include window sizes and locations, the ticked format, the current TeX file, the current DVI file and the most recently previewed page, including the scale and location for use by the "o" keyboard shortcut. The next time OzTeX starts up it will recall all this information so you can continue working from where you left off. (Double-clicking on a file to start up OzTeX will of course override some of this information.)

If you want to restore all settings to their initial values then delete or rename `OzTeX Prefs` so OzTeX can't find it. The next time OzTeX starts up it will create a new `OzTeX Prefs` file and initialize it with default settings. Note that `OzTeX Prefs` is created in `~/Library/Preferences`. `OzMF`, `OzMP` and `OzTtH` have their own `Prefs` files. If you like, you can move all `Oz* Prefs` files into the OzTeX folder; I prefer this because it simplifies my backup process. It's also recommended if you need to run two different versions of OzTeX on the same machine.

### 3 OzTeX's menus

Let's go through each menu and discuss each item in detail, concentrating on the things not mentioned in the guided tour.

#### 3.1 The File menu

##### Print DVI...

Prints a selected DVI file. Low-level PostScript printing is discussed in section 2.6.2. Standard PostScript printing is discussed in section 2.6.3. Hit command-dot or escape to cancel printing.

##### Print ?.dvi

Prints the indicated DVI file. The file name changes after you print/view a DVI file, or if a DVI file is created by TeX or a program in the Tools menu.

##### Print Recent

Prints the DVI file selected from the submenu.

##### Print Text...

Prints a selected text file. If OzTeX decides to use low-level PostScript printing then this file is sent after the `TEXTtoPS.pro` file which is kept in the `TeX/PS/` folder. The tab width is defined in that file. If OzTeX uses standard PostScript printing then you'll get a print dialog with an extra edit box that lets you specify the tab width. The default tab width is 3. Text files are printed using 10 pt Courier.

##### Page Setup...

The "Page Setup" dialog lets you select various printer-specific options that will be used in standard PostScript printing. If you select OK then the dialog settings are remembered for later OzTeX sessions. If a different printer is selected then the settings will be restored to default values for the new printer. OzTeX ignores the "Page Setup" settings when doing low-level PostScript printing.

##### Force PostScript Printing

This flag is now always ticked. See section 2.6.4 for more details.

##### Use Standard PostScript

Tick this flag if you want to use the standard Mac print dialogs when printing a DVI file or text file to a PostScript printer. (If the current printer is *not* a PostScript device then OzTeX ignores this flag.)

If "Use Standard PostScript" is ticked, then remember that "Page Setup" must be used to select the paper size and orientation before printing a DVI file or text file.

##### Send PostScript...

Sends a selected PostScript file to the current printer by piping the file's contents to a Unix command specified by the `send_ps` parameter (see section 2.6.2).

**Send ?.ps**

Sends the indicated PostScript file to the current printer.

**Send Recent**

Sends a recently used PostScript file to the current printer.

**Save Recent Files**

OzTeX can remember your TeX/DVI/PostScript files and provide quick access to them via submenus called “TeX Recent”, “Print Recent”, “View Recent” and “Send Recent”. Files will only be remembered if the “Save Recent Files” flag is ticked. Some people might want to leave it ticked all the time; others might want to turn it off after all of their favorite documents have been processed. Up to 999 files can be saved. The “Print Recent” and “View Recent” submenus are automatically kept synchronized.

**Show Paths**

If this flag is ticked then the recent-file submenus show an alphabetical list of full or partial path names (the latter are used for files anywhere within your OzTeX folder). If the flag is unticked then the submenus show simple file names, with duplicate names followed by their folder paths.

**Check Recent Files**

Checks the existence of all files listed in the recent-file submenus. If one or more files don't exist then you'll be asked if you want to remove the corresponding items.

You can also remove any item by holding down the command key before clicking in the menu bar. The submenu names change to “Remove Item” to reflect what will happen. To remove *all* items just select the first item in the submenu; as a precaution you'll be asked if that's what you really want to do.

**Save All Text . . .**

Saves the entire contents of the OzTeX window to a given text file. The `Default` config file sets the file name to `Oz.text`. This item is disabled if the OzTeX window is empty or not in front.

**Delete**

Lets you delete files ending in a given string selected from a submenu. The submenu items are specified by the `delete_files` parameter. The open-file dialog has a “Do all files” button that lets you delete all matching files in the selected folder. You won't be able to recover deleted files so use this feature with caution!

**Quit**

Quits OzTeX.

### 3.2 The Edit menu

OzTeX does not have an integrated text editor, so most of the standard editing items are normally disabled. You can however use the Copy item to copy selected text from the OzTeX window to the Clipboard.

**Clear All Text**

Clears all text in the OzTeX window. The text is also cleared automatically if you ever manage to generate 32,000 lines in a session.

**Auto Clear**

If this item is ticked then the OzTeX window is automatically cleared every time a new task is begun. If unticked then the window is not cleared and tasks are separated by a line of asterisks.

**Copy All Text**

Copies the entire contents of the OzTeX window to the Clipboard. Note that you don't have to select all the text first. This item is disabled if the OzTeX window is empty or not in front.

**Font**

Selects the screen font used to display text in the OzTeX window. You can choose from any of the currently installed fonts, but for best results it is advisable to use a fixed-width font like Monaco or Courier. This item is disabled if the OzTeX window is not in front.

**Size**

Selects the point size of the font used in the OzTeX window. This item is disabled if the OzTeX window is not in front.

**Color . . .**

Lets you change the colors used in the OzTeX window. This item is disabled if the OzTeX window is not in front.

**Text Smoothing**

If ticked then anti-aliasing is used to display text in all windows (including the view window if the font size is above 24 pt).

**Edit ?.tex**

If selected, OzTeX will send an "open file" event to your preferred text editor. OzTeX will launch the editor if necessary. This item normally contains the name of the most recently typeset file (it will always match the file shown in the TeX menu's 2nd item).

**3.3 The TeX menu****TeX . . .**

Typesets a selected input file using the currently ticked format, or a format determined by looking in the input file (see below). One of the nicest things about TeX is that it behaves the same way on a large range of different computer systems. Virtually everything you read in *The TeXbook* [8] will apply to the version of TeX run by OzTeX, and the same goes for Leslie Lamport's *L<sup>A</sup>TeX book* [11] if you use that format. A few extra features have been added in OzTeX:

- $\TeX$  looks for “`%&format`” on the first line of a selected input file and, if present, uses the given format regardless of the currently ticked format in the  $\TeX$  menu. Or if there is no such comment and the “Auto Detect Format” flag is ticked then OzTeX tries to determine the required format based on the first few commands in the file; see the description of the “Auto Detect Format” item below for details on how OzTeX does this.
- Input files can have either Mac or Unix line endings.
- Most of  $\TeX$ 's capacity parameters are set at run-time in a configuration file rather than at compile-time. If you get a “capacity exceeded” error then you should be able to overcome the problem by increasing the appropriate parameter; see section 17.3.
- OzTeX allows the use of Unix or Mac path names to include files in your  $\TeX$ / $\LaTeX$  input files; see section 4.2.4.
- If an included file doesn't have an explicit path then  $\TeX$  will look for it in the current folder first, then in the folders specified by the `input_folders` parameter. Similarly, when looking for TFM files,  $\TeX$  will look in the current folder first and then in the folders specified by the `tfm_folders` parameter.
- If  $\TeX$  is asked to read the file `foo` (via `\input`, `\openin` or the  $\TeX$  menu) and `foo.tex` is not found then OzTeX looks for `foo` with no extension. Most  $\TeX$  implementations support extensionless files — handy for verbatim file typesetting.
- All text files created by OzTeX will have your `text_editor` value as their creator code, so double-clicking on a `.log/.aux` file will open it in the specified editor.
- Set `write_8_bit = true` in your Local file if you want  $\TeX$  to output 8-bit characters instead of using the `^^hh` notation. Note that the `write_8_bit` flag is only tested by INITEX so you'll need to rebuild the format file(s) after changing it to `true`.
- You can interrupt a  $\TeX$  session at any time by typing command-dot or escape. Depending on what it is currently doing,  $\TeX$  usually responds immediately with a suitable message and the “?” prompt. If you hit command-dot or escape at this stage (or whenever the block cursor is visible) then  $\TeX$  will immediately abort.
- $\TeX$  will continue typesetting in the background if you switch to another application.

### **$\TeX$ ?.tex**

Typesets the indicated input file using the currently ticked format, or a format determined by looking in the input file (see below). OzTeX remembers the most recently typeset file.

### **$\TeX$ Recent**

Typesets the input file selected from the submenu.

### **Auto Detect Format**

If this item is ticked, and there is no “`%&format`” comment on the first line of the input file, then OzTeX tries to determine the required format by looking at the first few commands in the file. If `\documentstyle` or `\documentclass` is found then OzTeX uses `LATEX.fmt`. If “`\input amstex`” occurs early enough then OzTeX uses `AMSTEX.fmt`. Any comment lines or blank lines at the start of the file are ignored. If it can't determine the format then OzTeX will use the currently ticked format, so it's probably best to tick Plain if you decide to use this option.

## INITEX

Runs INITEX, a special version of T<sub>E</sub>X normally used to create format files; see section 9.

### List of formats

The items below INITEX are specified by the `format_files` parameter. The `Default` config file sets the formats to `LaTeX`, `AMSTeX` and `Plain`.

The ticked format will be the one used when you typeset a file, although this will be overridden if the first line of the file starts with “`%&format`”, or if the “Auto Detect Format” item is ticked and OzT<sub>E</sub>X is able to detect the required format (see above). To change the ticked format, simply select a different format item. OzT<sub>E</sub>X remembers the ticked format and will only change it in the unlikely event that you load a config file containing a `format_files` list that does not include your ticked format; if this happens then the first format in the list will be ticked.

## 3.4 The View menu

### View DVI...

Previews a selected DVI file. OzT<sub>E</sub>X will open the view window and set its title to the name of the selected DVI file. You'll then be presented with a dialog box (see page 5) that lets you choose various viewing options:

- The initial page to display. A scroll bar allows you to locate any page in the DVI file. The left and right arrow keys can be used to move the thumb box. You can also type in a specific DVI page or T<sub>E</sub>X page; if the given number is valid then the other page number is updated and the scroll bar repositioned. If there is only one page in the DVI file then the scroll bar will be inactive.
- The paper orientation (landscape or portrait). Any change to this option will be remembered the next time you view a DVI file, unless you Cancel the dialog. The check box setting will also be used in the dialog for low-level PostScript printing.
- Prescanning for color `\specials`. You only need to enable this option if your document uses color; see section 16.6.
- The next two check boxes can be used to tell OzT<sub>E</sub>X to ignore bad `\special` commands and/or missing fonts. It is a good idea to select both these check boxes when previewing a DVI file created by another T<sub>E</sub>X system, otherwise you might get a lot of annoying warning messages. OzT<sub>E</sub>X remembers these check box settings.

The previewer recognizes all `dvips`-specific `\special` commands and silently ignores those it can't handle (like rotation and other commands that use raw PostScript code). Ditto for `tpic` `\specials`. This means that most people should never need to tick the “Ignore bad `\specials`” option.

- The last check box can be used to make missing PK fonts (but only if “Ignore missing fonts” is unticked). OzT<sub>E</sub>X remembers the check box setting. See section 11 for more details.
- The DVI magnification (expressed as an integer 1000 times the desired magnification). You should only alter the default value if you plan to print the DVI file at a different magnification. See Chapters 4 and 10 in *The T<sub>E</sub>Xbook* [8] for details. A document using only PostScript fonts can be viewed/printed at almost any magnification (OzT<sub>E</sub>X allows values from 1 to 10000). OzT<sub>E</sub>X does not remember changes to the DVI magnification.

If you Cancel the dialog then OzTeX will close the view window. If you select the View button then the current DVI/TeX page numbers, and the total number of pages, will be appended to the DVI file name in the view window's title bar. OzTeX will locate the requested page and display it. The initial scale factor and page location will be set to values determined by the default view (see section 2.5.1).

### View ?.dvi

Previews the indicated DVI file. Exactly the same sequence of events described above will occur. The file name changes after you print/view a DVI file, and after a DVI file is created by TeX or a program in the Tools menu.

### View Recent

Previews the DVI file selected from the submenu.

### Show OzTeX or Show View

Brings either the OzTeX window or the view window (if open) to the front. The return key is the default keyboard shortcut for this menu item.

### Page Info

Displays information about the current DVI page in the OzTeX window, bringing it to the front if necessary. This item is disabled if the view window is closed. The information displayed includes:

- The ten TeX page counters stored with the current page (trailing counters with zero values are not shown).
- The viewing resolution, screen resolution, DVI magnification and paper dimensions.
- A list of all the fonts used in the entire DVI file. For each font appearing on the current page OzTeX will show the total number of characters used. Note that the order of fonts may change as you move from page to page; OzTeX sorts the list so that all fonts actually used on the current page appear first. A PostScript font is indicated by its corresponding TFM path name and the requested point size. A PK font is indicated by a PK path name. Missing PK or TFM files are flagged by the message "DOES NOT EXIST". Missing screen fonts are flagged by the message "NO SCREEN FONT".
- A list of all virtual fonts used in the DVI file. Those used on the current page are flagged by a comment.
- The number of rules on the current page.
- The location and argument of each \special command on the page. The location is expressed in terms of OzTeX's paper coordinate system; see section 2.5.2. If the \special command includes a valid EPSF/PICT/PNTG file then the width and height of the image's bounding box is also displayed.

Note that if the DVI file uses virtual fonts then "Page Info" won't necessarily detect all missing PK files. A missing PK file referenced in a virtual font will only be detected if a character requiring that PK file is used somewhere on the current page. Use dvips and create a .ps file if you want to detect all missing PK fonts.

**Previous Page**

Displays the previous DVI page. This item is disabled if the view window is inactive or currently displaying the first page.

**Next Page**

Displays the next DVI page. This item is disabled if the view window is inactive or currently displaying the last page.

**Go to Page . . .**

Brings up a dialog box allowing you to select any DVI page for display. This item is disabled if the view window is inactive or if the DVI file only has one page.

**Back Link**

Takes you back one HyperTeX link; see section 16.7.

**Forward Link**

Takes you forward one HyperTeX link; see section 16.7.

**Default View**

Restores your preferred scale factor and location. This item is disabled if the view window is inactive or already displaying the default view.

**Full View**

Changes the scale factor and location so that the entire DVI page and paper edges are displayed in the middle of the current view window. The scroll bars are disabled and the scale factor is set to its maximum value. This item is disabled if the view window is inactive or already displaying a full view.

**Actual Size**

Changes the scale factor so that the DVI page is displayed as near as possible to its actual size. OzTeX will try to keep the current mid-point in the middle of the new view, but shifting might be necessary to keep the page within the scrolling limits. This item is disabled if the view window is inactive or already displaying the page at its actual size.

The scale factor is actually set to the nearest integer equal to the viewing resolution divided by the Mac screen resolution. A likely calculation is  $300/72 = 4.17$ , so the scale factor is set to 4. Because of this approximation, a 3 in wide rule in your TeX input file won't appear exactly 3 in wide in the view window. If this is a serious problem then set the viewing resolution to some multiple of the screen resolution. Note that  $360 = 72 \times 5$ .

**Zoom In**

Decrements the scale factor. OzTeX will try to keep the current mid-point in the middle of the new view. This item is disabled if the view window is inactive or already at minimum scale.

**Zoom Out**

Increments the scale factor. OzTEX will try to keep the current mid-point in the middle of the new view. This item is disabled if the view window is inactive or already at maximum scale.

**Save as Default View**

Saves the current scale factor and page location as the default view (see section 2.5.1). This item is disabled if the view window is inactive.

**Font Darkness**

Lets you adjust the darkness level of displayed characters. Outline and PK fonts can be adjusted separately.

**3.5 The Tools menu**

NOTE: If the option key is held down before selecting an item in the Tools menu then OzTEX will bring up the open-file dialog so you can select a file rather than use the current file.

**DVIPS ?.dvi**

This item is provided so you can always create a .ps file even if the current printer is not a PostScript printer. When `dvips` is called via one of the Print items in the File menu then the default destination is the current printer, but if called from the Tools menu then the default destination is a file.

Simply click on the Save button to create `foo.ps` in the same folder as `foo.dvi`. Option-click on the Save button to bring up the save-file dialog and save the output anywhere you like.

**DVIDVI ?.dvi**

The DVIDVI item can be used to carry out pagination tricks on the given DVI file. For more information, select the DVIDVI item in OzTEX's Help menu.

DVIDVI will look in `help_folders` for a file named "DVIDVI examples". If found, its contents will be displayed just before the Options prompt so you can cut and paste the examples.

**DVIcopy ?.dvi**

Replaces all virtual fonts in the given DVI file with actual fonts. See section 10.7 for information about virtual fonts. Most people won't need to use DVIcopy because OzTEX automatically resolves all references to virtual fonts. If you decide to use DVIcopy then you need to be aware of a potential problem; see the DVIcopy item in OzTEX's Help menu for details.

**PSbook ?.ps**

Rearranges pages from a PostScript file into "signatures" for printing books or booklets.

**PSnup ?.ps**

Reads a given PostScript file and creates a new PostScript file with multiple logical pages on each physical sheet of paper.

**PSselect ?.ps**

Selects pages from a PostScript file, creating a new PostScript file.

**PStoPS ?.ps**

Rearranges pages from a PostScript file, creating a new PostScript file.

The above PS tools are Mac implementations of Angus Duggan's utilities for manipulating PostScript files (usually created by `dvips`). All the tools can process files with Mac or Unix line endings. For more information about each tool select the corresponding item in the Help menu.

Each tool will look in `help_folders` for a file named "`<tool> examples`". If found, the file's contents will be displayed in the *OzTeX* window just before the `Options` prompt so you can cut and paste the examples.

New `.ps` files created by `dvips` and the PS tools will have a creator code specified by the `ps_creator` parameter. The `Default` file sets `ps_creator = gsVR` so that double-clicked `.ps` files will be opened by MacGSView, a free Mac port of Ghostscript which can preview PostScript files.

By default, DVIDVI and the PS tools will create an output file with the same name as the input file. They will also repeat the `Options` prompt if an error is detected in the given options, so there is no need to select the input and output files again.

You can use the `extra_tools` parameter to provide quick and convenient access to other applications by adding items at the end of the Tools menu. For details on how to use `extra_tools` see section 4.8. The `Default` config file adds a number of useful items.

**3.6 The Config menu****Show**

Shows the current values of configuration parameters in the *OzTeX* window. You can choose to show all parameters or only certain subsets. The output is displayed with the correct syntax for a configuration file; this makes it easy to copy lines from the *OzTeX* window and paste them into the `Local` config file where you can make the desired changes.

**Edit Local**

Opens the `Local` configuration file in your preferred text editor. After making changes to any parameter(s) remember to save the file, and after switching back to *OzTeX* make sure you select the `Default` item (see below); this will reload the `Local` file and enable the new parameter settings.

**Load Config...**

Lets you select and load any configuration file without having to add it to the `config_files` list. This means you don't have to clutter up your Config menu with rarely used config files.

**Default**

Loads the `Default` configuration file, resetting all parameters to their default values, then loads the `Local` config file (if it exists). You should select this item after making changes to the `Local` file, or whenever you want to restore *OzTeX* to a "clean" state without having to restart the application.

### The supplied configuration files

The remaining items in the Config menu are determined by the `config_files` parameter. Items without any Unix or Mac path prefix must correspond to files stored in folders specified by the `config_folders` parameter (the default location is `TeX/Configs/`). Selecting an item causes the corresponding configuration file to be loaded. A tick appears next to the most recently loaded file. Here is a brief description of some of the more useful config files supplied with OzTeX:

**Linotronic** configures OzTeX for a Linotronic typesetter. The printing resolution is set to 1270 and the mode to `linohi`.

**600 dpi Printer** configures OzTeX for a 600 dpi printer. The printing resolution is set to 600 and the mode to `ljfour`. This mode is suitable for a HP LaserJet 4MP. Note that the `TeX/Configs/` folder contains other config files with suitable settings for a variety of printers; see the comments in the `Local-Notes` file.

**A5 Portrait** sets the paper dimensions for A5 paper in portrait mode.

**A4 Landscape** sets the paper dimensions for A4 paper in landscape mode.

**US Letter** sets the paper dimensions for US Letter paper in portrait mode.

**Show All Specials** shows all `\special` markers when previewing.

**Hide All Specials** hides all `\special` markers when previewing.

**Big Magnifier** magnifies the entire view when you click within the view window.

**View at 720 dpi** sets the viewing resolution to 720 dpi. This increases the number of magnification scales, but previewing will be a bit slower and use more memory. Best used in conjunction with outline fonts (see below) rather than PK fonts.

**Add Outline Fonts** updates the current list of outline fonts so that OzTeX will use TrueType versions of the CM and AMS fonts.

Section 4.6 describes how to create your own config files and add them to the Config menu.

### 3.7 The Help menu

The items appearing in this menu are specified by the `help_files` parameter. An item without any path prefix should correspond to the name of a text file in one of the folders specified by the `help_folders` parameter. (Note that the default help files for each Oz\* application are stored in the Help subfolders inside the corresponding TeX, MF, MP and TtH folders.) Selecting an item from the Help menu simply causes the contents of the corresponding file to appear in the OzTeX window.

Feel free to add your own items to the Help menu. If you create a new help file then just add its path to the `help_files` list in your Local file:

```
help_files = "myhelp/Good Tips" $c
```

Or if you'd prefer just the name to appear in the Help menu, do this:

```
help_folders = myhelp/ $c
help_files = "Good Tips" $c
```

If you do create new help files then avoid using tabs.

## 4 Configuration files

Oz $\TeX$  and the other Oz\* applications can all be customized by changing parameter values in configuration files. The supplied config files for each Oz\* application can be found in the `Configs` subfolders inside the corresponding `TeX`, `MF`, `MP` and `TtH` folders.

### 4.1 The required syntax

A configuration file is a simple text file. Follow these rules when editing a config file:

- Blank lines are ignored. Lines starting with “%” are also ignored and can be used to include comments. This is also true for lines starting with “@”, but in this case the rest of the line is displayed in the application’s main window (useful for debugging).
- All other lines are parameters of the form “keyword = value”. Keywords are case insensitive. Spaces or tabs around a keyword are ignored. The same keyword can appear more than once in a config file; later values override earlier ones.
- The “=” character is optional, but I recommend you put it in. Any spaces or tabs around “=” are ignored. The parameter value consists of one or more strings. If only one string is required then the rest of the line is ignored. Multiple strings are separated by spaces or tabs, so use double quotes around a string that includes spaces. To include a double quote in a string use “\””. To include “|” in a string use “\|”. Use “\\” to include “\” if it is followed by “n” or “|” or “\”.
- To avoid very long lines, use “|” to split lines anywhere you like. Upon reading a “|”, the application will ignore the rest of the current line and continue reading from the next line. The best place to use “|” is where a space is allowed; that way you can safely indent the next line to make the file more readable.
- A line starting with “!” tells the application to stop reading the config file. This lets you add comments without having to start each line with “%”.

Note that the syntax is the same for all Oz\* applications, but each one recognizes a different set of keywords.

### 4.2 Path names

Some parameters are used to tell the Oz\* applications where to find various types of files. The parameter values consist of one or more strings in the form of path names. Both Unix-style paths and Mac-style paths are supported, but Unix paths are preferred because they are the native syntax on OS X and provide better compatibility with  $\text{te}\TeX$  tools. Unix and Mac path names are both case insensitive (assuming you are using a HFS or HFS+ file system). When loading a config file, each application will generate a warning message if a given path name does not specify a known folder or file.

Path names must be enclosed in double quotes if they contain any spaces, as in the Unix path “my config files/” or its equivalent Mac path “:my config files:”. These are examples of partial paths. When used in a config file they specify a location relative to the Oz\* application. (When used in a  $\text{TeX}$ ,  $\text{METAFONT}$  or  $\text{MetaPost}$  input file they are relative to the location of the input file; see section 4.2.4.) Partial paths are the best way to specify folders and files because then you can move or rename the Oz $\TeX$  folder without having to change any path names. Full paths include a volume name (either explicit or implied) and should be avoided because they’re more likely to stop working if you move or rename folders.

### 4.2.1 Unix path syntax

In Unix, a folder is usually called a directory, but most of the time I'll stick to the word folder. A Unix path name is a sequence of folder names separated by slashes. In standard Unix a path that doesn't end with a slash can refer to a file or folder. To avoid ambiguity, the Oz\* apps require that a path to a folder must end with a slash; e.g., `TeX/TFM/`. This is an example of a partial Unix path; an equivalent notation you might also see is `./TeX/TFM/`. A partial path starting with `../` can be used to go up to the parent folder, and `../..` would go up yet another folder, etc. A full Unix path starts with a slash. For example:

```
/Applications/      - the boot volume is not specified
/volumes/hd/foo/    - other volumes are in /Volumes/
```

Here are two equivalent Unix paths:

```
~/Documents/tex/foo.tex
$HOME/Documents/tex/foo.tex
```

Note that a `~/..` path can only be used in a  $\TeX$  input file if the `~` character's catcode value is changed, so it is easier to use `$HOME/..`. If a Unix path starts with `$XXX/..` then `XXX` is assumed to be an environment variable and `$XXX` will be replaced by its value. Environment variable names are case sensitive, so `$home/..` won't work.

WARNING: Avoid creating files or folders with `/` in their names. A  $\LaTeX$  command like `\include{foo/bar}` will not find a file called `foo/bar.tex` because Oz $\TeX$  will assume it is a partial Unix path to a file called `bar.tex` in a subfolder called `foo`.

### 4.2.2 Mac path syntax

A Mac path name is a sequence of folder names separated by colons. If the path name ends with a colon then it specifies a folder, otherwise a file.

A full Mac path begins with the name of a volume; e.g., `HD:OzTeX:TeX:PS:.` A partial Mac path begins with a colon; e.g., `:TeX:PS:.` The first folder in a partial path is assumed to be in the same location as the application. It is also possible to use a partial path to refer to a folder outside the Oz $\TeX$  folder; e.g., `:mydocs:` specifies a folder located alongside the Oz $\TeX$  folder. The `..` prefix means go up one folder, and `...` means go up two folders, etc., but it is difficult to think of an example where you'd ever need to use more than two colons.

### 4.2.3 Symbolic links and aliases

An alias is the traditional Mac way of creating a pointer to a file or folder. The Oz\* applications only support aliased files, not aliased folders. In a path like `mytex/docs/foo.tex` only the last element in the path, `foo.tex`, can be an alias.

A symbolic link is the standard Unix way of creating a pointer to a file or folder. Symbolic links on OS X are supported by all Oz\* applications and any part of a Mac or Unix path can be a link, not just the ending file. Consider these (equivalent) examples:

```
\includegraphics{myfigs/foo.eps}
\includegraphics{:myfigs:foo.eps}
```

Both paths will work on OS X if `myfigs` and/or `foo.eps` are symbolic links, but only `foo.eps` can be an alias. Note that symbolic links aren't quite as flexible as aliases. If you move or rename the original item then the link will no longer work.

Creating a symbolic link is easy. In Terminal you can type a command like this:

```
ln -s existing-file-or-directory link-name
```

If typing commands into Terminal is too much like hard work then download Nick Zitzmann's SymbolicLinker from <http://seiryu.home.comcast.net/symboliclinker.html>. This is a contextual menu plugin that lets you create symbolic links to files or folders by control-clicking on them in the Finder.

#### 4.2.4 Path names in input files

The Oz\* applications also allow the use of Unix-style path names to include files in your T<sub>E</sub>X/METAFONT/MetaPost input files. (Mac-style paths are also supported but should be avoided if you want compatibility with t<sub>E</sub>X.) For example, in a plain T<sub>E</sub>X document you can do things like:

```
\input /Users/joe/Documents/macros/mydefs.tex
```

This is an example of a full path (it starts with a slash), but it is generally a better idea to use partial paths. For example, in a L<sup>A</sup>T<sub>E</sub>X document you could do this:

```
\include{chapter1/ch1}
\includegraphics{figures/fig1.eps}
```

Partial paths are relative to the location of the main input file; i.e., the initial file opened by OzT<sub>E</sub>X. Remember that partial paths in configuration files are relative to the location of the OzT<sub>E</sub>X application.

The use of Unix or Mac path names in input files is something you should try to avoid — it will make your documents much less portable. You can minimize this problem by using macros to isolate all system-specific paths in one file. For example, on a Mac you could create a file called, say, `paths.tex` that would contain macro definitions like this:

```
\def\chone{chapter1/}
\def\chtwo{chapter2/}
...
\def\figs{figures/}
```

Your input files can then use commands like this (assuming L<sup>A</sup>T<sub>E</sub>X):

```
\include{paths}
\include{\chone ch1}
\includegraphics{\figs fig1.eps}
```

If you need to move your document to another system then all you have to do is change the definitions in `paths.tex`.

Another approach is to completely avoid the use of path names. This can be done by specifying the locations of all included files in T<sub>E</sub>X's search path. Most T<sub>E</sub>X systems make this easy to do; in OzT<sub>E</sub>X's case you could put something like

```
input_folders = mybook/* $c
```

in your `Local` file. All you need to do is make sure that all included files have unique names. However, this approach is not as efficient as using macros, especially for large projects that might require T<sub>E</sub>X to search dozens/hundreds of folders.

### 4.3 Search paths and subfolder expansion

Some of the most important config file parameters are those that tell the Oz\* apps which folders to search for various types of files. All of the following parameters accept a list of strings where each string is a full or partial path name for an existing folder. All Oz\* apps use these two parameters:

```
config_folders  - to find config files
help_folders    - to find help files
```

Oz $\TeX$  uses these parameters:

```
format_folders - used by  $\TeX$  to find *.fmt files
input_folders  - used by  $\TeX$  to find input files: *.tex/*.sty/*.fd/EPST/ETC.
                also used by previewer/dvips to find EPSF/PICT/PNTG files
tfm_folders    - used by  $\TeX$ /previewer/dvips to find *.tfm files
vf_folders     - used by previewer/dvips to find *.vf files
enc_folders    - used by previewer to find screen font encoding files
ps_folders     - used by dvips to find various input files:
                config.*/psfonts.*/*.map/*.pro/*.enc/*.pfa/*.pfb/LWFN
```

NOTE: The previewer and dvips use the pk\_files parameter to find PK files. This parameter is a list of file templates; see section 10.3.

OzMF uses these parameters:

```
base_folders   - used by METAFONT to find *.base files
input_folders  - used by METAFONT to find *.mf files
tfm_folders    - used by VFtoVP to find *.tfm files
ps_folders     - used by AFM2TFM to find *.enc files
```

OzMP uses these parameters:

```
mem_folders    - used by MetaPost to find *.mem files
input_folders  - used by MetaPost to find *.mp files
tfm_folders    - used by DVItOMP to find *.tfm files
vf_folders     - used by DVItOMP to find *.vf files
```

OzTtH uses this parameter:

```
input_folders  - used by TtH to find input files
```

In almost all cases the current folder will be searched first before using the above parameters. The only exceptions are config\_folders, help\_folders and enc\_folders.

The Oz\* apps can also be told to include subfolders by using special characters at the end of a path name. Let's look at how Oz $\TeX$  would handle these three examples:

```
input_folders = mytex/inputs/*
input_folders = mytex/inputs/**
input_folders = mytex/inputs/#
```

In the first example Oz $\TeX$  will include subfolders one level deep. In the second example Oz $\TeX$  will include subfolders up to two levels deep; i.e., the number of asterisks specifies the maximum depth. In the third example Oz $\TeX$  will include *all* subfolders. Note that in all cases Oz $\TeX$  will only include a folder if it contains at least one file.

WARNING: Adding special characters to include subfolders can cause config file loading to become very slow, especially when using `#`. This is because every item in every folder has to be examined to see if it is a file or folder. My advice is to stick to single asterisks as much as possible. The best way to use multiple asterisks or `#` is to use them only to generate a search list; you can then select “Show Files and Folders” from the Config menu to display the list so it can be copied and pasted back into your config file (and then use asterisks to reduce duplication). This is the approach I used to create the `Use texmf tree` config file located in `TeX/Configs/`.

#### 4.4 The Default configuration files

The Oz\* applications all read a config file called `Default` when starting up. Each application has its own `Default` file; OzTeX reads `TeX/Configs/Default`, OzMF reads `MF/Configs/Default`, etc. The supplied `Default` files contain examples of all possible keywords. They also contain detailed notes (at the end of each file) discussing how the important parameters are used and other handy tips.

It is strongly recommended that you do not change any of the supplied `Default` files. Make all changes to the `Local` file instead; that will make life much easier when upgrading to future versions of OzTeX. Note that the bottom item of the Help menu in each Oz\* application contains the path to the corresponding `Default` file. Selecting this item provides a quick way to see examples of all parameters and to read the important notes. It’s also a much safer way than using an editor to look at the `Default` files because you can’t make any changes (you can still copy stuff and paste it into your `Local` file).

#### 4.5 The Local configuration file

After loading their respective `Default` files, all Oz\* applications load a `Local` config file if it exists in the OzTeX folder. This file is where you make changes to the default settings. The supplied `Local-Notes` file can be used as a template to create your `Local` file. `Local-Notes` doesn’t actually do anything, but it does contain important documentation so make sure you read it. It includes examples showing how to use `$c` to make incremental changes to any parameter that accepts a list of strings (very handy for extending search paths). Also check out the use of `current_app` to control the visibility of keywords; this allows each Oz\* application to read the same config file.

Note that the Config menu in each Oz\* application has an “Edit Local” item so you can quickly open the `Local` file in your preferred text editor. If you make any changes then remember to save the file before switching back to the Oz\* app, and then select the `Default` item in the Config menu to load the new settings.

#### 4.6 Creating your own configuration files

The Oz\* applications all have a Config menu that lets you switch rapidly from one configuration file to another. In OzTeX’s case, this is especially useful if you have access to more than one printer because each printer will probably require a separate config file. There are many other uses for config files; section 3.6 has brief descriptions of some of OzTeX’s supplied config files.

It is a simple matter to create your own config files and add their names to the `config_files` list in your `Local` file. To make future upgrading easier it is best to put all your own config files in a new folder called something like `myconfigs` and then put a line like this in the OzTeX-only section of your `Local` file:

```
config_folders = myconfigs/ $c
```

This tells OzTeX to look for config files in `myconfigs` before looking in the default location (`TeX/Configs/`). To add a new item to OzTeX's Config menu use a line like this:

```
config_files = "add more fonts" $c
```

This will add your file to the top of the list; if you'd prefer it at the bottom then put the file after the `$c`. Of course it only makes sense to add new Config menu items if you need to change some parameters temporarily. If you want the changes to be permanent then do this:

```
load_config = "add more fonts"
```

Both `config_files` and `load_config` allow the use of paths to specify the explicit location of a config file; for example:

```
load_config = "myconfigs/add more fonts"
```

If a path is provided then OzTeX does not search `config_folders`.

## 4.7 Changing keyboard shortcuts

All Oz\* apps use the `command_keys` parameter to assign command keys to menu items. The parameter accepts a list of strings where each string has up to three substrings separated by commas:

```
menu name,item name or number[,key]
```

The first substring should be a known menu name, like File or Edit. The Show submenu is also allowed.

The second substring is either an item name, or an item number ranging from 1 to the number of items in the menu. A warning message is displayed if the item name is unknown, or if the item number is illegal or out of range. The item name need not specify all characters in the menu item; you can supply just enough starting characters to get a match, like this:

```
command_keys = $c Edit,Ed,e
```

Any attempt to set a command key for a separator item or submenu is silently ignored.

The third substring specifies the command key; it's either a single character or a hexadecimal number ranging from `$0` to `$FF`. When specifying a letter it is best to use lowercase (see why below); OzTeX will convert it to uppercase when it adds it to the menu item.

To turn off a command key just skip the third substring, like this:

```
command_keys = $c Show,1
```

OzTeX uses View menu entries to set various single-key shortcuts for use when previewing. You don't need to press the command key to use these shortcuts, and if you specify lowercase letters then you don't have to press the shift key. If the specified character is less than a space (`$0` to `$1F`) then no command key is set but a keyboard shortcut is created. For example, `"View,Show,$D"` makes the return key a shortcut for the "Show View/OzTeX" item.

Also, the `"TeX,2,..."` key can be used when previewing (to go to a TeX page number) and when typesetting (to run TeX again). The `"View,2,..."` key can also be used when typesetting (to open the DVI file).

OzTeX uses a few additional parameters (`first_page`, `last_page`, `scroll_forwards` and `scroll_backwards`) to assign single-key shortcuts that are only used when previewing and are not associated with any menu items.

## 4.8 Adding extra tools

The `extra_tools` parameter is used to add extra items to the Tools menu. Each entry in the given list can either be “--” (for a separator item) or a sequence of up to four substrings separated by commas:

```
sig,menu item text[,tex|dvi|ps|*[,xxx]]
```

The first substring must be the 4-character signature of the application you want OzTeX to call. To find the signature of an application simply drop it onto the XRay shareware utility.

The second substring should be the application name, but it is only used for the menu item so it can contain any text you like (except a comma).

The third substring can be `tex`, `dvi`, `ps` or `*`. The first three values tell OzTeX to check for the existence of the current TeX, DVI or PostScript file respectively; the `*` value tells OzTeX to look for the current TeX file first, then the current DVI file, then the current PostScript file. If the file exists then OzTeX will ask the specified application to open another file that results from combining the existing file’s base name with the extension given by the fourth substring (see below).

If there is no current TeX/DVI/PS file then the menu item ends in “...” to indicate that selecting it will bring up the open-file dialog so you can choose a file to be processed. The third substring can also be empty or simply not given. For example, the entry “OzMF,MF” is equivalent to “OzMF,MF,,” and tells OzTeX to call OzMF without opening any file.

The fourth substring is a file name extension that replaces the extension given by the third substring; the resulting file is opened when OzTeX calls the application. For example:

```
CAR0,Acrobat,tex,pdf
```

This entry tells OzTeX to open `foo.pdf` in Acrobat, assuming `foo.tex` is the current TeX file. If the fourth substring is empty or not given then the third substring is used. For example, “gsVR,MacGSView,ps” tells OzTeX to open `foo.ps` in MacGSView. The fourth substring is ignored if the third substring is empty.

The `extra_tools` parameter can also be used to tell OzTeX to call Unix commands. The commands are added to the Tools menu by using a special type of entry that has up to five substrings:

```
unix,command text,menu item text[,tex|dvi|ps|*[,xxx]]
```

The first substring must be `unix`, the second is the desired command. The last three substrings have the same semantics as above. More information about this powerful feature is provided in the next section.

Note that if the menu item text in an `extra_tools` entry matches an existing item then the earlier entry is replaced (instead of a new item being appended). This makes it easy to replace one or more items supplied in the `Default` config file.

## 5 Calling Unix commands

OzTeX can call arbitrary Unix commands and display the output in the OzTeX window. This means you can have quick and easy access to a huge amount of free, high-quality Unix software. In particular, you can run many useful teTeX and Ghostscript programs from within OzTeX (teTeX is the standard TeX implementation on Unix systems).

If you don’t have teTeX or Ghostscript on your Mac then download Gerben Wierda’s installer; see <http://www.rna.nl/tex.html> for more details. It’s a large download (over 60Mb) so you might prefer to use the installer included on the OzTeX CD.

A Unix command can be added to the Tools menu by using a “`unix,...`” entry in the `extra_tools` list. The following entries appear in the `extra_tools` list in the Default config file; they show how to call various  $\text{TeX}$  and Ghostscript programs:

```
"unix,pdflatex -efmt=pdflatex -shell -int=nonstopmode %f,pdflatex,tex"
"unix,pdfetex -efmt=pdfetex -shell -int=nonstopmode %f,pdftex,tex"
"unix,latex -int=nonstopmode %f,latex,tex"
"unix,tex -int=nonstopmode %f,tex,tex"
"unix,bibtex %b,bibtex,tex,aux"
"unix,oz-mac2unix %f; makeindex %f,makeindex,tex,idx"
"unix,ps2pdf %f; open %b.pdf,ps2pdf,ps"
"unix,dvips -Ppdf -o %b.ps %f; ps2pdf %b.ps; open %b.pdf,dvi2pdf,dvi"
"unix,oz-tex2pdf %f,tex2pdf,tex"
```

If an entry specifies a fourth substring (`tex/dvi/ps/tt *`) then  $\text{OzTeX}$  automatically changes the current working directory to the location of the input file before calling the Unix command.

$\text{OzTeX}$  also substitutes `%f` with the input file name and `%b` with the input file’s base name; e.g., `foo` if `foo.tex` is the input file. Note that `bibtex` requires a base name. Most programs should be given a full name, especially if you want to run `latex` or `pdflatex` on a file name ending in “.ltx”. Other rarely needed substitutions are `%%` for a percent sign and `%c` for a comma. If the view window is open then `%p` is substituted with the current DVI page number so you can specify a command like this:

```
"unix,dvi2tty -w132 -P%p %f,dvi2tty,dvi"
```

Selecting a user-defined Tools menu item will not close the view window.

The “`-int=nonstopmode`” option used in some of the above commands prevents user interaction. This is a good idea because  $\text{OzTeX}$  can’t really handle a command that requires user input; if it happens then the process stops at that point.

The “`-efmt`” option used in the `pdflatex` item will override any “`%&latex`” line in the input file; this lets you typeset a file with either `pdflatex` or  $\text{OzTeX}$ ’s built-in  $\text{L}^{\text{A}}\text{TeX}$ . Similarly, the “`-efmt`” option in the `pdftex` item overrides any “`%&plain`” line in the input file so you can typeset a file with either `pdftex` or  $\text{OzTeX}$ ’s plain  $\text{TeX}$ . Note that the `pdftex` item actually calls `pdfetex`.

The “`-shell`” option allows `pdflatex` and `pdftex` to run shell scripts. The main reason for doing this is so `pdflatex` can call `epstopdf` to automatically convert EPS files to PDF files; see the “`\ifx\pdfoutput`” code near the top of `TeX/Docs/LaTeX/ozuser.tex`.

Some  $\text{TeX}$  programs, like `makeindex`, can only process input files with Unix line endings, so to run these programs on Mac files you can use `oz-mac2unix` to translate the line endings, as in the above example. The `oz-mac2unix` program is stored in `TeX/Unix/` and might be useful in other situations — it’s safer than using `tr` or a Perl script because it does not change a given file’s type or creator info, nor does it delete the file’s resource fork. Note also that unlike the old `MakeIndex` app, the `makeindex` command does not look for a `.ist` file in the same folder as the `.idx` file; instead it looks for a `.mst` file which must use Unix line endings.

Prefixing a Unix command with “`quiet` ” tells  $\text{OzTeX}$  not to display any output in the  $\text{OzTeX}$  window, nor will the view window be closed if it’s open. This is nicer for some commands like this example which opens the Terminal application:

```
"unix,quiet open -a Terminal,Terminal"
```

If you want to do lots of commands in one step then just write a shell script and call that. The `tex2pdf` item in the earlier example runs `oz-tex2pdf`, a script located in `TeX/Unix/`, which contains these commands:

```
#!/bin/sh -e
# call latex if first line of input file starts with "%&latex"
if [ $(tr "\r" "\n" < $1 | head -n1 | grep -q -c "%&latex") == 1 ] ; then
    latex -int=nonstopmode $1
else
    tex -int=nonstopmode $1
fi
basename=${1%.*} # strip off extension
echo "" ; echo "----- calling dvips -----"
dvips -Ppdf -o $basename.ps $basename.dvi
echo "" ; echo "----- calling ps2pdf -----"
ps2pdf $basename.ps
echo "Done! The pdf file will now be opened..."
open $basename.pdf
```

The “-e” option on the first line causes the process to stop if a command terminates with an error status.

The `ps2pdf`, `dvi2pdf` and `tex2pdf` items all use the `open` command to open the PDF file in your preferred viewer. These items rely on Ghostscript and will use its default paper size. Note that Gerben Wierda’s Ghostscript installer allows the default paper size to be set to A4 or US Letter at installation time.

Don’t forget you can assign command keys to any of these menu items. For example, to assign `command-9` to the `tex2pdf` item, put this in your `Local` file:

```
command_keys = $c Tools,tex2pdf,9
```

## 5.1 Changing the shell

OzTeX runs a given Unix command by invoking a shell and passing along your command. The default shell used by OzTeX is `sh`. If for some reason you’d prefer to use `tcsh` then put “`unix_shell = tcsh`” in your `Local` file, but note that before executing your commands `tcsh` will automatically execute the commands in `~/ .tcshrc` or `~/ .cshrc` if either file exists. Or if you decide to set “`unix_shell = zsh`” then `~/ .zshenv` is executed. If a simple name like `tcsh` or `zsh` is specified then it must exist in `/bin`, so if the shell you want to use is somewhere else then supply the full path; e.g., “`unix_shell = /usr/local/bin/bash`”.

## 5.2 Changing PATH

Unix shells use the `PATH` environment variable to locate commands that don’t contain an explicit location. Before calling a Unix command, OzTeX sets `PATH` to the directories specified by the `unix_path` parameter. The `Default` setting assumes you used Gerben Wierda’s installer. If you have a different version of `teTeX` (or if you installed Gerben’s version using `fink`) then you’ll probably need to set `unix_path` to the value returned by typing “`echo $PATH`” in Terminal.

A couple of the above examples call scripts stored in the `TeX/Unix/` folder. The Unix path corresponding to this folder is automatically appended to `PATH` so that the scripts can be called without having to specify an explicit location. This also lets you override those scripts by simply putting your own versions in a directory specified anywhere else in the `unix_path` value.

NOTE: OzTeX does *not* use the `unix_path` and `unix_shell` parameters when it calls the `send_ps` command to do low-level PostScript printing.

If you have any Unix or `teTeX`-related problems then make sure you subscribe to Gary Gray’s “TeX on OS X” mailing list by sending email to [MacOSX-TeX-on@email.esm.psu.edu](mailto:MacOSX-TeX-on@email.esm.psu.edu).

## 6 OzTeX on OS X

This section discusses how to make the most of the new features in OS X, the problems that still exist, and a quick look at Unix from an OzTeX user's perspective.

### 6.1 Advantages

One of the nicest features in OS X (from a programming point of view) is how easy it is to call Unix commands from an application. OzTeX tries to take full advantage of this by letting you call any Unix commands you like. You can add Unix commands to the Tools menu for quick selection (see section 5) or include them in a task list that is activated when a file is dropped onto OzTeX (see section 7). The ability to call Unix commands also makes it possible to implement low-level PostScript printing via the `send_ps` parameter (see section 2.6.2).

Memory management in OS X is much nicer. Applications have access to the full (virtual) memory space and don't have to specify a preferred memory size in which to run. This means OzTeX can build large formats or preview complicated graphic files without you having to use the Finder to increase OzTeX's memory size.

### 6.2 Known problems

OzTeX has problems previewing PostScript fonts under OS X because the PostScript font renderer built into OS X is buggy and does not produce the same results as ATM on OS 8/9. The best solution is to use TrueType fonts instead of PostScript fonts. If you look in the `Extras/PStoTTF/` folder you'll find a FontForge script that can be used to convert PostScript fonts into TrueType format. A large set of TrueType fonts are provided on the OzTeX CD.

Even using TrueType fonts you might see the occasional problem. For example, on Panther the previewer fails to find some hidden fonts like `%CMMI10`. I'm not sure if the bug is in Panther or in the hidden fonts created by Selwyn Hollis, but the solution is to install his non-hidden fonts. They are supplied on the OzTeX CD or you can download them from <http://www.appliedsymbols.com/cmfs/>.

If you use the standard OS X print dialog to print a DVI file to a PostScript printer but try to click the Preview or Save As PDF button then OzTeX will display this error message: "PICT-with-PS document format is not supported!". There's not much I can do about this; OzTeX has to use that format to be able to include PostScript code (via `dvips`), but Apple's PDF driver does not support that format. But there really is no need to use the print dialog to create a PDF file. If you have `teTeX` and Ghostscript installed then just select the `dvi2pdf` item from the Tools menu. Section 15 describes various ways to create PDF files.

The Oz\* apps currently do not support the much longer file names that are allowed on OS X, so you'll have to restrict yourself to 31 characters.

### 6.3 Using teTeX's texmf tree

If you have `teTeX` installed on your system then you might want to tell the Oz\* apps to use the files in `teTeX`'s `texmf` tree. The `Use texmf tree` config file in the `TeX/Configs/` folder specifies search paths that match the folder structure in Gerben Wierda's latest `teTeX` distribution. However, if you have a `~/Library/texmf` tree which you want to be searched first then you'll need to do things like this in your `Local` file:

```
current_app = Oz*
load_config = "TeX/Configs/Use texmf tree"
```

```

current_app = OzTeX OzTtH
input_folders = ~/Library/texmf/tex/** $c

current_app = OzTeX OzMF OzMP
tfm_folders = ~/Library/texmf/fonts/tfm/** $c

```

WARNING: It is not a good idea to tell  $\text{TeX}$  to use *OzTeX*'s files, so avoid modifying  $\text{TeX}$ 's search paths to look in *OzTeX*'s supplied folders, and avoid copying files from those folders into a directory where  $\text{TeX}$  tools will find them (such as `~/Library/texmf`). If you must do something like that then you need to be aware that some  $\text{TeX}$  tools like `texconfig` can't handle input files with Mac line endings, so you'll get an error if it tries to input an *OzTeX* file like `TeX/DVIPS/Inputs/config.ps`.

## 6.4 Tips for Unix beginners

Sooner or later you will need to confront Unix face to (inter)face. The Terminal application (in `/Applications/Utilities`) is your window into Unix. It provides a command-line interface into which Unix commands can be typed. Here are some of the more useful commands:

```

man      – displays help on a given topic or command (e.g., man man)
pwd      – displays the full path of the current directory
cd       – changes the current directory
ls       – lists information about files
chmod    – changes file permissions and other attributes
rm       – removes files (permanently, so be careful!)

```

Here are some non-obvious tips that might come in handy:

- If a `man` command doesn't find anything useful then try typing "`apropos foo`". It will display a list of `man` topics that mention the word `foo`.
- If a `man` display shows references like `foo(2)` you can display the entry for that reference by typing "`man 2 foo`".
- The `open` command can be used to open files in a given application. You can also use it to open directories in the Finder and explore the Unix directory structure. For example, "`open /usr/local/TeX`" will open the  $\text{TeX}$  directory.
- You can drag files or folders from the Finder to a Terminal window and it will enter the full path name for you. So if you need to change the current directory to a folder visible in the Finder then type "`cd` " and drag the folder into the Terminal window. The correct path will be inserted so you can just hit return.
- After typing part of a file or folder name in a Terminal window, hit the tab key to complete the name. If there are multiple matches they will be displayed and the command is then completed up to the point where the names differ. Type another character or two and hit tab again.
- Type "`locate foo`" and it will instantly list all file names containing `foo`. You might first need to type "`sudo /usr/libexec/locate.updatedb`" to update the database used by the `locate` command.
- Use the up/down arrows to scroll backwards/forwards through previous commands.

## 7 Task lists: a simple scripting facility

OzTeX can be told to carry out a list of tasks if a TeX/DVI/PS file is sent to OzTeX via an “open file” event. The Finder sends such an event whenever you drop a file onto the OzTeX icon. The tasks for each type of file are specified by three parameters whose `Default` settings are compatible with OzTeX’s old behavior:

```
drop_tex = tex
drop_dvi = view
drop_ps  = send
```

Each `drop_*` list can include any of the following tasks:

```
tex           - typeset the current TeX file
view          - view the current DVI file
print         - print the current DVI file
send          - send the current PS file to your printer
unix,<cmd>    - call the given Unix command
wait,<file>   - wait for the given file to be created
wait,<number> - wait for the given number of seconds
front[,<sig>] - bring OzTeX or the given app to the front
quit[,<sig>]  - quit OzTeX or the given app
[run,]<tool>  - run the given item in the Tools menu
[load,]<file> - load the given config file
```

Note that the “run,” and “load,” prefixes are optional. You only need to supply them to avoid ambiguity — e.g., you might want to load a config file that has the same name as a Tools menu item. If a `drop_*` string does not match a known task then OzTeX looks for a matching item in the Tools menu and, if found, runs that tool. If the string does not match any item in the Tools menu then OzTeX searches the current `config_folders` list for a matching config file and, if found, loads that file. Case is ignored when matching.

Enough theory, let’s look at some examples. If you want to view the DVI file after typesetting a dropped TeX file then add this line to your `Local` file’s OzTeX-only section:

```
drop_tex = tex view
```

In the following example OzTeX will typeset a given TeX file twice, load a config file, run `dvips`, run `psnup`, then call `ps2pdf`:

```
drop_tex = tex tex "add outline fonts" dvips psnup ps2pdf
```

A problem with the above example is that the `dvips` and `psnup` tasks will pause for user input. You can avoid user interaction in any built-in tool (`dvips` to `pstops`) by specifying the desired options, as in this example:

```
drop_tex = tex tex dvips,-PCMPS "psnup,-2 -d2" ps2pdf
```

Note that supplying options to `dvips` overrides any options in the `dvips_options` parameter. If you want to use `dvips_options` but still avoid the `dvips` dialog then use “`dvips,$c`”.

You can also call Unix commands without having to add items to the Tools menu:

```
drop_tex = tex tex dvips,-PCMPS "psnup,-2 -d2" |
          "unix,ps2pdf %f; open %b.pdf,x,ps"
```

The `unix` task syntax is exactly the same as the syntax for `unix` entries in the `extra_tools` list. The only difference is that the third substring is ignored because no menu item is created.

A task list is aborted if an error occurs, or if you hit escape or command-dot.

### 7.1 Waiting for an application to finish

After calling an external, non-Unix application you might need to tell OzTeX to wait until that app has finished doing its work. This is necessary in the following example which calls Distiller:

```
drop_dvi = dvips,-PCMPS Distill wait,%b.pdf Acrobat
```

The Distill and Acrobat tasks assume that suitably named items have been added to the Tools menu via `extra_tools`. If `foo.dvi` is processed by the above task list then “wait,%b.pdf” tells OzTeX to wait until `foo.pdf` has been created. Unfortunately this might not always work. Detecting that a new file has been created is not a simple task, and you might find that Acrobat is called before Distiller has finished creating `foo.pdf`. If so then try this:

```
drop_dvi = dvips,-PCMPS Distill wait,%b.pdf wait,5 Acrobat
```

OzTeX will wait 5 seconds before calling Acrobat; that should give Distiller enough time to really finish. Or if you want Distiller to quit after it has done its work then do this:

```
drop_dvi = dvips,-PCMPS Distill wait,%b.pdf quit,DSTL Acrobat
```

The “quit,DSTL” task tells OzTeX to send a quit event to the application with the signature DSTL (i.e., Distiller). OzTeX then waits for the app to quit before doing the next task, so there’s no need for the “wait,5” task.

### 7.2 File-specific tasks

One obvious way to use the `drop_*` parameters is to create config files for the various tasks you’d like OzTeX to perform and put them in the Config menu where they can be selected. But this isn’t very convenient if you want to have different tasks carried out on different files.

To allow different tasks to be carried out on a per-folder or per-file basis, OzTeX looks for two special config files in the same folder as the dropped TeX/DVI/PS file. If `drop.config` exists then OzTeX loads it. If `<basename>.config` exists then it loads that file; e.g., `foo.config` if `foo.tex` was dropped. These config files can be used to change the `drop_*` tasks, but the change is only temporary. OzTeX automatically saves the current `drop_*` tasks before either config file is loaded and restores the tasks after the dropped file is processed. This avoids any possibility of an unexpected task list being activated when a different file is dropped.

If `teTeX` is installed then you could drop `.eps` files onto OzTeX and convert them to `.pdf` files by creating (in the same folder) a `drop.config` file with this line:

```
drop_ps = "unix,epstopdf %f; open -a Preview %b.pdf,x,ps"
```

The special config files can be used to change any parameters, not just `drop_*` tasks, but you need to be careful because such changes are *not* temporary. If you have such changes then it’s a good idea to add “Default” as the last task in any `drop_*` lists. This loads the `Default` config file and resets all parameters to their start-up state.

### 7.3 Using menu items to “drop” files

If you hold down the option key before selecting a TeX/DVI/PS file from one of OzTeX’s TeX/Print/View/Send menu items then the file will be processed as if it was dropped onto OzTeX. This saves you having to switch to the Finder to locate and drop a file. Note that the menu item name will change to “Drop...” to indicate what will happen. If the menu item has a command key equivalent then option-command-*key* will also work. For example, if you are using the default `command_keys` then option-command-T will drop the most recent TeX file.

## 8 *L<sup>A</sup>T<sub>E</sub>X*

Oz $\TeX$  is distributed with the latest release of  $\LaTeX$  (formerly called  $\LaTeX$  2e). This version of  $\LaTeX$  is described in the 2nd edition of *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System* by Leslie Lamport [11] and *The L<sup>A</sup>T<sub>E</sub>X Companion* by Michel Goossens, Frank Mittelbach and Alexander Samarin [4].

All the important  $\LaTeX$  packages are provided. The `psnfss` package makes it easy to switch to PostScript text fonts instead of Computer Modern. The `graphics/graphics` packages make it easy to include Macintosh graphic files. See the documentation and test files in the `psnfss` and `graphics` subfolders in `TeX/Inputs/LaTeX/`. Other common packages are kept in the `tools` subfolder. More information about  $\LaTeX$  can be found in the `.tex` files in the `base` subfolder.

The  $\LaTeX$  system included in the download version of Oz $\TeX$  is *not* a full distribution (that would be far too big for most users). After installing the `base` files and most of the `required` packages (all except `babel`), I deleted all the `.ins`, `.dtx` and `.fdd` files, as well as a few other files that were irrelevant for Oz $\TeX$  users. If you need a file or package not provided in Oz $\TeX$  then you can ftp it from any CTAN site or mirror. Note that the subfolders in `TeX/Inputs/LaTeX/` more-or-less correspond to the CTAN directories under `/tex-archive/macros/latex/`. If you only need one or two files then ask me first and I can send them by email, but you must be a registered Oz $\TeX$  user. Another alternative is to buy the Oz $\TeX$  CD which does contain the complete  $\LaTeX$  distribution; see section 18.2.

### 8.1 Installing a new package

A  $\LaTeX$  package usually consists of a `.ins` file, plus one or more `.dtx` files, and possibly some `.fdd` files. To install a package so that Oz $\TeX$  can use it, run  $\LaTeX$  on the `.ins` file. This should create one or more files with extensions of the form `.sty`, `.cfg` or `.fd`. Move all these files into a folder where  $\TeX$  can find them. A good approach is to put them in a new `mytex/xxx/` subfolder (where `xxx` is the name of the package) and then add a line like this to your `Local` config file:

```
input_folders = mytex/xxx/ $c
```

If the package also includes TFM and VF files then put them all in `mytex/xxx/` and use the above technique to extend `tfm_folders` and `vf_folders`. Some packages might include files required by `dvips` (e.g., `config.*`, `*.map`, `*.enc`); if so put them in the same folder and extend `ps_folders`.

If you want to read detailed documentation about the package then use  $\LaTeX$  to process the `.dtx` file and create a `.dvi` file. After using the package and verifying that everything works, you can safely delete the `.ins/.dtx/.fdd` files.

### 8.2 Using $\LaTeX$ 2.09

The old  $\LaTeX$  2.09 system is no longer distributed with Oz $\TeX$ . Everybody should now be using the new  $\LaTeX$ ; note that it can typeset  $\LaTeX$  2.09 documents in “compatibility” mode. If you have the old  $\LaTeX$  system from an earlier version of Oz $\TeX$  and you want to use it alongside the new  $\LaTeX$ , then the best approach is to copy the `LaTeX209.config` and `LaTeX.config` files in the `TeX/Configs/` folder and make any necessary changes to the `input_folders` list.

Just before running  $\TeX$ , Oz $\TeX$  will load `<format>.config` if it exists in the same folder as the input file, so move the `.config` files into the same folder(s) as your  $\LaTeX$  documents. Make sure all your old  $\LaTeX$  documents start with “`%&latex209`” and the corresponding format file is called `LaTeX209.fmt`. Similarly, start all your new  $\LaTeX$  documents with “`%&latex`”.

## 9 Building format files

A format file contains a pre-compiled set of macros that  $\TeX$  can load in very quickly. To build a format file you need to run `INITEX` (in the  $\TeX$  menu). The best approach is to put all new format files in a folder called something like `mytex/formats/` and then set

```
format_folders = mytex/formats/ $c
```

in the `OzTeX`-only section of your `Local` config file. Putting your folder first ensures that `OzTeX` will find your format files before the supplied formats in the `TeX/Formats/` folder.

Here are the steps needed to create `Plain.fmt`:

1. Run `INITEX`, wait for the “\*\*” prompt, type “`Plain\dump`” and hit the return key. `INITEX` will begin reading `TeX/Inputs/Plain/plain.tex`.
2. You’ll eventually see the save-file dialog asking you to save `Plain.fmt`. As explained above, the best location for all new format files is in a new folder like `mytex/formats/`.
3. Note that `Plain.log` is created in the current folder. It’s not needed, so delete it.

The steps needed to create `LaTeX.fmt` are very similar:

1. Run `INITEX`, wait for the “\*\*” prompt, type “`LaTeX.ltx`” and hit return (you don’t need to add `\dump`). `INITEX` will read `TeX/Inputs/LaTeX/base/latex.ltx`.
2. The save-file dialog will show `LaTeX.fmt`, so just save this file in your `mytex/formats/` folder. Delete `LaTeX.log` if you don’t want to keep it.

Creating `AMSTeX.fmt` is just as easy: run `INITEX`, type “`AMSTeX.ini`” and save `AMSTeX.fmt` in a suitable folder.

If you create a completely new format called `foo.fmt` and you want its name to appear in the  $\TeX$  menu then just add the line “`format_files = $c foo`” to your `Local` config file.

### 9.1 Building big formats

Some people with very large or complicated documents will need to build bigger formats than the one’s supplied with `OzTeX`. Here’s what to do:

1. Edit the `Local` config file and put your new  $\TeX$  parameter settings in the `OzTeX`-only section.
2. Restart `OzTeX` and follow the steps in the above section to rebuild all your formats.

### 9.2 Building custom formats

`INITEX` can also be used to create custom formats to speed up typesetting, especially for  $\LaTeX$  documents that use lots of packages. Let’s say you have a file called `thesis.tex`. Edit this file, cut out everything before `\begin{document}` and paste it into a new file called, say, `thesis.ini`. Set the current folder by selecting “ $\TeX$ ...”, locating `thesis.ini` and hitting `Cancel`. Select `INITEX`, type “`&LaTeX thesis.ini\dump`” and save `thesis.fmt` in the current folder. Add a new top line in `thesis.tex` containing `%&thesis`. Now when you typeset `thesis.tex` `OzTeX` will load `thesis.fmt` and you should see a significant reduction in the typesetting time.

## 10 Fonts

OzTeX can use the following sources of font information in the process of typesetting, previewing and printing a document: TFM files, PK files, VF files, PostScript font files (for downloading non-resident fonts) and Macintosh screen fonts. The following subsections discuss all these font sources in some detail.

### 10.1 TFM files

A TeX Font Metric file contains the crucial typesetting information for a font, such as each character's height, depth and width. The actual character images are not stored in a TFM file. TFM files are the only source of font information used when running TeX; it doesn't need to know anything about character images to be able to create a DVI file. See *The TeXbook* [8] if you want to know more about how TeX uses fonts, especially Chapter 4 and Appendix F.

OzTeX reads TFM files when printing or viewing a DVI file containing any of the non-PK fonts listed in your config file(s). The TFM file specified by the `dummy_tfm_file` parameter will also be read if a requested PK file can't be found (PK files are discussed in section 10.3).

OzTeX will look for a TFM file in the current folder, then in the list of folders specified by `tfm_folders`. The `Default` file tells OzTeX to search all subfolders in the `TeX/TFM/` folder. The supplied subfolders contain all the standard TFM files needed by most TeX/LaTeX documents.

### 10.2 Creating new TFM files

If the OzTeX distribution doesn't provide a TFM file for a font you would like to use then there are a number of things you can do:

1. If you have access to the Internet then look for the TFM file in the nearest CTAN site (see the "Related Software" item in the Help menu). If the font is called `foo` then look for `foo.tfm`. If you find it, remember that a TFM file must be transferred as a binary file.
2. If you can't find `foo.tfm` then look for `foo.mf`. Such a file can be used by METAFONT to create `foo.tfm` (and a corresponding set of PK files). See section 11.2.
3. If `foo` is a PostScript font then look for `foo.afm`. An AFM (Adobe Font Metric) file contains very similar information to a TFM file. Use `AFM2TFM` in OzMF to convert `foo.afm` to `foo.tfm`; see the `dvips` manual for detailed information about `AFM2TFM`. Alan Jeffrey's `fontinst` package can also convert an AFM file into a PL or VPL file which OzMF can then convert into a TFM file using `PLtoTF` or `VPltoVF`.
4. As a last resort you could even create a PL file from scratch. A very tedious job.

Having found or created a new TFM file, you need to place it in a folder where OzTeX can find it. A recommended approach is to put all new TFM files in a new folder called `mytex/tfm/` and then add these lines to your `Local` file:

```
current_app = OzTeX OzMF
tfm_folders = mytex/tfm/ $c
```

Note the use of `current_app` to ensure that OzTeX and OzMF search the same set of folders.

If the TFM file is for a PostScript font then you must also add a suitable entry between `add_fonts` and `end_fonts`. And if you use `dvips` then it needs a similar type of entry added to a `psfonts.*` or `*.mf` file. Section 10.4 has all the information about these font entries.

### 10.3 PK files

PK (packed pixel) files store the character images needed to print or view a DVI file containing bitmap fonts; these are all the fonts *not* listed in your configuration files (or `psfonts.*` and `*.map` files in the case of `dvips`). Each bitmap font can have any number of PK files, each one representing the same font but at a different size; e.g., `cmr10.300pk` and `cmr10.144pk`. When `OzTeX` (or `dvips`) is looking for a PK file it calculates the requested size using:

$$\text{size} = \text{resolution} \times \text{magnification}$$

where *resolution* is either the printing resolution or the viewing resolution, and *magnification* is the overall font magnification (i.e., the DVI magnification times the individual font scaling). The *size* is then rounded up to the nearest integer. For example, assuming a *resolution* of 300 and no document magnification:

| $\text{\TeX}$ input                            | <i>size</i>            | PK file                  |
|--|------------------------|--------------------------|
| <code>\font\ra=cmr10</code>                    | $300 \times 1.2^0$     | <code>cmr10.300pk</code> |
| <code>\font\rb=cmr10 scaled\magstephalf</code> | $300 \times 1.2^{0.5}$ | <code>cmr10.329pk</code> |
| <code>\font\rc=cmr10 scaled\magstep1</code>    | $300 \times 1.2^1$     | <code>cmr10.360pk</code> |
| <code>\font\rd=cmr10 scaled\magstep2</code>    | $300 \times 1.2^2$     | <code>cmr10.432pk</code> |
| <code>\font\re=cmr10 scaled\magstep3</code>    | $300 \times 1.2^3$     | <code>cmr10.518pk</code> |
| <code>\font\rf=cmr10 scaled\magstep4</code>    | $300 \times 1.2^4$     | <code>cmr10.622pk</code> |
| <code>\font\rg=cmr10 scaled\magstep5</code>    | $300 \times 1.2^5$     | <code>cmr10.746pk</code> |

$\text{\TeX}$ 's `\magnification` command has a cumulative effect on font scaling:

```
\magnification=\magstep1           % document magnification = 1.2
\font\bigrm=cmr10 scaled\magstep2  % font magnification = 1.44
```

The font size is now  $300 \times 1.2 \times 1.44 = 518.4$ , so `OzTeX` will look for `cmr10.518pk`. Note that  $\text{\LaTeX}$  has no `\magnification` command; it uses options like `11pt` or `12pt` to increase the size of all fonts in a document.

`OzTeX` uses a very flexible search scheme when looking for fonts. If `foo` is a PostScript font then `OzTeX` looks for `foo.tfm` in the current folder, then in the `tfm_folders` list. If `foo` is not a PostScript font then `OzTeX` looks for a PK file using the list of file templates specified by the `pk_files` parameter. `OzTeX` uses the first PK file that matches a template after doing the following substitutions:

- `%f` is replaced by the desired font name (e.g., `cmr10`)
- `%d` is replaced by the desired font size (e.g., `360`)
- `%b` is replaced by the base resolution (e.g., `300`)
- `%m` is replaced by the mode (e.g., `laserwriter`)
- `%%` is replaced by `%` (in case you want `%` in a folder name)

For example, using the `Default` settings and a calculated size of 300, `OzTeX` will look for:

1. `foo.300pk` (in the current folder)
2. `TeX/PK/300/foo.300pk` (relative to `OzTeX`)
3. `New PKs/laserwriter/foo.300pk` (ditto)

In fact, at each step `OzTeX` will also add and subtract 1 from 300 because rounding errors can occur in the size calculation. If still not found, `OzTeX` either gives up or calls `OzMF` to make the missing PK file (see section 11.1). Note that the final template ("`New PKs/%m/%f.%dpk`")

matches the `pk_file` template in OzMF's `Default` config file. This allows OzTeX (and dvips) to find new PK files created by OzMF.

Most of the PK files supplied with OzTeX have been specifically generated for a 300 dpi, write-black laser printer (like the Apple LaserWriter). If your printer doesn't have the same characteristics then you can build new PK files using OzMF as described in section 11.

Note that the 300 dpi PK files supplied with OzTeX can also be used on a 360 dpi printer like the StyleWriter (because  $360 = 300 \times 1.2$ ). Two subfolders, 394 and 896, complete the standard set of magnifications for a 360 dpi printer: 394 is the nearest integer to  $360 \times 1.2^{0.5}$  and 896 is the nearest integer to  $360 \times 1.2^5$ .

## 10.4 PostScript fonts

To be able to view or print a PostScript font in a TeX document, OzTeX needs to know certain information about the font, especially the name of its corresponding TFM file. This information is specified in configuration files on lines between `begin_fonts` and `end_fonts`. The supplied `Default` file has entries for the standard Adobe fonts. Config files can also *append* font information by using `add_fonts` instead of `begin_fonts`; see the `Add Outline Fonts` file in `TeX/Configs/`.

For dvips, information about PostScript fonts is usually stored in `psfonts.*` or `*.map` files. By default, dvips reads `TeX/DVIPS/Inputs/psfonts.map`; this file has font entries that correspond to those in OzTeX's `Default` file. When adding new PostScript fonts it is important to keep the OzTeX and dvips information synchronized.

The PostScript TFMs stored in the `PS` subfolder in `TeX/TFM/` are meant to be used in conjunction with virtual font files stored in the `PS` subfolder in `TeX/VF/` (section 10.7 has more information about virtual fonts). These TFMs and VFs were created for the `psnfss` package by Sebastian Rahtz. The fonts are supplied in both OT1 and T1 encodings. The default encoding is OT1; this is the old TeX text font encoding described in *The TeXbook*. The OT1 TFM/VF files have names ending in `7t`. The T1 (or Cork) encoding is the new TeX text font encoding for 256-character fonts. The T1 TFM/VF files have names ending in `8t`. See `test0.tex` in `TeX/Inputs/LaTeX/psnfss/` for how to select the different encodings.

Note that virtual font names should not appear in a config file's list of fonts, nor in a `psfonts.*/*.map` file. Only the corresponding "raw" fonts should be listed; these are indicated by TFM names ending in `8r`. For example, when OzTeX is processing virtual fonts in a DVI file created with the L<sup>A</sup>TeX `times` package, it will use the commands in `ptmr7t.vf` to replace virtual characters from `ptmr7t.tfm` with actual characters from `ptmr8r.tfm`.

### 10.4.1 Reencoding PostScript fonts

Most of the raw PostScript TFM files distributed with OzTeX use the `8r` encoding. The corresponding screen and printer fonts do not use this encoding, so OzTeX must be told to reencode the screen fonts and dvips must be told to reencode the printer fonts.

To reencode screen fonts during previewing, OzTeX uses information stored in encoding files. The names of these files can appear in a config file's list of PostScript fonts. For example, the `Default` config file includes an entry like this:

```
ptmr8r Times-Roman Times Mac8r.enc
```

This tells OzTeX to map character codes from `ptmr8r.tfm` to characters in the Times screen font using a mapping defined in `Mac8r.enc`. OzTeX looks for encoding files in the `enc_folders` list (it does *not* look in the current folder first). For more information about encoding files, see section 10.6. Note that OzTeX ignores the `Times-Roman` field; it only serves as a reminder of which printer font is represented by the rather cryptic TFM name.

To reencode printer fonts, `dvips` gets the necessary information from a `psfonts.*` or `*.map` file. The `psfonts.map` file supplied in `TeX/DVIPS/Inputs/` contains lines like:

```
ptmr8r Times-Roman "TeXBase1Encoding ReEncodeFont" <8r.enc
```

This tells `dvips` to reencode `ptmr8r.tfm` using an encoding vector called `TeXBase1Encoding`. The vector is downloaded in a file called `8r.enc`; `dvips` looks for `8r.enc` in the current folder first, then in the `ps_folders` list. The encoding file is usually recognized by the `.enc` extension but it isn't essential; if `dvips` sees an entry like "`<[foo.blah`" then it assumes `foo.blah` is an encoding file.

NOTE: Screen font reencoding is not as reliable as printer font reencoding because some characters simply don't exist in Mac screen fonts (font suitcases normally use the standard Mac text encoding called "Standard Roman Encoding"). This means that `OzTeX`'s preview might not match the output from `dvips`. The solution to this problem is to use a font editor/creator like `Fontographer` to create suitable Mac fonts, or to purchase such fonts from a supplier like `Y&Y`. (`Y&Y` also sell a font manipulation package that lets you create new font suitcases to match PostScript fonts, but it is only available for PCs.)

#### 10.4.2 Downloading PostScript fonts

Every PostScript printer has a certain set of resident fonts. To print an alphabetical list of the PostScript fonts available in your current printer, choose "Send PostScript..." from the File menu and send the file `TeX/PS/fontlist.ps`.

The font you wish to use may not reside in the current printer, so `dvips` provides a mechanism for downloading non-resident PostScript fonts. For example, the following line appears in the supplied `psfonts.map` file:

```
putr8r Utopia-Regular "TeXBase1Encoding ReEncodeFont" <8r.enc <putr.pfb
```

When `dvips` reads a DVI file, it checks to see if `putr8r` is used anywhere on a selected page. If so, the above entry tells `dvips` to include `putr.pfb` in the PostScript output (along with `8r.enc`). `dvips` looks for `putr.pfb` in the current folder first and then in the list of folders specified by `ps_folders`.

The standard version of `dvips` downloads font data stored in `.pfa` or `.pfb` files. `OzTeX`'s implementation of `dvips` can also download font data stored in Macintosh PostScript fonts; i.e., files of type `LWFN`. The type of the file specified after "`<`" is checked and if it is `LWFN` then `dvips` will download the PostScript code stored in the file's `POST` resources. On OS X, `LWFN` files are not all that useful from `OzTeX`'s point of view, mainly because the OS X PostScript font renderer does not produce the same results as ATM on OS 8/9. On OS X a `$f` entry is replaced by the following paths, if they exist:

```
~/Library/texmf/fonts/type1/**
/usr/local/teTeX/share/texmf.local/fonts/type1/**
/usr/local/teTeX/share/texmf.gwtex/fonts/type1/**
/usr/local/teTeX/share/texmf.tetex/fonts/type1/**
/usr/local/teTeX/share/texmf/fonts/type1/**
```

These are the search paths used by `teTeX`'s `dvips` to locate `pfa/pfb` files, assuming you are using Gerben Wierda's latest `teTeX` installation.

Partial downloading of PostScript fonts is available via the `-j` option. This is an excellent way to reduce the amount of output, and it uses less printer memory. However, partial downloading might cause problems with some fonts, so you can use an entry like "`<<foo`" to force a font to be fully downloaded even when the `-j` option is used.

Another handy feature supported by `dvips` is the automatic downloading of PostScript fonts used (but not included) in EPS files; see section 14.1 for details.

### 10.4.3 PostScript fonts on a non-PostScript printer

OzTeX can print a DVI file that uses PostScript fonts on a non-PostScript printer by sending `dvips` output to your printer using CUPS, so this should never be a problem.

### 10.4.4 PostScript or Computer Modern?

There are some important differences between a PostScript font like Times-Roman and an “equivalent” Computer Modern Roman font:

1. A PostScript font can be requested at virtually any size. For example, you can do things like “`\font\helv=ptmr8r at 33.3pt`”. You can also ask for `cmr10` at any size, but a separate PK file is needed for each size. However, you can avoid PK files altogether by installing PostScript versions of the Computer Modern fonts; see below.
2. When a PostScript font like Times-Roman is scaled to the same design size as a CMR font it tends to look darker and larger. It is not a good idea to mix the two font designs in the same document.
3. For Plain TeX users, the standard TeX commands for accents and foreign letters need to be redefined for a PostScript text font. The file `pstext.tex` in the `Plain` subfolder in `TeX/Inputs/` contains the required macro definitions. Another file, `psfonts.tex` in `TeX/Docs/Plain/`, inputs these macros and illustrates the use of PostScript fonts in a Plain TeX document.

LaTeX users can typeset a document with PostScript text fonts rather than Computer Modern by simply adding a suitable package declaration. For example,

```
\documentclass{article}
\usepackage{times}
```

will switch the default text font to Times-Roman. See the documentation for the `psnfss` package in `TeX/Inputs/LaTeX/` for other possibilities. Note that CM math fonts will still be needed to typeset mathematics, but you can buy commercial font packages such as Lucida or MathTime that include PostScript replacements for these math fonts. All the necessary support files for Lucida and MathTime are stored in the `Extras/Y&Y/` folder. A cheaper alternative is to use the `mathptmx` package that comes with `psnfss`.

If you like the look of Computer Modern fonts, but you also want the flexibility of PostScript fonts, then you can have the best of both worlds. PostScript versions of the CM and AMS fonts are available at CTAN sites and on the OzTeX CD. On OS X it is best to install TrueType fonts because the PostScript font renderer built into OS X does not produce the same results as ATM. Selwyn Hollis has created free TrueType versions of the CM and AMS fonts. They are included on the OzTeX CD, or you can download them from Selwyn’s site: <http://www.appliedsymbols.com/cmfs/>.

The `Add Outline Fonts` config file works with either PostScript or TrueType versions of the CM and AMS fonts. The corresponding `config.CMPS` and `psfonts.CMPS` files for `dvips` are in the `TeX/DVIPS/Inputs/` folder.

## 10.5 Macintosh screen fonts

The font entries listed between `begin_fonts` (or `add_fonts`) and `end_fonts` in a configuration file are for non-PK fonts — that usually means PostScript or TrueType fonts.

When previewing a DVI file containing a non-PK font, OzTeX draws characters using the corresponding Macintosh screen font given in the config file. If the requested screen font can't be found then OzTeX will warn you and use the font specified by the `dummy_screen_font` parameter (Geneva is the default setting). All characters from a missing font will be drawn in a color specified by the `missing_fonts` parameter (red is the default).

If you look inside the `Add Outline Fonts` config file you'll see screen font names like `“%cmr10”`. The percent at the start of the name indicates that the font should be “hidden”; i.e., its name should not appear in the Font menu of any Mac application. This is done to avoid cluttering up these menus with fonts that aren't really used by most apps. OzTeX can still find and display hidden fonts. In fact OzTeX checks for both hidden and non-hidden versions of a requested screen font name. This means that config files like `Add Outline Fonts` will work regardless of whether you've installed the hidden or non-hidden fonts.

## 10.6 Encoding files

Encoding files give you control over which glyph is displayed when a particular character from a Macintosh screen font is to be viewed. (The `.enc` files downloaded by `dvips` during PostScript printing serve a similar purpose to OzTeX's encoding files, but the syntax is quite different.)

Encoding files are loaded after a config file containing font entries between `begin_fonts` (or `add_fonts`) and `end_fonts` has been loaded. Every screen font entry must be followed by an encoding entry which is either the name of an existing encoding file, or `nil` if the screen font uses the same encoding as its corresponding TFM file. OzTeX looks for a given encoding file in the current `enc_folders` list. It does *not* look in the current folder because that concept isn't well-defined when a configuration file is loaded.

If the encoding file is found then OzTeX uses it to build an encoding array of 256 elements. This array defines the mapping of DVI character codes into screen font glyphs. You can see the currently loaded encoding arrays by selecting the appropriate “Show” item in the Config menu.

The syntax of an encoding file is described in detail in the standard encoding file called `Mac.enc` which is kept in the `TeX/Encodings/` subfolder. This encoding file maps characters from a standard Adobe text font into equivalent characters in a standard Macintosh text font. Depending on the encoding scheme used in your TFM files or screen fonts, you might need to create other encoding files.

## 10.7 Virtual fonts

Virtual fonts provide a convenient and flexible way to specify a mapping from TeX's notion of a font character to the capabilities of a particular output device (e.g., screen or printer). In particular, virtual fonts make it easy to use PostScript fonts with TeX. The definitive document on virtual fonts is Donald Knuth's TUGboat article [7]. A text version called “Knuth on VFs” can be seen in OzMF's Help menu.

OzTeX and `dvips` process virtual fonts automatically, so most people never need to worry about whether or not a DVI file uses such fonts. To decide if font `foo` is virtual, OzTeX and `dvips` look for a file called `foo.vf` in the current folder, then in the list of folders specified by `vf_folders`. If the `.vf` file is found then the commands in the file are processed and all virtual characters are eventually replaced by characters from one or more “real” fonts. (That's the usual case. Virtual characters can actually be made up of rules or even `\special` commands, so a real font might never be referenced.)

## 11 OzMF

OzMF is a companion program for OzTeX which includes various font related tools. The most important of these is METAFONT, a program written by Donald Knuth and used to create TeX-compatible fonts. For more information, see *The METAFONTbook* by Knuth [9]. The other font related tools included in OzMF are:

**MakeTeXPK** reads a given text file, normally with a `.make` extension, and uses commands in the file to build one or more PK files. You don't need to create `.make` files yourself. If OzTeX is asked to make missing PK fonts then it will pass a temporary file called `OzTeX-to-OzMF.make` to OzMF so that MakeTeXPK will create the required PK file.

**GFtoPK** converts a GF (generic font) file into a PK (packed pixel) file. GF files are created by METAFONT, but most DVI-reading programs like OzTeX prefer to use PK files, so MakeTeXPK automatically runs GFtoPK after running METAFONT.

**GFtoDVI** converts a GF file into a DVI file. Used by font designers to create character proofs.

**GFtype** converts a GF file into a human-readable text file. Mainly used to validate GF files.

**PKtype** converts a PK file into a human-readable text file. Mainly used to validate PK files.

**TFtoPL** converts a TFM (TeX font metric) file into a PL (property list) file.

**PLtoTF** converts a PL file into a TFM file.

**VFtoVP** converts a VF (virtual font) file into a VPL (virtual property list) file.

**VPtoVF** converts a VPL file into a VF file and a TFM file.

**AFM2TFM** converts an AFM (Adobe font metric) file into a TFM file.

For more details about each tool, see the corresponding items in OzMF's Help menu. OzMF provides some nice features for running these tools in a Mac environment. The open-file dialog has a "Do all files" button which allows you to process all selected files in the current folder. Those tools that read text files recognize either return or linefeed as the end-of-line character, so Unix text files can be read unmodified. All tools can run in the background.

The OzMF application should be kept in the same folder as OzTeX, along with the MF folder which contains these subfolders:

**Configs** contains some configuration files used by OzMF, especially the `Default` file which is loaded when OzMF starts up. Before using OzMF, examine the supplied parameters in the `Default` file and read the comments at the bottom. If you need to make changes to any parameters then do so in the OzMF-only section of your `Local` config file.

**Inputs** contains a large library of `*.mf` files. Most of these files contain METAFONT code for creating character bitmaps (in GF files) and corresponding font metrics (in TFM files). METAFONT looks for "`%&base`" on the first line of a selected `.mf` file and, if present, uses the given base to override the currently ticked base in the Metafont menu.

**Bases** contains `mf.pool` (read by INIMF) and various `*.base` files. Base files are the METAFONT equivalent of TeX's format files. INIMF is used to build base files. Two base files are distributed: `Plain.base` and `MakeTeXPK.base`. The latter is used by MakeTeXPK and contains common `mode_defs` for Mac printers, plus some settings suitable for Mac screens; see `MF/Inputs/Etc/MakeTeXPK.mf`.

**Help** contains the files used in OzMF's Help menu.

### 11.1 Making PK fonts on demand

All of OzTeX's dialog boxes for printing or viewing a DVI file have a check box called "Make missing PK fonts". If this box is checked and OzTeX detects a missing PK font, then OzMF will be called to build the required PK file.

OzTeX does not call METAFONT directly. Instead it starts up MakeTeXPK which runs METAFONT to build a GF (Generic Font) file, then runs GFtoPK to convert the GF file into a PK file. OzTeX uses the new PK file to continue previewing or printing.

The location of the PK file is determined by the `pk_file` template which should expand to a valid file path name after the following substitutions:

- `%f` is replaced by the desired font name (e.g., `cmr10`)
- `%d` is replaced by the desired font size (e.g., `360`)
- `%b` is replaced by the base resolution (e.g., `300`)
- `%m` is replaced by the mode (e.g., `laserwriter`)
- `%%` is replaced by `%` (in case you want `%` in a folder name)

OzMF will automatically create any new folders that might occur as a result of these substitutions. The template in OzMF's `Default` file is `"New PKs/%m/%f.%dpk"`. This means that PK files will be created inside a folder called `New PKs` alongside the OzMF application. Furthermore, each PK file is stored in a subfolder based on the current mode (e.g., the `viewing_mode` value when previewing). OzMF will create the `New PKs` folder if required, as well as any new mode subfolders.

NOTE: If you decide to change the template then you'll also need to make sure the identical template appears in the `pk_files` list used by OzTeX so it can find PK files created by OzMF.

It is possible that the font creation process will fail. For example, METAFONT might not find the required `.mf` input file. Or you might decide to cancel the process. In such cases OzTeX will report a missing PK font and append a line of information to a file specified by the `missing_fonts_file` parameter. The `Default` value creates a file called `missfont.make` in the OzTeX folder.

If "Make missing PK fonts" is *not* checked then OzTeX won't attempt to create a missing PK font, but it will append a line to `missfont.make`. This file can be dropped onto OzMF to build all the listed PK fonts. This might be more convenient than having OzTeX build PK fonts on the fly because it can take a while to create lots of fonts.

Note that MakeTeXPK will keep processing if errors occur. All error messages are appended to a `MakeTeXPK.log` file in the same folder as the `.make` file. This is handy when building lots of PK fonts because a minor METAFONT error won't stop the entire process.

It is a good idea to delete `missfont.make` after using it to build PK files. OzTeX simply appends missing font info to `missfont.make` without checking to see if that info already exists, and MakeTeXPK simply builds all the listed PK files, even if they already exist.

If you preview a DVI file and select the "Ignore missing fonts" check box then the "Make missing PK fonts" setting is ignored. If a missing PK font is detected then OzTeX won't attempt to call MakeTeXPK, nor will it append any information to `missfont.make`. This feature is useful when previewing certain DVI files (like `nasty.dvi`) where you know there are missing fonts and have no desire to create them.

### 11.2 Making TFM files

If you have a `foo.mf` file and need a corresponding TFM file then first create a `foo.make` file containing the line:

```
MakeTeXPK foo 303 101 3.0 nullmode
```

Dropping `foo.make` onto OzMF will create `foo.tfm`. Note that the special `nullmode` (defined in `MF/Inputs/Etc/modes.mf`) tells `MakeTeXPK` not to create a `GF/PK` file, and to ignore the `delete_tfm` flag so you don't need to select the "Keep TFM files" config file. After creating a new TFM file you should put it in a folder in your `Local` file's `tfm_folders` list so that `OzTeX` can find it. `OzTeX` will call OzMF to build new PK files as and when required.

### 11.3 Setting the Metafont mode

To be able to build a PK file, `METAFONT` needs to know what mode to use. `OzTeX` gets this information from either the `printing_mode` or `viewing_mode` parameter and simply passes the string on to `METAFONT`. For `dvips`, the default mode string is specified in `config.ps` but can be overridden using the `-mode` option.

NOTE: The given strings must exactly match known `mode_def` values defined in `modes.mf` (kept in `MF/Inputs/Etc/`). Also make sure the current printing/viewing resolution matches the `pixels_per_inch` value in the corresponding `mode_def`, otherwise `METAFONT` won't create a suitable PK file.

`OzTeX`'s `Default` config file sets both modes to `laserwriter`. You might need to change these modes, either by selecting one of the supplied config files (e.g., `600 dpi Printer`), or by adding suitable lines to your `Local` config file. If a mode string is empty ("") then `METAFONT` will use the `localfont` mode which is currently equivalent to the `ljfour` mode.

Common modes for `OzTeX` users are `laserwriter`, `stylewriter`, `imagewriter`, `ljfour` and `linohi`. If none of these modes suit your particular printer then look in `modes.mf` to see if there is a suitable definition. The most important thing to look for is a `pixels_per_inch` value that matches the resolution of your printer. If you can't find a matching `mode_def` (unlikely), you will have to create your own and run `INIMF` on `MakeTeXPK.mf` to rebuild `MakeTeXPK.base`. Note that you'll also need to rebuild that base file if you ever download a new version of `modes.mf`.

## 12 OzMP

OzMP is a Mac implementation of John Hobby's `MetaPost` program for producing PostScript pictures using a `METAFONT`-like language. The OzMP application is kept in the same folder as `OzTeX`, along with the `MP` folder which contains these subfolders:

**Configs** contains some configuration files used by OzMP, especially the `Default` file which is loaded when OzMP starts up. Before using `MetaPost`, examine the supplied parameters in the `Default` file and read the comments at the bottom. If you need to make changes to any parameters then do so in the OzMP-only section of your `Local` config file.

**Docs** contains John Hobby's documentation for `MetaPost`. Before printing anything, select "MetaPost" from the Help menu and carry out the given steps.

**Help** contains the files used in OzMP's Help menu.

**Inputs** contains `*.mp` files.

**Mems** contains `mp.pool` (read by `INIMP`) and `Plain.mem`. Mem files are the `MetaPost` equivalent of `METAFONT`'s base files. Use `INIMP` to build mem files.

OzMP includes two supporting tools, `MPtoTeX` and `DVItoMP`. You don't need to run these tools manually because OzMP will call them automatically if your `.mp` file contains `btex ...etex` or `verbatimtex ...etex` sections. Such sections contain `TeX/LATeX` code, so OzMP will also call `OzTeX` in between `MPtoTeX` and `DVItoMP`.

## 13 OzTtH

OzTtH is a Mac implementation of Ian Hutchinson’s TtH program for translating  $\text{\TeX/L\TeX}$  files into HTML files. Please note that TtH is free to use only for non-commercial purposes. For details on the conditions of use select “License” from OzTtH’s Help menu. The OzTtH application should be kept in the same folder as Oz $\text{\TeX}$ , along with the TtH folder which contains these subfolders:

**Configs** contains some configuration files used by OzTtH, especially the **Default** file which is loaded when OzTtH starts up. Before running OzTtH, examine the supplied parameters in the **Default** file and read the comments at the bottom. If you need to make changes to any parameters then do so in the OzTtH-only section of your **Local** config file.

**Docs** contains Ian Hutchinson’s TtH manual (in HTML format).

**Help** contains the files used in OzTtH’s Help menu. Make sure you look at “Read Me” and “OzTtH on OS X”.

**Unix** contains the **tth** program called by OzTtH and some shell scripts that might be called by **tth**.

## 14 DVIPS

Oz $\text{\TeX}$  uses an enhanced version of Tom Rokicki’s **dvips** program to convert DVI files into PostScript. The **TeX/DVIPS/** folder has a couple of subfolders:

**Docs** contains the **dvips** documentation and some test files. To print off a copy of the **dvips** manual, run **dvips.tex** through  $\text{\TeX}$  and print/view **dvips.dvi**.

**Inputs** contains various input files required by **dvips**, such as **config.ps**, **psfonts.map**, **\*.pro** files, **\*.enc** files, etc.

The version of **dvips** included in Oz $\text{\TeX}$  contains a number of enhancements:

- Input text files can have either Mac or Unix line endings.
- The **add\_T\_option** parameter tells Oz $\text{\TeX}$  whether or not to add the **-T** option to the **dvips** command line to specify the paper size using the current **paper\_width** and **paper\_height** values. The **Default** setting is **true**, so you’ll need to change it to **false** (in your **Local** file) if you want to use the “**-t landscape**” option.
- All the Oz $\text{\TeX}$ -specific **\special** commands for including EPSF/PICT/PNTG files are supported. I’ve also added support for Oz $\text{\TeX}$ ’s method of including raw PostScript files followed by arbitrary PostScript code, although this is not something I recommend. Because of conflicts with **dvips**-specific **\special** commands, there are some rare situations where inclusion of a raw PostScript file will no longer work. In such cases you will need to add a **rawps** keyword; e.g., **\special{rawps=foo.ps ...}**.
- **dvips** can download Mac PostScript font files (of type LWFN) as well as **pfa/pfb** files.
- A number of **config.\*** files have been added to **TeX/DVIPS/Inputs/** so that you can pass an extra option like “**-P CMPS**” to tell **dvips** to load **config.CMPS** after **config.ps**; **config.CMPS** contains the line “**p +psfonts.CMPS**” which tells **dvips** to append the info in **psfonts.CMPS** to **psfonts.map**. Note that **psfonts.CMPS** refers to all the **pfb** files stored in the **CM** and **AMS** subfolders of **TeX/Type1/**.

- The TFM names in `psfonts.*` are case insensitive for consistency with Mac file names.

Users of `dvips` on other systems should note that the following environment variables discussed in the `dvips` manual have been replaced by equivalent `OzTeX` parameters:

|                           |                              |  |
|---------------------------|------------------------------|--|
| <code>TEXCONFIG</code>    | = <code>ps_folders</code>    | (to find <code>config.*</code> , <code>psfonts.*</code> )  |
| <code>DVIPSHEADERS</code> | = <code>ps_folders</code>    | (to find <code>*.pro/*.enc/*.pfa/*.pfb/LWFFN</code> files) |
| <code>TEXINPUTS</code>    | = <code>input_folders</code> | (to find <code>\special</code> files)                      |
| <code>TEXFONTS</code>     | = <code>tfm_folders</code>   | (to find TFM files)  |
| <code>VFFONTS</code>      | = <code>vf_folders</code>    | (to find VF files)   |
| <code>TEXPKS</code>       | = <code>pk_files</code>      | (to find PK files)   |

When searching for files, `dvips` always looks in the current folder (the same folder as the DVI file) before looking in the folders specified by the above parameters.

In most operating systems, `dvips` uses the above environment variables to override the settings in `config.ps`. In `OzTeX`'s case, `config.ps` can be used to override the above parameters, but it is not something I'd recommend doing unless you are an experienced `dvips` hacker.

NOTE: `dvips` does not use `OzTeX`'s `printing_resolution` or `printing_mode` parameters. It uses the resolution and mode specified in `config.ps`. However, if the File menu's "Use Standard PostScript" item is ticked then the `printing_resolution` value must match the `dvips` resolution, otherwise the printed output will be the wrong size.

## 14.1 Downloading fonts used in EPS files

A self-contained EPS file includes the data for all non-printer-resident fonts it might require. But that can waste a lot of disk space if you have many EPS files, so it is often better to create EPS files that rely on the printing software to download the font data once.

The automatic downloading of a PostScript font used (but not included) in an EPS file is supported by `dvips`, but only if the font name appears in a `psfonts.*` or `*.map` file along with a "<" entry specifying the `pfa/pfb/LWFFN` file to be downloaded. See the bottom of `psfonts.map` in `TeX/DVIPS/Inputs/` for some examples. If `dvips` cannot find an entry for a required font then it will generate a suitable error message and Courier will be substituted. If you do not have a suitable `pfa/pfb/LWFFN` file for downloading then try using FontForge (<http://fontforge.sourceforge.net/>) to convert your TrueType or OpenType fonts into `.pfa` files.

## 15 Creating PDF files

PDF (Portable Document Format) files are a popular way of distributing high-quality documentation that is both device and platform independent. Adobe provide a freely available program called Acrobat Reader (or more recently, Adobe Reader) that can be used to view PDF files. This section discusses a number of different ways to create a PDF file from a `TeX/LaTeX` file.

### 15.1 PDF via dvips

If you have a `LaTeX` document that you'd like to convert into a PDF file with nice hypertext links then one approach is to add a line like this just before `\begin{document}`:

```
\usepackage[colorlinks]{hyperref}
```

After typesetting the `LaTeX` file a couple of times, run `dvips` with the `-z` option and save the output in a `.ps` file. Then use Ghostscript or Distiller (see below) to convert the output to PDF.

One advantage of this approach is that the HyperTeX links in your DVI file are recognized by OzTeX's previewer. (An annoyance is that all links in the PDF file will be surrounded by a dashed border. This is undesirable if you use the `colorlinks` option, but you can prevent the border by following the instructions in `TeX/DVIPS/Inputs/hps-noborder.pro.`)

Another approach is to use the `pdfmark` option:

```
\usepackage[pdfmark,colorlinks]{hyperref}
```

The resulting DVI file will contain `\special` commands targeted for PDF; see `manual.pdf` in `TeX/Inputs/LaTeX/hyperref/` for details. The biggest disadvantage is that the links in the DVI file are not active in OzTeX's previewer because the `\special` commands don't conform to the HyperTeX standard. If you use `pdfmark` then run `dvips` without the `-z` option.

## 15.2 Ghostscript or Distiller?

After running `dvips` (with or without `-z`) to create a `.ps` file you can use Ghostscript or Distiller to convert it into a `.pdf` file. If you have Ghostscript installed then just select the `ps2pdf` item in the Tools menu. Distiller is a commercial product available from Adobe (see [www.adobe.com](http://www.adobe.com)). The Default file's `extra_tools` parameter shows how to call Distiller or `ps2pdf` from the Tools menu.

Note that PDF files look terrible if bitmap fonts are used, so make sure your DVI file does not use any PK fonts. One solution is to include something like `\usepackage{times}` in your L<sup>A</sup>T<sub>E</sub>X document so that it uses the standard Adobe text fonts. A better solution is to run `dvips` with the `-PCMPS` option so that it loads `config.CMPS`.

## 15.3 PDF via pdfTeX or dvi2pdf

You don't have to use `dvips`/Ghostscript/Distiller to create PDF files. One alternative is to use pdfTeX, a modified version of T<sub>E</sub>X that can generate a PDF file rather than a DVI file. Another alternative is the `dvi2pdf` program which can create a PDF file from a DVI file. If you have teTeX installed then you can call `pdftex` and `dvi2pdf` from OzTeX's Tools menu or via a task list; see sections 5 and 7. You can also call `epstopdf` to convert EPS files to PDF files.

# 16 Graphics

Creating illustrations with standard T<sub>E</sub>X commands isn't easy. It is possible to place small dots anywhere on a page and build up an arbitrarily complex picture, but time and memory limitations make such a scheme impractical (although the Xy-pic package can achieve impressive results). L<sup>A</sup>T<sub>E</sub>X provides a `picture` environment, but it is very inefficient and only suitable for small, simple diagrams.

T<sub>E</sub>X does however provide a `\special` command that can be used to pass arbitrary information to the DVI reader. For example, OzTeX can interpret a `\special` command containing the name of a graphic file created by some other application and then include that file in the preview or printed output.

Using `\special` reduces the portability of your T<sub>E</sub>X documents because DVI-reading programs can only support a limited set of `\special` commands and the few attempts to define a standard set of commands have largely been ignored by most implementors (myself included!). The best way to enhance portability is to hide the actual `\special` call inside a macro; this is the approach taken by L<sup>A</sup>T<sub>E</sub>X and its high-level `\includegraphics` command.

## 16.1 Including Macintosh graphic files

L<sup>A</sup>T<sub>E</sub>X users should never need to use explicit `\special` commands. Here are some examples of the `\includegraphics` command available in the `graphicx` package:

1. To include an EPS image at quarter size:

```
\includegraphics[scale=0.25]{ozicon.eps}
```



2. To set the height to 0.5in and preserve the aspect ratio:

```
\includegraphics[height=0.5in]{ozicon.eps}.
```



3. To set the height to 0.3in and the width to 1in:

```
\includegraphics[height=0.3in,width=1in]{ozicon.eps}.
```



The figure on page 1 of this user guide was created by using

```
\includegraphics[scale=0.72]{folders.eps}
```

to include an extract from a Macintosh screen dump. The scale value of 0.72 was chosen because it is a multiple of  $0.24 = 72/300$ , where 72 dpi is the resolution of the bitmap and 300 dpi is the resolution of my printer. This technique can improve the print quality of scaled bitmaps.

Plain T<sub>E</sub>X users can also use macro packages; see Larry Siebenmann's `boxedeps.doc` and `boxedeps.tex` in `TeX/Inputs/Plain/` or `epsftest.tex` in `TeX/Docs/Plain/`.

The remainder of this section describes OzT<sub>E</sub>X's low-level `\special` support. If you use the above high-level commands to include graphic files then you don't really need to know all the following details.

A `\special` command can appear almost anywhere in an input file. It behaves like an invisible box of zero height and width. T<sub>E</sub>X simply stores the given information in the DVI file at the current page position. OzT<sub>E</sub>X allows the inclusion of three common types of Macintosh graphic files:

- EPS files are the best format to use if you have a choice. There are many good tools for converting graphic files from one format into another. A popular shareware tool is `GraphicConverter`. There are plenty of free Unix tools like `netpbm` and `ImageMagick`. EPS files contain encapsulated PostScript code. If the file has a corresponding PICT resource (with ID = 256) then OzT<sub>E</sub>X can use it for previewing (note that the PICT resource is not needed; see section 16.2). To include an EPS file you can use a command of the form:

```
\special{epsf=filename}
```

OzT<sub>E</sub>X also supports the `dvips` syntax for EPS file inclusion; see section 16.3.

- PICT files can be generated by almost all drawing/painting programs on the Mac. To include a PICT file, use a command of the form:

```
\special{pict=filename}
```

- PNTG files are better known as MacPaint files. They only contain black and white images and aren't very common these days. To include a PNTG file, use a command of the form:

```
\special{pntg=filename}
```

Keywords and file names are case insensitive and spaces before the file name are ignored. For example, `\special{EPSF=F00}` is the same as `\special{epsf= foo}`.

To find a given `\special` file, OzT<sub>E</sub>X uses the same search strategy as that for a T<sub>E</sub>X input file. It looks in the current folder first, and then in the `input_folders` list. The file name can also be a partial path name relative to the current folder, or a full path name. For example,

```
\special{pict=my-pict-files/foo}
```

will include a file called `foo` from a subfolder called `my-pict-files` in the current folder. Spaces and other special characters can be included in a file name or path name by enclosing them in double quotes, but it is not recommended, particularly if  $\TeX$ 's `\input` command might be used to read the file. Avoid using double quotes in a file name.

The `epsf/pict/pntg \special` commands all allow you to include graphics in your DVI file, but if you have a choice it's best to use EPS files because:

1. The output on a PostScript printer should be of much higher quality than a similar PICT/PNTG image.
2. A colored EPS file will print correctly on a PostScript printer (assuming it supports color).  $\text{Oz}\TeX$  converts all PICT files to black and white bitmaps when printed via `dvips`.
3.  $\TeX$  can extract the bounding box information and position the image automatically.
4. EPS files are much more portable than PICT/PNTG files.

When creating a PICT/PNTG file it doesn't really matter where you position the picture;  $\text{Oz}\TeX$  will find the smallest bounding box enclosing all non-white pixels and place the bounding box so that its bottom left corner coincides with the location of the `\special`.

$\text{Oz}\TeX$  does much the same thing for an EPS image; it places the lower left corner of the bounding box at the location of the `\special`, but it makes no attempt to determine the smallest bounding box because the picture's dimensions should match the `BoundingBox` comment in the data fork ( $\text{Oz}\TeX$  will warn you if the dimensions differ by more than 3 bp).

Note that  $\text{Oz}\TeX$  converts a PICT file into a bitmap for previewing, so the results might not be optimal. Ditto for the PICT resource in an EPS file or a corresponding PDF file (see section 16.2). The bitmap's resolution is set to the `image_resolution` value if it's greater than 0. The default value is 144 dpi and there is not much to gain by choosing a higher value —  $\text{Oz}\TeX$  will use more memory, previewing will be slower, and it won't improve the quality of bitmap images originally created at lower resolutions. If `image_resolution` is set to 0 then  $\text{Oz}\TeX$  will use the current `viewing_resolution` value.

For colored EPSF/PICT files,  $\text{Oz}\TeX$  uses the `pixmap_depth` parameter to control the maximum number of colors that can be viewed. This parameter is only used when previewing.

$\text{Oz}\TeX$  allows optional `keyword=value` pairs after the file name in an `epsf/pict/pntg \special`. The keyword can also be terminated by a space. Here are the currently supported keywords:

- `scale=n` – scales a picture by a factor  $n$  ( $n$  must be  $> 0.0$  and  $\leq 100.0$ ).
- `hscale=n` – sets the horizontal scale factor (and alters the aspect ratio).
- `vscale=n` – sets the vertical scale factor (and alters the aspect ratio).
- `width=bp` – sets the horizontal scale to get a picture at the desired width ( $bp$  is a number  $> 0.0$  and  $\leq 32000.0$  in bp units, where  $72\text{ bp} = 1\text{ in}$ ). If no previous `height` keyword has been used then the vertical scale is set to the new horizontal scale to preserve the aspect ratio.
- `height=bp` – sets the vertical scale to get a picture at the desired height. If no previous `width` keyword has been used then the horizontal scale is set to the new vertical scale to preserve the aspect ratio.

If a particular keyword is used more than once in a single `\special` command then the last value is used. An illegal keyword or value will produce an error message when previewing or printing. Synonyms exist for most keywords:

```

epsf      = eps or epsfile
pict      = pictfile
pntg      = macpaint
hscale    = xscale
vscale    = yscale
width     = wd
height    = ht

```

## 16.2 Previewing EPS files

OzTeX will first look for a corresponding PDF file in the same folder as the EPS file; e.g., it looks for `foo.pdf` if the EPS file name is `foo.eps`. If the PDF file exists, and it is *newer* than the EPS file, then OzTeX uses it to display the image.

If there is no corresponding PDF file then OzTeX looks for a PICT 256 resource in the EPS file and will display that. If there is no PDF file and no PICT 256 resource then the `convert_eps` parameter tells OzTeX what to do. The possible values are:

- 0** — Do nothing. The previewer simply displays the bounding box in a color specified by the `missing_picts` parameter, along with the name of the file and the message “no picture” near the `\special` marker.
- 1** — Call the `oz-convert-eps` script in `TeX/Unix/` to create a temporary PDF file which OzTeX can then use to display. This is the default setting. The EPS-to-PDF conversion is done by calling Ghostscript and each EPS file will have a unique PDF file — these temporary files are created in a system-specified directory and are deleted when OzTeX quits.
- 2** — Call the `oz-convert-eps` script (if necessary) and use the temporary PDF file to create a PICT 256 resource in the EPS file. The PICT image resolution will be the current `printing_resolution` value when printing. When previewing, the image resolution is specified by the `image_resolution` parameter if its value is greater than 0 (the default value is 144). If you set `image_resolution` to 0 then the current `viewing_resolution` value is used, but there is usually little to be gained by doing that.
- 3** — Call the `oz-convert-eps` script to create a corresponding PDF file in the same location as the EPS file. The PDF file is permanent and will be used by future runs, as described above. This is handy for `pdflatex` users because it can’t include EPS files, but it can include corresponding PDF files.

OzTeX will automatically re-generate a temporary or permanent PDF file if its modification date becomes older than that of the corresponding EPS file. So if you switch out of OzTeX and modify an EPS file, when you return and re-display the page the image will be updated to the correct version.

**WARNING:** The `oz-convert-eps` script can convert MetaPost output files that use TeX fonts, but it replaces all such fonts with Times-Roman, so your labels might not look correct (especially if they use math or symbol fonts).

### 16.3 Compatibility with other DVI translators

To provide better `\special` compatibility with `dvips` and other DVI translators, `OzTeX` can process `\special` commands like

```
\special{psfile=foo llx=100 lly=100 urx=172 ury=172 rwi=720 rhi=720}
```

where `rhi` is optional (it is used to change the aspect ratio). Note that `llx` and the other keywords shown above can only be used after `psfile`. Their order doesn't matter. It is best *not* to type in such `\special` commands yourself; you should let a macro package generate them.

The given file (`foo` in the above example) should normally be an EPS file containing a `BoundingBox` comment. For compatibility with Wilfried Ricken's `DirectTeX`, `OzTeX` also allows PICT files in a PSFILE `\special`. If `foo` is a file of type PICT then `OzTeX` will read in the picture stored in the data fork. If the given file name ends in `".pict"` then `OzTeX` will get the picture from a PICT 256 resource (and if you put a suitable `BoundingBox` comment in the data fork then `epsf.tex` or `boxedeps.tex` can be used to position the image).

Note that if the given file does not contain a `BoundingBox` comment (PICT files cannot have such a comment) then you will need to supply the bounding box coordinates to `epsf.tex`; for example, `\epsfbox[0 0 72 72]{foo}`. The numbers must be given in bp units and in the following order: lower left x, lower left y, upper right x, upper right y.

### 16.4 Including PostScript files

`OzTeX` allows you to include a "raw" PostScript file using `TeX`'s `\special` command. However, if you have a choice then any of the methods described in the above sections are preferable because `OzTeX` cannot preview such a file.

Assuming you have a file called `fig.ps` that contains PostScript code for generating some sort of figure, the following command can be used to include this figure in a `TeX` or `LATeX` document:

```
\special{fig.ps}
```

When printing the DVI file, `OzTeX` interprets the `\special` argument as a file name and then includes this file with the rest of the PostScript output. If it can't find the file in the current folder then it looks in the `ps_folders` list. Don't worry about removing any `showpage` command at the end of your PostScript file because `OzTeX` temporarily disables it.

It may take a little practice to position the figure correctly. The default PostScript origin is at the bottom left corner of the paper. However, when you include a file, `OzTeX` automatically moves the origin to the position of the `\special`. You can move the figure about by shifting the `\special` position, or by using a `translate` command in the PostScript file. The latter option is faster because you don't have to run `TeX` again. Alternatively, you may prefer to place your figure at an absolute position on the paper. Simply use a command like `initgraphics` in the PostScript file. The position of the `\special` command then becomes irrelevant; just make sure it's on the right page!

`OzTeX` also allows arbitrary PostScript code to appear after the file name. At least one space must be used to terminate the file name; further characters are included as a new line at the *start* of the given file. For example,

```
\special{fig.ps 2 1 scale}
```

could be used to double the width of the figure. This feature allows you to include the same PostScript file more than once, but with a different starting line each time — useful if you need to produce the same diagram many times but with slight variations.

## 16.5 Rotating T<sub>E</sub>X material

Rotating arbitrary T<sub>E</sub>X material is quite simple. L<sup>A</sup>T<sub>E</sub>X users can use the `\rotatebox` command from the `graphics/graphicx` packages; see the examples in `grfguide.dvi`. Plain T<sub>E</sub>X users should look at `TeX/DVIPS/Docs/rotsample.tex`.

## 16.6 Color

OzT<sub>E</sub>X supports the color `\special` commands used by `dvips` and generated by L<sup>A</sup>T<sub>E</sub>X's `color` package. The `hyper.tex` file in `TeX/Docs/LaTeX/` shows how the `color` package can be used. For more details, L<sup>A</sup>T<sub>E</sub>X users should read `grfguide.dvi` and Plain T<sub>E</sub>X users should read the final section in the `dvips` manual.

If your DVI file uses color `\specials` then you should enable the “Prescan color `\specials`” option in the view dialog. Prescanning might be needed because OzT<sub>E</sub>X allows you to jump to any page, but there might be a `\special` on an earlier page that sets the text color (or the background) for all future pages. The prescan option tells OzT<sub>E</sub>X to do a quick scan of `\specials` on all earlier (unseen) pages to make sure that the colors on the current page are correct. Although this scan is quite fast it's probably best to keep the prescan option disabled if you never use color. OzT<sub>E</sub>X remembers the state of the prescan option.

When OzT<sub>E</sub>X sees a page for the first time, the color `\specials` on that page are remembered. As more pages are seen, the prescan will take less time. In fact, if you preview pages 1, 2, 3, etc., then the prescan isn't done at all.

## 16.7 HyperT<sub>E</sub>X

OzT<sub>E</sub>X's previewer supports the HyperT<sub>E</sub>X `\special` commands. These `\specials` let you jump to another place in a document by simply clicking on a link, such as a reference to a section number. The `hyperref` package (in `TeX/Inputs/LaTeX/`) makes it very easy for L<sup>A</sup>T<sub>E</sub>X users to add HyperT<sub>E</sub>X links to their documents. Just add the line `\usepackage{hyperref}` before `\begin{document}` and run L<sup>A</sup>T<sub>E</sub>X a couple of times.

To see how HyperT<sub>E</sub>X links work, preview the DVI files in the `TeX/Docs/LaTeX/` folder. If the mouse moves over a link then the cursor changes to a pointing finger. Clicking on a link will take you to the target. By default, all links are underlined and colored green. If you prefer to use the `[colorlinks]` option with `hyperref` then it's best to add these lines to the OzT<sub>E</sub>X-only section in your Local file:

```
hypertex_links = invisible
underline_links = false
```

The “Back Link” and “Forward Link” items in the View menu make it easy to move to previously visited links; they work just like the Back/Forward buttons in a web browser. Their default keyboard shortcuts are “[” and “]” respectively.

OzT<sub>E</sub>X also supports links to other DVI files on your Mac and to URLs anywhere on the Internet; see page 3 in `hyper.dvi`. Links to URLs call appropriate helper applications.

The HyperT<sub>E</sub>X `\special` commands are also supported by `dvips`. If the `-z` option is given then `dvips` will generate “HyperPostScript”; this can be used by Ghostscript or Distiller to produce a PDF file with hypertext links. For more information, see section 15.

## 17 Recovering from errors

Oz $\TeX$  and the other Oz\* applications all report errors by displaying messages in the main window or in a dialog box. In the latter case there are three types of error dialogs:

1. Warning messages appear in a dialog box with an OK button. In some cases you might want to hit escape or command-dot to cancel the dialog and interrupt the current task; e.g., if Oz $\TeX$  detects lots of `\special` errors while viewing a DVI file (try page 16 in `nasty.dvi`).
2. In a few situations you might get a dialog with Yes and No buttons.
3. Fatal messages appear in a box with a Quit button (which will quit the application). These sort of errors should only occur very rarely.

Most errors are accompanied by a beep. The remainder of this section concentrates on the most likely errors you'll encounter while using Oz $\TeX$ ; it also discusses what you can do to prevent the errors recurring.

### 17.1 Errors while loading a configuration file

If an error occurs while loading in a configuration file then Oz $\TeX$  will display a suitable message in the Oz $\TeX$  window. The error message will also include the line number in the file. It is advisable to switch to your editor, fix the error and reload the configuration file before attempting to do anything else. See section 4.1 for a precise description of the syntax required in a configuration file.

### 17.2 Errors while loading an encoding file

These errors can occur after Oz $\TeX$  has loaded a configuration file containing a list of PostScript fonts. If Oz $\TeX$  complains that it can't find an encoding file, check the `enc_folders` parameter. The encoding files supplied with Oz $\TeX$  are all stored in the `TeX/Encodings/` subfolder. Note that Oz $\TeX$  does *not* look in the current folder for an encoding file.

Various syntax errors are possible when reading an encoding file. The resulting message should provide you with enough information to fix the problem. See the comments at the start of `Mac.enc` for a detailed description of the legal syntax.

### 17.3 Errors while running $\TeX$

Nearly all the errors you'll see when running  $\TeX$  will be the result of mistakes in your input file. *The  $\TeX$ book* [8] is the best place to look for solutions, especially Chapters 6 and 27. Lamport's  $\LaTeX$  manual [11] also has a section devoted to errors. The following errors are more likely to occur due to Oz $\TeX$ 's particular implementation of  $\TeX$ .

#### $\TeX$ capacity exceeded

Such an error message will be accompanied by additional text describing which  $\TeX$  parameter has been exceeded. Before doing anything else, first check for some common errors. If the exceeded parameter is `stack_size` then make sure you haven't defined an infinitely recursive macro.

Look at the current  $\TeX$  parameters by selecting the appropriate "Show" item in the Config menu. If the exceeded parameter appears in the resulting display then you should be able to solve the problem by increasing the parameter's value in the Oz $\TeX$ -only section of your `Local`

file. After saving the change, don't forget to restart OzT<sub>E</sub>X or select Default from the Config menu. Note that if you change the value of `mem_top`, `hash_size`, `hash_prime` or `hyph_size` then you will also need to rebuild all format files (see section 9).

### Fatal format error

You probably changed one of the critical T<sub>E</sub>X parameters in your configuration file and forgot to rebuild the format. Run INITEX and rebuild the offending format file (see section 9).

### Can't find file

There are a number of likely reasons why T<sub>E</sub>X can't find a file:

1. The file does not exist in the current folder or any of the folders in the `input_folders` list. If T<sub>E</sub>X was looking for a format file then make sure it is in one of the folders specified by the `format_folders` parameter.
2. You could have spelt the file name incorrectly.
3. The file name might contain an illegal character. A Macintosh file name can contain just about any character, but T<sub>E</sub>X is not so flexible. Don't use spaces and avoid non-alphabetic characters in all file names that might be seen by T<sub>E</sub>X.

If you see such an error then the best thing to do is hit command-dot or escape to abort T<sub>E</sub>X and either rename the file or move it to the correct folder.

## 17.4 Errors while viewing a DVI file

### `\special` errors

When interpreting a DVI page OzT<sub>E</sub>X will check any `\special` commands and warn you about the following problems:

- An unknown keyword.
- A file that can't be opened.
- An included file's type does not match the first keyword (`epsf`, `pict` or `pntg`). If the file has no type (quite common on OS X) then it is assumed to be:

EPSF if the extension is `.eps` or `.epsf` or `.ps`, or if the file starts with `"%!PS"`;

PICT if the extension is `.pict` or `.pct`;

PNTG if the extension is `.pntg`.

Case is ignored when comparing the extension.

- A bad value after an optional keyword.

No warnings appear if the "Ignore bad `\specials`" option is checked.

### Missing fonts

During interpretation of a DVI page you may get a warning message if OzT<sub>E</sub>X can't find a PK file, or if a PostScript font's matching screen font is not installed. Such warnings will only appear once for each missing font (but not at all if the "Ignore missing fonts" check box was selected). The "Page Info" item indicates all missing fonts. Note that OzMF can be used to build missing PK files on demand; see section 11.1.

### Page off paper

If OzTeX detects that any part of a DVI page is off the paper then it will beep. It will also switch to a full view if the `auto_full_view` parameter is set to `true` (this is the default setting). TeX will often catch these errors (e.g., `overfull \hbox` or `\vbox`) but there are quite a few situations where it can't. TeX really has no idea about what paper size you intend to print its pages on.

If you're sure the problem isn't in your input file then use the "Page Info" item to see what OzTeX thinks the paper dimensions are. You could have forgotten to reset the landscape option or you might have made a mistake when changing the paper dimensions in your `Local` file.

## 17.5 Errors while printing a DVI file

OzTeX detects basically the same set of errors whether it is viewing or printing a DVI file. In the latter case the error messages appear in the OzTeX window along with a beep. If you've been sensible and previewed the DVI file before printing it then you should have discovered nearly all the errors (and fixed them).

To avoid wasting paper you should always view a DVI file before printing it and choose "Page Info" to check for any missing fonts. Another way to check for errors is to run `dvips` and send the output to a file instead of the printer.

### Printer errors

While sending output to any type of printer you might get status messages indicating a problem such as an empty paper tray or a paper jam. Once you've fixed the problem printing can normally continue.

### Bad `\special` command

If OzTeX detects any sort of error in a `\special` command then a suitable message will be displayed in the OzTeX window. See section 16 for the `\special` syntax and semantics supported by OzTeX.

### Can't open `\special` file

OzTeX expects the argument of a `\special` command to contain a file name. It won't be able to open the file if you spelt its name incorrectly or if the file isn't in the correct folder.

### PostScript errors

You shouldn't see any errors in the PostScript code generated by `dvips`, but there may well be problems in a PostScript file included by `\special`. You can always tell which is the culprit by commenting out the `\special` command.

So what can you do? For starters, get the *PostScript Language Reference Manual* by Adobe Systems [1]. Debugging a PostScript program can be a frustrating experience. Try downloading the error handler stored in `TeX/PS/errhandler.ps` so that errors will be printed out on paper (until the printer is turned off). Ghostscript is also a useful tool for locating problems in PostScript code.

## 18 Miscellany

### 18.1 Transferring TFM/VF/PK/DVI files

OzT<sub>E</sub>X reads standard TFM/VF/PK files, and reads and writes standard DVI files. If you have access to T<sub>E</sub>X on some other computer system you should be able to move such files to and fro without any further processing.

For the purposes of data transmission a TFM/VF/PK/DVI file must be treated as a binary file (a stream of arbitrary 8-bit bytes), so make sure your ftp client uses binary mode when getting or putting such files. If you transfer a DVI file to your Mac and a mistake occurs during transmission, or if you forget to send it as a binary file, then OzT<sub>E</sub>X will probably display some sort of error message soon after opening the file.

Note that DVI files contain TFM file names; this can be a cause for concern if you plan to transfer DVI files from one T<sub>E</sub>X system to another system with a different set of TFM files. Another cause of portability problems are `\special` commands. If you have a choice, it's always safer to transfer the original T<sub>E</sub>X input file so you can edit any incompatible font names or `\special` commands.

### 18.2 OzT<sub>E</sub>X CD

OzT<sub>E</sub>X is available on CD. The CD contains a full OzT<sub>E</sub>X system (of course!) all unpacked and ready to run, the complete L<sup>A</sup>T<sub>E</sub>X distribution, heaps of extra fonts and macro packages, lots of useful T<sub>E</sub>X and Mac-related software, all my latest shareware, plus other goodies. For more information on the CD, including how to order it, visit [www.trevorrow.com/oztex/ozcd.html](http://www.trevorrow.com/oztex/ozcd.html). Note that there is a substantial discount for registered users.

### 18.3 Getting more information

The T<sub>E</sub>X Users Group produce an excellent magazine called TUGboat. Once a year they also distribute a free copy of the “T<sub>E</sub>X Live CD” to all members; this is a wonderful resource chock full of useful stuff. It's worth joining TUG just to get the CD. See the “T<sub>E</sub>X Users Group” item in OzT<sub>E</sub>X's Help menu for contact details.

If you have access to the Internet then another good source of help and information about T<sub>E</sub>X-related topics is the newsgroup `comp.text.tex`. The following web sites are also worth visiting:

- T<sub>E</sub>X Users Group: <http://www.tug.org/>
- Gary Gray's T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X software for the Mac: <http://www.esm.psu.edu/mac-tex/>

More links can be found in the “Related Software” item in OzT<sub>E</sub>X's Help menu.

The next page has a list of recommended reading material. For Plain T<sub>E</sub>X users and all serious T<sub>E</sub>X hackers, [8] is essential. For L<sup>A</sup>T<sub>E</sub>X users, [11] and [4] should be adequate for all but the largest typesetting jobs. Also included are a number of other books recommended by OzT<sub>E</sub>X users; see [5], [10], [3], [13], [2] and [12].

## References

- [1] Adobe Systems Incorporated. *Postscript Language Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.
- [2] Wolfgang Appelt. *TEX für Fortgeschrittene*. Addison-Wesley, Reading, MA, USA, second edition, 1994.
- [3] Michael Doob. *TEX: Starting from 1*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1993.
- [4] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, Reading, MA, USA, second edition, 1994.
- [5] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion: Illustrating Documents with TEX and PostScript*. Tools and Techniques for Computer Typesetting. Addison-Wesley, Reading, MA, USA, 1997.
- [6] George Grätzer. *Math into L<sup>A</sup>T<sub>E</sub>X: an introduction to L<sup>A</sup>T<sub>E</sub>X2e and AMS-L<sup>A</sup>T<sub>E</sub>X 1.2*. Birkhäuser, Basel, Switzerland, 1996.
- [7] Donald Knuth. Virtual fonts: More fun for grand wizards. *TUGBoat*, 11(1):13–23, April 1990.
- [8] Donald E. Knuth. *The TEXbook*. Addison-Wesley, Reading, MA, USA, 1984.
- [9] Donald E. Knuth. *The METAFONT book*, volume C of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- [10] Helmut Kopka and Patrick W. Daly. *A Guide to L<sup>A</sup>T<sub>E</sub>X: Document Preparation for Beginners and Advanced Users*. Addison-Wesley, Reading, MA, USA, third edition, 1999.
- [11] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, second edition, 1994.
- [12] Norbert Schwarz. *Einführung in TEX*. Addison-Wesley, Reading, MA, USA, second edition, 1988. Available in English and Dutch translations.
- [13] Michael D. Spivak. *The Joy of TEX — A Gourmet Guide to Typesetting with the AMS-TEX macro package*. Addison-Wesley, Reading, MA, USA, 2nd revised edition, 1990.

## Index

- #, 25
- \$c, 11, 26
- \$f, 40
- %&format, 15
- \*, 25
  
- Acrobat Reader, 47
- add\_fonts, 37, 39, 42
- add\_T\_option, 46
- Adobe Reader, 47
- AFM files, 37, 43
- AFM2TFM, 43
- aliases, 23
- AMS- $\text{\LaTeX}$ , 1
- AMS- $\text{\TeX}$ , 1
- AppleTalk printers, 9
- atprint, 9
- Auto Detect Format, 15
- auto\_full\_view, 7, 56
  
- background typesetting, 4
- base files, 43
- begin\_fonts, 39, 42
- binary files, 57
- bypass\_print\_dialog, 10
- bypass\_view\_dialog, 5
  
- calling Unix, 28
- Clipboard, 3, 13, 14
- color, 16, 53
- command\_keys, 27
- comp.text.tex, 57
- config\_files, 11, 20, 21, 26, 27
- config\_folders, 11, 21, 27
- configuration files, 22
  - Add Outline Fonts, 21, 42
  - creating your own, 26
  - Default, 20, 26
  - editing Local, 20
  - loading, 20
  - Local, 26
  - location of, 1, 22
  - search paths, 25
  - showing parameters, 20
  - supplied, 21
  - syntax, 22
- convert\_eps, 51
- current\_app, 26, 37
  
- delete\_files, 13
- deleting files, 13
- Distiller, 48
- drop.config, 34
- drop\_tex/dvi/ps, 33
- dummy\_screen\_font, 42
- dummy\_tfm\_file, 37
- DVI files
  - portability, 57
  - printing, 9, 12, 56
  - transferring, 57
  - viewing, 5, 16, 55
- dvi2pdf, 31
- DVImcopy, 19
- DVIDVI, 19
- dvipdfm, 48
- DVIPS, 19, 46
  - config.CMPS, 46, 48
  - config.ps, 45–47
  - downloading PostScript fonts, 40
  - Hyper $\text{\TeX}$  support, 53
  - low-level dialog, 9
  - missing PK fonts, 17
  - Oz $\text{\TeX}$  enhancements, 46
  - psfonts.CMPS, 46
  - psfonts.map, 39, 40, 46, 47
  - reencoding PostScript fonts, 40
  - user manual, 46
- dvips\_options, 11
- DVItoMP, 45
  
- editing  $\text{\TeX}$  files, 3
  - choosing an editor, 3
  - creator code, 15
- enc\_folders, 39, 42, 54
- encoding files, 42
  - location of, 1
  - Mac.enc, 42, 54
  - Mac8r.enc, 3, 39
- end\_fonts, 37, 39, 42
- EPS files
  - fonts used in, 47
  - previewing, 51
- epstopdf, 34, 48
- errhandler.ps, 57
- errors
  - during installation, 3

- in configuration files, 54
  - in encoding files, 54
  - in PostScript files, 56
  - in `\special` commands, 55, 56
  - printer errors, 56
  - recovering from, 54
  - while previewing, 6, 55
  - while printing, 56
  - while running `TEX`, 54
- extensionless files, 15
- `extra_tools`, 20, 28
- Extras folder, 2
- FontForge, 31, 47
- `fontinst`, 37
- `fontlist.ps`, 40
- fonts, 37–40, 42
  - Computer Modern, 41
  - encoding files, 42
  - hidden fonts, 31, 42
  - PK files, 1, 38, 44
  - PostScript, 39–42
  - screen fonts, 42
  - TFM files, 37
  - TrueType, 42
  - virtual fonts, 42
- forcing PostScript printing, 12
- format files
  - automatic detection, 15
  - big formats, 36
  - building, 36
  - custom formats, 36
  - forcing a format, 4, 15
  - location of, 1
- `format_files`, 16, 36
- `format_folders`, 36, 55
- ftp, 57
- GFtoDVI, 43
- GFtoPK, 43
- GFtype, 43
- Ghostscript, 20, 28, 48, 51, 57
- GraphicConverter, 49
- graphics, 48, 49, 52, 53
  - EPS files, 49
  - including PostScript, 52
  - PICT files, 49
  - PNTG files, 49
- `graphics` package, 35
- guided tour, 3
- `hash_prime`, 55
- `hash_size`, 55
- help files, 21
  - location of, 1
- `help_files`, 21
- `help_folders`, 19–21
- hidden fonts, 31, 42
- HTML, 2, 46
- Hyper`TEX`, 18, 53
- `hypertex_links`, 53
- `hyph_size`, 55
- `image_resolution`, 50, 51
- ImageMagick, 49
- `\includegraphics`, 49
- INIMF, 43, 45
- INIMP, 45
- INITEX, 1, 16, 36, 55
- `input_folders`, 15, 24, 35, 47, 49, 55
- installation, 1
- keyboard shortcuts, 7, 27
- `LATEX`, 35
  - installing a new package, 35
  - `LATEX 2.09`, 35
  - `LATEX 2e`, 35
- `latex.ltx`, 36
- `load_config`, 11, 27
- Local config file, 1, 20, 26
- Local-Notes config file, 21, 26
- `lpr`, 9
- LW`FN` files, 40, 46
- MacGhostView, 48
- MacGSView, 20, 48
- Make`TeXPK`, 43–45
- `MakeTeXPK.base`, 43, 45
- `MakeTeXPK.mf`, 43, 45
- making missing PK fonts, 44
- mem files, 45
- `mem_top`, 55
- menus, 12
  - Config, 20
  - Edit, 13
  - File, 12
  - Help, 21
  - `TEX`, 14
  - Tools, 19
  - View, 16
- Metafont, 2, 43

- mode, 45
- MetaPost, 2, 45
- MF folder, 2, 22, 43
- missfont.make, 44
- missing\_fonts, 42
- missing\_fonts\_file, 44
- missing\_picts, 51
- modes.mf, 45
- mouse shortcuts, 7
- MP folder, 2, 22, 45
- MPtoTeX, 45
- netpbm, 49
- OS X, 31
- oz-convert-eps script, 51
- oz-mac2unix program, 29
- OzMF, 2, 43
  - keeping TFM files, 45
- OzMP, 2, 45
- OzTeX CD, 35, 41, 57
- OzTeX folder, 1
- OzTeX window, 3, 4
  - bringing to front, 17
  - changing color, 14
  - changing font, 14
  - changing font size, 14
  - copying, 13, 20
  - saving, 13
  - text smoothing, 14
- OzTtH, 2, 46
- page positioning, 11
- page setup, 12
- paper coordinates, 5
  - changing units, 5
- paper\_height, 11
- paper\_width, 11
- path names
  - full, 22, 23
  - in config files, 22
  - in input files, 24
  - partial, 22, 23
  - with spaces, 22
- Path syntax
  - Mac, 23
  - Unix, 23
- PDF files, 47
- pdfTeX, 48
- PfaEdit (now called FontForge), 47
- PICT 256 resource, 51
- pixmap\_depth, 50
- PK files, 38
  - location of, 1
  - making on demand, 44
  - missing, 17, 56
  - transferring, 57
- pk\_file, 44
- pk\_files, 38, 44, 47
- PKtype, 43
- PLtoTF, 43
- portability, 24, 48
- PostScript files
  - including via \special, 46, 52
  - location of, 2
  - manipulating, 20
  - sending to printer, 12
- PostScript fonts, 39
  - AMS, 41
  - Computer Modern, 41
  - downloading, 40
  - location of, 2
  - partial downloading, 40
  - reencoding, 39
  - with non-PostScript printer, 41
- preferences, 11
- previewing, *see* viewing DVI files
- printing DVI files, 9, 12
  - forcing PostScript printing, 10
  - low-level PostScript, 9
  - non-PostScript, 9
  - standard PostScript, 10
- printing text files, 12
- printing\_mode, 45, 47
- printing\_resolution, 10, 47
- ps2pdf, 48
- ps\_creator, 20
- ps\_folders, 35, 40, 47, 52
- PSbook, 19
- psnfss package, 35, 39
- PSnup, 19
- PSselect, 20
- PStoPS, 20
- Read-Me, 1
- recent files
  - checking, 13
  - saving, 13
  - showing paths, 13
- Register, 2
- remembering recent files, 13

- removing recent files, 13
- REVTeX, 1
- rotation, 53
- screen fonts, 42
- scripting, 33
- search paths, 25
- send\_ps, 9
- \special, 7, 16, 46, 48, 49, 51–53, 57
- src \special, 8
- src\_specials, 8
- stack\_size, 54
- standard PostScript, 10, 12
- starting, 3
- subfolder expansion, 25
- symbolic links, 23
- SymbolicLinker, 24
- task lists, 33
- teTeX, 28, 31
- TeX, 4, 14
  - OzTeX enhancements, 14
- TeX capacity exceeded, 54
- TeX folder, 1, 22
- TeX input files
  - boxedeps.tex, 49
  - editing, 3
  - epsftest.tex, 49
  - gentle.tex, 2
  - hyper.tex, 1, 53
  - location of, 1
  - nasty.tex, 1, 3, 4
  - ozuser.tex, 1
  - plain.tex, 36
  - rotsample.tex, 53
- TeX Live CD, 57
- TeX origin, 5, 11
- TeX parameters, 4, 54
  - for big format, 36
- TeX to HTML, 2, 46
- TeX Users Group, 57
- TeX/Configs/ folder, 1, 21, 27, 35
- TeX/Docs/LaTeX/ folder, 1, 3, 53
- TeX/Docs/Plain/ folder, 1, 3, 41, 49
- TeX/DVIPS/ folder, 2, 46
- TeX/Encodings/ folder, 1, 42, 54
- TeX/Formats/ folder, 1
- TeX/Help/ folder, 1
- TeX/Inputs/ folder, 1, 41, 49
- TeX/PK/ folder, 1
- TeX/PS/ folder, 2
- TeX/TFM/ folder, 1, 37, 39
- TeX/Type1/ folder, 2, 46
- TeX/Unix/ folder, 2
- TeX/VF/ folder, 1, 39
- text\_editor, 3
- TEXTtoPS.pro, 12
- TFM files, 15, 37
  - creating, 37, 44
  - location of, 1, 37
  - missing, 17
  - transferring, 57
- tfm\_folders, 15, 35, 37, 38, 45, 47
- TFtoPL, 43
- to\_printer, 10
- transferring files, 57
- TrueType fonts, 21, 31, 41
- TtH, 2, 46
- TtH folder, 2, 22, 46
- TUGboat, 57
- underline\_links, 53
- Unix, 28
- Unix support, 15, 43
- unix\_path, 30
- unix\_shell, 30
- upgrading, 1
- Use texmf tree, 31
- VF files
  - transferring, 57
- vf\_folders, 35, 42, 47
- VFtoVP, 43
- view window, 5
  - bringing to front, 17
  - closing, 8
  - title bar, 7, 17
- viewing DVI files, 5, 16
  - automatic update, 8
  - default view, 5, 19
  - ignoring bad \specials, 16
  - ignoring missing fonts, 16
  - keyboard shortcuts, 7
  - landscape, 16
  - making missing PK fonts, 16
  - mouse shortcuts, 7
  - overriding DVI magnification, 16
  - page information, 17
  - prescanning for color, 16
  - scrolling, 6

- selecting pages, 7
  - with EPS files, 51
  - zooming, 6, 18, 19
- viewing\_mode, 45
- viewing\_resolution, 5
- virtual fonts, 42
  - location of, 1
- VPtoVF, 43
  
- web sites, 57
- write\_8\_bit, 15
  
- XRay, 28
- Xy-pic, 48