

OzT_EX User Guide

Version 1.9, May 1995

Andrew Trevorrow
akt@kagi.com

OzT_EX is a shareware version of T_EX for the Macintosh (see OzT_EX's Help menu for details about the shareware fee). This document assumes you know how to use a Macintosh. It also assumes you are familiar with T_EX, the typesetting system developed by Donald Knuth at Stanford University. An understanding of PostScript is not essential but would be useful if you have access to a PostScript printer.

OzT_EX aims to provide a standard T_EX environment for the Macintosh that can easily be extended or customized to suit your particular needs. If you have had experience with T_EX on some other computer then the way OzT_EX works shouldn't be too surprising.

Comments, bug reports and suggestions are all welcome, but please note that I only reply to e-mail queries from registered OzT_EX users.

Contents

1	Introduction	1
2	Configuration files	2
2.1	The default configuration file	2
2.2	The required syntax	2
2.3	The parameters	3
2.3.1	Important folders and files	3
2.3.2	Previewing colours	4
2.3.3	T _E X parameters	4
2.3.4	PostScript fonts	4
2.4	Other configuration files	4
3	A guided tour of OzT_EX	5
3.1	Starting OzT _E X	5
3.2	Choosing an editor	5
3.3	Editing a T _E X input file	5
3.4	Running T _E X	6
3.5	Checking for virtual fonts	6
3.6	Viewing a DVI file	7
3.6.1	Setting the default view	7
3.6.2	Paper coordinates	8
3.6.3	Zooming in and out	8
3.6.4	Scrolling around	9
3.6.5	Checking for errors	9
3.6.6	Selecting pages	9
3.6.7	Keyboard short cuts	10
3.7	Printing a DVI file	10
3.7.1	Non-PostScript printing	11
3.7.2	Low-level PostScript printing	11
3.7.3	Standard PostScript printing	12
3.8	Positioning pages correctly	13
3.9	Continuing from where you left off	13
4	OzT_EX's menus	14
4.1	The File menu	14
4.2	The Edit menu	16
4.3	The T _E X menu	17
4.4	The View menu	18
4.5	The Config menu	21
4.6	The Help menu	22
5	L^AT_EX	23
5.1	Using L ^A T _E X 2.09	23
6	Building format files	24
6.1	Building big formats	24

7	Fonts	25
7.1	TFM files	25
7.2	Creating new TFM files	25
7.3	PK files	26
7.4	PostScript fonts	27
7.5	Macintosh screen fonts	28
7.6	Encoding files	29
7.7	Virtual fonts	29
8	OzTools	30
9	Transferring TFM/VF/PK/DVI files	30
10	Including graphics	31
10.1	Including Macintosh graphic files	31
10.2	Compatibility with other DVI drivers	34
10.3	Including PostScript files	34
10.4	Rotating T _E X material	35
11	Recovering from errors	36
11.1	Errors when loading a configuration file	36
11.2	Errors when loading an encoding file	36
11.3	Errors when running T _E X	36
11.4	Errors when viewing a DVI file	37
11.5	Errors when printing a DVI file	38
12	Changing resource data	39
13	OzT_EX and Metafont	39
13.1	Setting the Metafont mode	40
14	OzT_EX and Dvips	41
14.1	Why use Dvips?	42
15	More information	42

1 Introduction

Assuming you correctly carried out the installation instructions in the **Read-Me** file supplied with Oz \TeX , you should have a folder that looks something like this:

Here is a brief description of the various folders:

- The **TeX-inputs** folder contains a large number of \TeX input files stored in various sub-folders. \TeX looks in these sub-folders if it can't find a given file in your current folder.
- The **TeX-formats** folder contains the format files read by \TeX and the **TeX.pool** file read by INITEX. Section 6 describes how to create format files.
- The **TeX-fonts** folder contains many \TeX Font Metric files stored in various sub-folders. These TFM files are required by \TeX to carry out typesetting; see section 7.1.
- The **VF-files** folder contains Virtual Font files; see section 7.7.
- The **PS-files** folder contains various PostScript files. It also contains encoding files in the **Encodings** sub-folder; see section 7.6.
- The **PK-files** folder contains PK font files stored in various sub-folders. The name of each sub-folder indicates the “size” of the PK files kept within it; see section 7.3 for how this size is calculated and for more information about PK files. Oz \TeX is distributed with a fairly small set of PK files (sufficient to print out the documentation on the most common printers). New PK files at any desired size can be built by METAFONT; see section 13.
- The **Configs** folder contains a number of configuration files. Section 2 discusses how these files can be used to customize Oz \TeX .
- The **Help-files** folder contains a number of text files. The information in these files can be displayed by selecting their names from Oz \TeX 's Help menu.
- The **TeX-docs** folder contains examples of \TeX input files, including **nasty.tex** which is used in the guided tour of Oz \TeX . It also contains **gentle.tex**, the source file for Michael Doob's *A Gentle Introduction to \TeX* . \TeX beginners should find this a useful companion to *The \TeX book* by Donald Knuth.
- The **LaTeX-docs** folder contains the source files for this User Guide.

- The `BEdit Lite` folder contains the `BEdit Lite` application and its documentation. `BEdit Lite` is a text editor by Rich Siegel which can be used to create and modify `TeX` input files. If you prefer to use a different text editor then throw away the `BEdit Lite` folder.
- The `Dvips` folder contains Tom Kiffe’s implementation of `dvips`; see section 14.
- The `Metafont` folder contains Tom Kiffe’s implementation of `METAFONT`; see section 13.

Please note that Tom’s programs are distributed with `OzTeX` under a separate shareware fee; see the `Read-Me` file for details. System 7 or later is needed to run these programs.

2 Configuration files

2.1 The default configuration file

`OzTeX` reads a configuration file called `Default` when starting up. This file is kept in the `Configs` folder along with other configuration files. A configuration file is a simple text file which you can edit and change various parameters that control `OzTeX`’s behaviour.

Before modifying the `Default` config file it would be a good idea to save a copy first. Another good idea is to flag your changes with some sort of unique comment so that they can be reproduced easily if and when a new version of `OzTeX` is released.

2.2 The required syntax

Follow these rules when editing a configuration file:

- Lines starting with “%” are 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 appears in the `OzTeX` window. All other lines are said to be “significant”.
- Each significant line must contain at least one “=” character. Any characters before the first “=” are ignored, as are spaces or tabs immediately after it. The rest of the line contains the information used by `OzTeX`, usually in the form of one or more strings. If only one string is required then any further information is ignored. Use double quotes around a string that includes one or more spaces.
- Do *not* change the order of significant lines (except between the first and last “==”). `OzTeX` expects to find all the parameters in a fixed sequence.
- To avoid long significant lines you can use “|” to split lines anywhere you like. Upon reading a “|”, `OzTeX` 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.
- To help keep non-default configuration files short and simple, use these special characters:
 1. If “?” is the first character of a parameter’s value then the parameter is not changed and the rest of the line is ignored.
 2. “!” is similar to “?” but in addition all remaining parameters are left unchanged (in fact the rest of the file is ignored).

Note that these characters are not recognized as special in the `Default` file because every parameter must be given a default value, even if empty.

2.3 The parameters

This section discusses all the parameters in a configuration file, in the order required by OzTeX. It would be a good idea to look at the `Default` config file while reading the following information.

The list of config files appear as items at the end of the `Config` menu and must correspond to text files stored in the `Configs` folder. This list is ignored in non-default configuration files. The default config file will automatically load any config files whose names start with “+”.

The list of format files appear as items at the end of the `TeX` menu and must correspond to the names of `.fmt` files in the `TeX-formats` folder. The format file names must not contain any spaces. Place your preferred format first.

The list of help files appear as items in the `Help` menu and must correspond to text files in the `Help-files` folder. The `Help` menu will not appear if the list is empty.

NOTE: Any string starting with “-” in the above three lists will result in a disabled line in the corresponding menu.

The printer resolution (in dots per inch) is used when printing a DVI file and the viewing resolution is used when viewing a DVI file. In both cases the maximum value is 3000 and the minimum value is the resolution of your screen (typically 72). Both resolution values are followed by a string specifying the mode that will be used by METAFONT when making missing PK fonts; see section 13.1.

The units parameter tells OzTeX how to display dimensions when printing or viewing a DVI file. The normal TeX units are available (`in`, `cm`, `mm`, `pc`, `pt`, `bp`) as well as `px` (see section 3.6.2).

The paper width and height dimensions are used to detect page-off-paper errors when viewing a DVI file and during low-level PostScript printing. (In standard PostScript printing and non-PostScript printing, OzTeX uses the paper size and orientation selected in the “Page Setup” dialog.) If the width is greater than the height then OzTeX will print or view in landscape mode. The `Default` values specify A4 paper in portrait mode. If you use US Letter paper then see the `US Letter` config file for suitable default values.

The horizontal and vertical offsets affect the positioning of all pages when printing a DVI file. They should only be changed if your printer has problems positioning the TeX origin (see page 2 in `nasty.dvi`). These offsets are *not* used when viewing a DVI file.

The “show statistics” and other flag parameters specify the initial settings of some check boxes used in the dialogs for printing a DVI file.

2.3.1 Important folders and files

OzTeX needs to know the names and locations of various folders and files. These are specified by path names. A path name can be a simple file name, or it can be a sequence of names separated by colons that represent a valid location in your folder hierarchy. If the path name ends with a colon then it specifies a folder, otherwise a file. OzTeX will generate a warning message if a given path name does not specify a known folder or file. Note that under System 7, OzTeX supports aliased files but not aliased folders.

Partial path names begin with a colon. The first folder in such a path is assumed to be in the same location as the OzTeX application. This is the best way to specify folders and files because then you can move the entire OzTeX folder anywhere you like without having to change any path names.

Full path names begin with the name of a disk; e.g., `HD:OzTeX:Help-files:`. Path names must be enclosed in double quotes if they contain any spaces; e.g., `"My help:"`.

OzTeX can be told to search in more than one folder for TeX input files, TFM files, PK files, PS files and VF files. The corresponding parameters also allow “*” at the end of a path name to tell OzTeX to include all sub-folders of the given folder in the search path. OzTeX does *not* automatically include sub-folders within sub-folders.

2.3.2 Previewing colours

A number of elements can be coloured when previewing a DVI file; they are the paper edges, `\special` markers and missing fonts. The allowed colours are black, red, green, blue, cyan, magenta, yellow and white (white can be used to make something invisible). On black and white monitors all non-white colours appear black.

2.3.3 T_EX parameters

The T_EX parameters determine how much memory is allocated for various arrays used by T_EX and INITEX. Note that `mem_max` is temporarily set to `mem_top` whenever you run INITEX.

Comments in the `Default` file indicate the possible range of values for each parameter. The `Default` values are suitable for small to medium-size T_EX documents. If you decide to increase one or more of these parameters then you may need to increase OzT_EX's application memory size (via "Get Info" in the Finder's File menu). Note that a change to `mem_top`, `hash_size`, `hash_prime` or `hyf_size` will require rebuilding all format files; see section 6.

2.3.4 PostScript fonts

The lines between the first and last "==" tell OzT_EX which TFM files are for PostScript fonts (or more accurately, non-PK fonts). The ordering of lines doesn't matter; they are used by OzT_EX to build a list of PostScript fonts. In a non-default config file the first "==" can be replaced by "=+"; this tells OzT_EX to *update* the current list rather than delete it and build a new one.

Each line specifies a TFM name, a printer font name, an optional download file, a screen font name, an encoding entry and an optional style. Note that most of the lines in the `Default` file contain "raw" TFMs for corresponding virtual fonts; see section 7.7.

If a DVI file uses PostScript fonts, the TFM names get converted into printer font names when printing the file on a PostScript printer. OzT_EX allows non-resident PostScript fonts to be downloaded on demand. The Utopia lines at the end of the `Default` file have an extra string (starting with "<") after the printer font name. This specifies the file to be downloaded if the corresponding font is used; see section 7.4 for more information.

The screen font names are used when viewing a DVI file with PostScript fonts, and when printing the file on a non-PostScript printer. If you are unsure about which screen font would best match a particular PostScript font, you must still specify something (such as "?"). If OzT_EX can't find a specified screen font it will warn you and use the system font instead. To obtain the best results when viewing or printing a DVI file with PostScript fonts you must get the matching screen fonts.

Each screen font can appear in a certain style defined by a set of flags: "b" for bold, "i" for italic, "u" for underline, "o" for outline, "s" for shadow, "c" for condense and "e" for extend. If the style is not specified then OzT_EX uses the font's plain style.

An encoding entry is the name of a user-defined encoding file (or "nil" if the TFM file and screen font use the same encoding). Section 7.6 discusses encoding files.

2.4 Other configuration files

OzT_EX's Config menu lets you switch rapidly from one configuration file to another. This is especially useful if you have access to more than one printer because each printer will probably require a separate configuration file. There are many other uses for configuration files; see section 4.5 for a brief description of the various config files supplied with OzT_EX. It is a simple matter to create your own files and add their names to the list in the `Default` file.

3 A guided tour of OzTeX

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

3.1 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 the path name of the `Default` configuration file just loaded. The `Default` file's list of PostScript fonts contains references to an encoding file, so it will also be loaded.

If you made a mistake during installation, or when editing the `Default` file, then you might also get one or more error messages. The messages should give you some idea of how to fix the problem.

The OzTeX window is always present; note that it has no close box in the top left corner. This window provides a terminal-like interface for OzTeX. It cannot be edited, but you can copy its contents to the Clipboard or save it to a text file.

You can also start up OzTeX by double-clicking on any OzTeX-created file ending in `.xxx`; e.g., `foo.log` or `foo.dvi`. The advantage of this start up method is that `foo.tex` will appear in the TeX menu and `foo.dvi` will appear in the Print and View menus. Also, if `foo.log` exists and contains a known format then OzTeX will automatically make this the current format so you can immediately typeset `foo.tex` by hitting Command-T. If `foo.tex` can't be found then OzTeX will look for `foo.ltx` (this is a popular file naming scheme used to distinguish L^ATeX input files). Under System 7, you can drag-and-drop `foo.tex` or `foo.ltx` onto the OzTeX icon and OzTeX will immediately typeset the file.

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.

3.2 Choosing an editor

OzTeX does not have an integrated text editor, but if you are using System 7 or later you won't find this too much of a problem. After installing OzTeX and starting it up for the very first time, select "Choose Editor" from the Edit menu and use the standard file dialog to locate and select the application you'll be using to edit your TeX/L^ATeX input files. If the chosen application supports Apple events then OzTeX remembers its name and location for use in all future OzTeX sessions (or until you choose another editor).

Having chosen an editor, OzTeX will use it as the target of an "open file" event 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 4.2 for more details about the items in the Edit menu.

3.3 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` folder. There is nothing special about this folder, or `LaTeX-docs`. I simply like to keep my TeX and L^ATeX input files in separate folders. You can keep your input files anywhere you like.

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` 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.

3.4 Running TeX

Let's now typeset `nasty.tex`. Before starting TeX we need to make sure that the correct format will be used. (This check is not normally needed if the log file exists and "Use Format in Log" is ticked. However, when typesetting a file for the very first time it is best to manually set the correct format.) 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`.

Choose "TeX..." from the TeX menu and open `nasty.tex`. The OzTeX window will be cleared, TeX will start up, load the Plain format, and begin reading the given input 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* explains what you can do in such a situation. (Section 4.2 describes how you can type "e" to tell OzTeX to switch to your editor and open the file containing the error, possibly with the error line selected.) 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 standard 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.

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. Whenever OzTeX finishes a lengthy task in the background (like typesetting or printing), or if some sort of user interaction is required (say a TeX error occurs), then under System 6 or later you will be notified by a flashing OzTeX icon in the menu bar. A diamond mark will also appear next to OzTeX in the list of current applications.

3.5 Checking for virtual fonts

Before proceeding to view or print a DVI file created for the first time, it is a good idea to check and see if any virtual fonts are used (section 7.7 has more information about virtual fonts). Select "Fix VFs in DVI" from the TeX menu and open `nasty.dvi`. OzTeX lists all the fonts used in the DVI file and prefixes each font with its type:

PK indicates a packed pixel font (see section 7.3);
PS indicates a PostScript font (see section 7.4);
VF indicates a virtual font.

As it happens, `nasty.dvi` does use a couple of virtual fonts, so OzTeX processes every page, fixes all virtual font references and creates a new version of `nasty.dvi`.

If your documents use virtual fonts, it is advisable to tick “Fix VFs after TeX” in the TeX menu; then you don’t have to remember to run “Fix VFs in DVI” before viewing or printing a DVI file.

3.6 Viewing a DVI file

The next step is to preview the DVI file and check for problems that TeX may 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...” on page 18 for details.) Play around with the scroll bar if you like but make sure the 2nd DVI page is selected before proceeding. To view this page simply click in the View button or hit the Return key.

The 2nd page in `nasty.dvi` contains a single rule with some text below it. Dotted lines representing the paper edges should also be visible. 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 will remember the current window settings and use them the next time it starts up.

3.6.1 Setting the default view

OzTeX displays the top left corner of the paper 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/or 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.

To restore the initial default view, simply select “Actual Size” from the View menu, move the scroll bar thumb boxes up and to the left as far as possible, then select “Save as Default View”.

3.6.2 Paper coordinates

An understanding of the coordinate system used by OzTeX to view (and print) a DVI page would be useful. The px units mentioned earlier are “paper pixels”; they are OzTeX’s internal units. The resolution parameters in a configuration file define the number of paper pixels per inch; one parameter is only used when printing a DVI file and the other is only used when viewing a DVI file.

OzTeX uses a coordinate scheme in which 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 paper coordinates) in a box at the lower left corner of the view window. You can click in this box to change units.

3.6.3 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 change the current scale factor:

- “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” halves the current scale factor.
- “Zoom Out” doubles the current scale factor.

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

- Click-and-drag within the viewing area to zoom in to the selected rectangle. If you want to cancel the operation then make the rectangle very thin (not small, because that might be interpreted as a simple click).
- A simple click will zoom in by halving the scale factor.
- Command-click will zoom in by decrementing the scale factor.
- Option-click will zoom out by incrementing the scale factor.
- Shift-click will zoom out by doubling the scale factor.
- Option-Command-click will temporarily magnify a small area under the cursor until the mouse button is released. The scale factor in the magnified area is 1.
- A beep occurs if you can’t zoom in or out any further.

When you zoom in or out by clicking, OzTeX will try to make the point you clicked the middle of the new view. However, when you choose a menu item, OzTeX tries to keep the top left page location fixed. 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.

3.6.4 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".

3.6.5 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. It also lists any `\special` commands on the current page, showing their locations and arguments.

3.6.6 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 always displayed in the view window's title bar. 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. Pages 13 to 15 illustrate this behaviour. When you move from such a page to a normal page, OzTeX switches back to the default view.
- OzTeX may take a little while to interpret and display a selected page, depending on how complicated it is. Hit Command-C or Command-Dot if you get bored.
- The view window is updated in the following manner. Visible paper edges are drawn first, then `\special` bitmaps and markers, then rules, then characters on a font by font basis, starting with the font that has the least number of characters on the page. Try a full view of page 19.
- Page 3 has examples of `\special` commands. OzTeX is able to preview included files of type PICT, PNTG or EPSF; see section 10 for more information. The location of each `\special` is indicated by a small 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 7.4.
- All the fonts used on page 12 are deliberately missing.
- Page 16 has examples of bad `\special` commands.

When you have finished previewing the DVI file just click in the view window's close box. The view window is also closed automatically when you run TeX, print a DVI file or load a configuration file.

3.6.7 Keyboard short cuts

OzTeX provides a number of keyboard short cuts to make it easier to proofread a DVI file:

- Type “g” to go to a given page and switch to the default view.
- Type “f” to move to the first page. The current scale and location do not change.
- Type “l” to move to the last page. The current scale and location do not change.
- Type “n” to move to the top of the next page without changing the current scale or horizontal position.
- Type “b” to move to the bottom of the previous page without changing the current scale or horizontal position.
- Hit the space bar to scroll forwards through a document.
- Hit the Delete 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.
- 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.

Note that if the page selected by any of the above short cuts is off the paper then OzTeX will beep and switch to a full view.

3.7 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. Choosing the second item simply avoids the standard file dialog.

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 four(!) methods of printing a DVI file:

1. Non-PostScript printing.
2. Low-level PostScript printing.
3. Standard PostScript printing.
4. Using `dvips`.

Of course, most people will only ever use one or two of these methods. The printing method chosen by OzTeX depends on whether or not the current printer (selected in the Chooser) is a PostScript device, and the current state of two flags in the File menu: “Use Standard PostScript” and “Use DVIPS”.

If “Use DVIPS” is ticked then OzTeX ignores both the current printer and the “Use Standard PostScript” flag. Instead, it will call `dvips` to translate the DVI file into a PostScript file, as described in section 14. If “Use DVIPS” is not ticked then one of the following three printing methods will be selected.

3.7.1 Non-PostScript printing

If the current printer is not a PostScript device then the “Use Standard PostScript” flag is ignored. OzTeX will open the DVI file and display a printer-specific dialog. (If the printer resolution parameter does not match the actual resolution of the current printer then you’ll first be warned about possible scaling problems; see page 39.) For example, if the current printer is a StyleWriter then you’ll get a dialog like this:

The dialog box lets you change various options specific to the current printer. OzTeX adds some extra items at the bottom of the dialog:

- Some text showing the first and last DVI/TeX pages. This information can be useful when entering DVI page numbers in the “From” and “To” edit boxes.
- Edit boxes for changing the page increment and DVI magnification.
- Check boxes for reversing the page order, showing statistics and making missing PK fonts.

Click on the Print button or hit the Return key to start printing. At any time while OzTeX is printing you can switch to another application and OzTeX will continue to print in the background.

3.7.2 Low-level PostScript printing

If the current printer is a PostScript device and “Use Standard PostScript” is *not* ticked then OzTeX will open the DVI file and display the following dialog:

This dialog box lets you change a variety of print options. Most of them should be fairly obvious; details can be found in the description of “Print DVI...” on page 14. The option you’ll most often want to change 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. Click in the Print button or hit Return to start printing.

If the printer is found then a status box will appear and keep you up-to-date on your job’s progress and the state of the printer. As each page is translated into PostScript its DVI/TEX page numbers are displayed in the OzTEX window.

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. (OzTEX may start translating page 18 before the error is seen.)

The above dialog box also lets you send the PostScript output to a file instead of the printer. This file is an ordinary text file with a default name of `nasty.ps`. A warning: the PostScript code generated by OzTEX is resolution-dependent. If you send this code to a PostScript device with a resolution different to that of the current printer resolution parameter then the output might look a little strange.

A note for PostScript hackers: OzTEX begins the PostScript output by copying the contents of a file called `DVItoPS.pro` (kept in the `PS-files` folder). OzTEX will also look for a file called `global.ps` in the current folder and, if found, include its contents at the end of `DVItoPS.pro`. For example, `global.ps` in the `PS-files` folder shows how you can get the word “DRAFT” printed in the background on every page.

3.7.3 Standard PostScript printing

If the current printer is a PostScript device and “Use Standard PostScript” *is* ticked then OzTEX will open the DVI file and display the standard dialog for that printer:

Below the normal options for the current printer, OzTEX adds some extra items (the same ones described under the dialog box for non-PostScript printing on page 11). Click on the Print button or hit the Return key to start printing.

PostScript hackers should note that when standard PostScript printing is used to print a DVI file, OzTEX does *not* use `DVItoPS.pro`; instead it uses a file called `OzTeXdict.pro` (also kept in

the `PS-files` folder). The PostScript code in `OzTeXdict.pro` is very similar to `DVItoPS.pro`, but there are some subtle differences. I decided it would be unsafe to allow inclusion of a `global.ps` file when using standard PostScript, but I have allowed for similar tricks by making OzTeX look for `OzTeXdict.pro` in the current folder before looking in the PS folder(s).

Although there are significant advantages to using standard PostScript, there are some disadvantages. There is a limitation in LaserWriter print drivers older than version 8 that causes a PostScript error when printing all but the simplest of DVI files. OzTeX is able to predict when this error is likely to occur and will display a suitable warning message. Also, standard PostScript code takes longer to print and saved PostScript files are a little bit larger. If you want to use OzTeX's more efficient low-level printing code then leave "Use Standard PostScript" unticked. Another advantage of low-level PostScript printing is that any PostScript error messages will appear in the OzTeX window (although on some printers, like the LaserWriter Select 310, you might need to send `geterrors.ps` first).

In standard PostScript printing, OzTeX has to go through all selected DVI pages first and determine font and character usage. All the PK character bitmaps used in the document are then downloaded at the beginning, so the first page in a long document might take some time to be printed.

3.8 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. If the printed page shows a significant discrepancy then adjust the horizontal and vertical offsets in the `Default` configuration file to shift the origin to the correct position.

Note that if your printer is set up for US Letter paper then before adjusting the offsets you should first change the `Default` file's paper width to `8.5in` and the paper height to `11in`.

If you use more than one printer then you will probably need a separate configuration file for each printer. You might also need to create a separate configuration file for landscape printing.

3.9 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 TeX file, the format, the DVI file and the most recently previewed DVI page, including the scale and location for use by the "o" keyboard short cut. 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 default values then simply delete `OzTeX Prefs` or move it into a folder where 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.

`OzTeX Prefs` is created in the Preferences sub-folder under System 7 and later, or in the System folder itself in earlier System versions. If you like, you can move `OzTeX Prefs` to the same folder as OzTeX; I prefer this because it simplifies my backup process.

4 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.

4.1 The File menu

Print DVI...

Prints a selected DVI file. After selecting the DVI file you'll be presented with a dialog box that depends on the current printer and the settings of "Use Standard PostScript" and "Use DVIPS" (these flags are described later). The dialog box for low-level PostScript printing (see page 11) lets you change these printing options:

- The page range. The first and final pages are controlled by separate scroll bars; their current values are displayed in the form "DVI page/[TeX page]".
- The paper orientation (landscape or portrait).
- The page order (reverse or normal).
- Printer memory management (conserve VM or not). VM is PostScript's "virtual memory" and the output generated by OzTeX can consume an awful lot of it, especially if your document uses a large number of non-PostScript fonts. If you get a "VMerror" when printing a document then try again with the conserve VM option checked. You should only need to do this for a very unusual document, like a font catalog.
- The paper source (manual feed or normal input tray).
- Whether or not to show statistics about font, rule and `\special` usage.
- Whether or not to make missing PK fonts.
- The number of copies of each page.
- The page increment. The primary purpose of this option is to simplify the printing of a document on both sides of the paper. If the first page is 1 and an increment of 2 is chosen then OzTeX will print pages 1, 3, 5, 7, etc. Depending on how your printer stacks the output, you can then put these pages back into the input tray and print the DVI file again using the same increment but starting from the second DVI page (and reversing the page order; this will not effect *which* pages are printed but simply changes the *order* in which they are done).
- The DVI magnification (expressed as an integer 1000 times the desired magnification). You should use the default value unless you know what you're doing. If the DVI file uses non-PostScript fonts then you should probably stick to the numbers corresponding to TeX's `\magstep` values; i.e., 1000, 1095, 1200, 1440, etc. See Chapters 4 and 10 in *The TeXbook* for details. A document using only PostScript fonts can be printed at almost any magnification (OzTeX allows values from 1 to 10000).

What happens next depends on which dialog button you select. The default Print button will send the PostScript output to the current printer. A status box will appear and indicate what is happening to your job and warn you about any printer problems. Or you might prefer to save the PostScript output to a given text file. 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 dialog boxes for standard PostScript printing (see page 12) and non-PostScript printing (see page 11) let you change various options specific to the current printer. OzTEX adds some extra items at the bottom of the dialog so you can change the page increment or DVI magnification, and decide whether or not to reverse the page order, show statistics, or make missing PK fonts.

Regardless of which printing method OzTEX decides to use, the messages displayed in the OzTEX window are almost identical. The DVI/TEX page numbers are displayed as each page is processed. Any error messages or statistics apply to the most recently displayed page numbers (although this is not true for messages sent back by a PostScript printer because these occur asynchronously). At any time you can hit Command-C or Command-Dot to cancel printing, or you can switch to another application and OzTEX will continue to print in the background.

Print ?.dvi

Prints the indicated DVI file. You can print the file even after moving to another folder. This item is initially disabled; the file name changes after you select a DVI file via “Print DVI...” or “View DVI...”, or create a DVI file by running TEX.

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 PS-files folder. If you know a little about PostScript then you might like to modify TEXTtoPS.pro to suit your own needs. For example, the tab width is defined in TEXTtoPS.pro.

If OzTEX uses standard PostScript printing, or non-PostScript printing, then you'll get a printer-specific dialog with an extra edit box that lets you specify the tab width. The default tab width is 3; this is the value used in various text files supplied with OzTEX. 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 or non-PostScript printing. If you select OK then the dialog settings are remembered for later OzTEX sessions. However, if the Chooser is used to select a different printer then the settings will be restored to default values for the new printer.

Note that in low-level PostScript printing, OzTEX ignores the “Page Setup” settings. In this case the paper size is determined by config file parameters.

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. (However, if the currently chosen printer is *not* a PostScript device then OzTEX ignores this flag and switches to non-PostScript printing.)

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. You can also select other handy features like 2-up printing if you have the LaserWriter 8 print driver.

Use DVIPS

This item is only enabled under System 7 or later. Tick this flag if you want OzTEX to call dvips to translate a DVI file into PostScript. The top two items in the File menu change to

“DVIPS...” and “DVIPS foo.dvi” when “Use DVIPS” is ticked. For more information about `dvips`, see section 14.

If “Use DVIPS” is ticked then OzTEX ignores the “Use Standard PostScript” flag. Of course, “Use DVIPS” is ignored when printing a text file.

Send PostScript ...

Sends a selected text file, presumably a PostScript program, to the current printer. (If it isn't a PostScript printer then you won't get far!) The `PS-files` folder contains a number of interesting files that can be sent to any PostScript printer.

If a PostScript file is sent to spooling software rather than directly to the printer then you probably won't see any messages sent back to the OzTEX window. In such an environment it's a good idea to send `errhandler.ps` first so that any PostScript errors will be printed on paper.

Save OzTEX Window ...

Saves the 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 the front window.

Quit

Quits OzTEX.

4.2 The Edit menu

OzTEX does not have an integrated text editor, so the standard items in this menu are normally disabled (although Cut/Copy/Paste/Clear can be used in a dialog box with edit items).

Copy to Clipboard

Copies the contents of the OzTEX window to the Clipboard so you can paste it into a text file. This item is disabled if the OzTEX window is empty or not the front window.

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 the front window.

Size

Selects the point size of the font used in the OzTEX window. (The available sizes are stored in a MENU resource in OzTEX, so use ResEdit if you want to change the supplied values.) This item is disabled if the OzTEX window is not the front window.

Choose Editor ...

This item is only enabled under System 7 or later. It lets you select the target application (presumably an editor) to which OzTEX will send “open file” events. If the chosen application can receive such events then OzTEX remembers its name and location for later use (as described below).

Select Error Line

This item is only enabled under System 7 or later. If ticked, this flag tells OzTeX to include an extra parameter in the “open file” event sent when you type “e” in response to TeX’s “?” prompt (which normally occurs after a TeX error). The extra parameter specifies the line number so your editor can automatically select the line containing the error. However, some editors can’t handle the extra parameter and won’t even open the file. You will need to experiment with your editor to see if it can select the error line; if it can’t then make sure “Select Error Line” is *not* ticked. Note that BBEdit Lite can’t handle the extra parameter but the full version of BBEdit can. Recent versions of Alpha can.

Edit ?.tex

This item is only enabled under System 7 or later. If selected, OzTeX will send an “open file” event to your current 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).

4.3 The TeX menu**TeX...**

Typesets a selected input file using the current format (or the format in the corresponding log file if “Use Format in Log” is ticked; see below for more about this item). The OzTeX window acts like a terminal during a TeX session.

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* will apply to the version of TeX run by OzTeX (and the same goes for Leslie Lamport’s *L^ATeX* book if you use that format). Only a few special features have been added in OzTeX:

- Most of TeX’s capacity parameters are set at run-time in a configuration file rather than at compile-time. If you get a “TeX capacity exceeded” error then you should be able to overcome the problem by increasing the appropriate parameter (see page 37).
- TeX can be told to look in more than one folder when searching for an input file or TFM file (by setting the appropriate config file parameters).
- You can interrupt a TeX session at any time by typing Command-C or Command-Dot. Depending on what it is currently doing, TeX usually responds immediately with a suitable message and the “?” prompt. If you hit Command-C or Command-Dot 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 current format (or the format in the corresponding log file if “Use Format in Log” is ticked). This item remembers the most recently typeset file.

Use Format in Log

If this item is ticked, then before typesetting `foo.tex` OzTeX will look in `foo.log` to see which format should be used. (In log files created by OzTeX the format appears after the “&” on line 2.) If the log file exists and the format is known (i.e., exactly matches one of the formats at

the bottom of the `TeX` menu) then `OzTeX` automatically changes the current format if necessary. If the log file does not exist or does not contain a known format then the current format will be used. The state of “Use Format in Log” is remembered when you quit `OzTeX`.

Note that if you manage to create a log file with a known but incorrect format then you'll have to turn off “Use Format in Log” and manually select the correct format before typesetting the corresponding input file.

Fix VFs in DVI...

Selects a DVI file and replaces all virtual font references with appropriate characters and commands from actual fonts. See section 7.7 for more information about virtual fonts.

Fix VFs after TeX

If this item is ticked then `OzTeX` automatically fixes any virtual fonts in the DVI file created at the end of a `TeX` run. The state of this flag is remembered when you quit `OzTeX`.

INITEX

Runs `INITEX`, a special version of `TeX` normally used to create format files; see page 24.

List of formats

The items below `INITEX` are read from the format list specified in the most recently selected configuration file. The `Default` config file sets the format list to `LaTeX` and `Plain`.

The current format is indicated by a tick and will be the one used when you typeset a file (although this will be overridden if “Use Format in Log” is ticked and a log file exists). To change the current format, simply select a different format item. `OzTeX` automatically ticks the first format found when loading a configuration file, so it's a good idea to place your preferred format first. Note that if a non-default configuration file uses the “?” character to avoid changing the format list then `OzTeX` won't change the current format.

4.4 The View menu

View DVI...

Previews a selected DVI file. `OzTeX` 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 7) 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 `TeX` 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.
- Two more check boxes can be used to tell `OzTeX` 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 `TeX` system, otherwise you might get a lot of annoying warning messages. `OzTeX` remembers these check box settings.

- The last check box can be used to make missing PK fonts (but it will be ignored if “Ignore missing fonts” is selected). *OzTeX* remembers this check box setting. See section 13 for more details.
- The DVI magnification. You should only alter the default value if you plan to print the DVI file at a different magnification. See the description of “Print DVI...” in section 4.1 for more details. Note that *OzTeX* does not remember any change 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 will be appended to the DVI file name in the view window's title bar and *OzTeX* will locate the requested page, interpret it and display it. You might have to wait a moment before seeing anything, especially if it is a complicated page.

OzTeX sets the initial scale factor and page location to values determined by the default view (see section 3.6.1). However, if any part of the page is off the paper then *OzTeX* will beep and show a full view instead (see the “Full View” item below).

View ?.dvi

Previews the indicated DVI file. Exactly the same sequence of events described above will occur. You can preview the file even after moving to another folder. This item is initially disabled; the file name changes after you select a DVI file via “Print DVI...” or “View DVI...”, and after *TeX* creates a DVI file.

Show *OzTeX* or Show View

Brings either the *OzTeX* window or the view window (if open) to the front. The Return key is a keyboard short cut 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 *TeX* page (trailing counters with zero values are not shown).
- The total number of pages, viewing 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”.
- The number of rules on the 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 3.6.2. If the `\special` command includes a valid PICT/PNTG/EPST file then the width and height of the image's bounding box is also displayed (if there is an image).

Previous Page

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

Next Page

Displays the next DVI page. This item is disabled if the view window is closed 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 closed or if the DVI file only has one page.

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 closed 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. The top left corner of the view window will still show the same page location, unless shifting is necessary to keep within the scrolling limits. This item is disabled if the view window is closed 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 switch to a configuration file that sets the viewing resolution to some multiple of the screen resolution; you'll also need a matching set of PK files. Note that $360 = 72 \times 5$.

Zoom In

Halves the current scale factor. The top left corner of the view window will still show the same page location, unless shifting is necessary to keep within the scrolling limits. This item is disabled if the view window is closed or already displaying the page at minimum scale factor.

Zoom Out

Doubles the current scale factor. The top left corner of the view window will still show the same page location, unless shifting is necessary to keep within the scrolling limits. This item is disabled if the view window is closed or already displaying the page at maximum scale factor.

Save as Default View

Saves the current scale factor and page location as the default view (see section 3.6.1). This item is disabled if the view window is closed.

4.5 The Config menu

Show

Shows the current values of configuration parameters in the *OzTeX* window. You can choose to show all parameters or just some important subsets. If you select “All” then the output is displayed with the correct syntax for a configuration file; this makes it easy to create a new configuration file by simply saving the *OzTeX* window.

Load Config...

Lets you select and load a configuration file (not necessarily in the `Configs` folder) without having to add it to the list of configuration files in the `Default` file. 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.

Other configuration files

The remaining items in the Config menu are determined by information given in the `Default` configuration file. The items must correspond to the names of text files in the `Configs` folder. Selecting one of these items causes the corresponding configuration file to be loaded. A tick appears next to the most recently loaded file. Here is a brief description of all the non-default config files supplied with *OzTeX*:

Big TeX — Sets all the `TeX` parameters to very large values. It also sets the list of formats to `Big-LATeX` and `Big-Plain`. See section 6.1 for more details.

StyleWriter — Configures *OzTeX* for the StyleWriter printer. The printer and viewing resolutions are set to 360 and both `METAFONT` modes to `stylewriter`. The reverse page order parameter is set to `true` because this printer stacks pages face up.

ImageWriter — Configures *OzTeX* for the ImageWriter printer. The printer and viewing resolutions are set to 144 and both `METAFONT` modes to `imagewriter`.

Linotronic — Configures *OzTeX* for a Linotronic typesetter. The printer resolution is set to 1270 and the mode to `linohi`. The viewing resolution and mode are not changed.

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.

View at 144 dpi — Sets the viewing resolution to 144 and the mode to `imagewriter`. The printer resolution and mode are not changed. Characters from 144 dpi PK files scale better on most Mac screens (normally 72 dpi). Previewing will also be faster and use much less memory.

View CM using PS — Tells *OzTeX* to preview Computer Modern text fonts using the nearest “equivalent” PostScript screen fonts. If you have the corresponding TrueType or ATM (Adobe Type Manager) fonts, your text will be much more readable, apart from some strange kerning and the odd incorrect character (e.g., you won't see any “ffi” ligatures). It

is important to realise that this is a bit of a hack and you should not try to print documents using this config file, so remember to switch back to the `Default` config file before printing. The viewing resolution is set to 360 (5×72) so that “Actual Size” dimensions will be accurate.

Add BaKoMa Fonts — Updates the current list of PostScript fonts so that OzTeX can use the BaKoMa fonts created by Basil Malyshev. These fonts are free PostScript and TrueType versions of Computer Modern. They can be obtained by ftp from any CTAN site. Get `/tex-archive/fonts/cm/ps-type1/bakoma/mac/00readme.mac` first.

Add CM/PS Fonts — Updates the current list of PostScript fonts so that OzTeX can use the CM/PS fonts available from Blue Sky Research. The CM/PS fonts are PostScript versions of the Computer Modern fonts. The screen fonts provide a much more readable preview.

Add Lucida Fonts — Updates the current list of PostScript fonts so that OzTeX can use the Lucida fonts available from Y&Y. Lucida is an alternative to Computer Modern, including math fonts, in Adobe Type 1 format.

Add MathTime Fonts — Updates the current list of PostScript fonts so that OzTeX can use the MathTime fonts available from The TeXplorators Corporation. MathTime is a set of math fonts suitable for use with Times Roman.

NOTE: Before using any of the above four config files you might need to change the path name that tells OzTeX where to find the font files that must be downloaded when printing to a PostScript device. These files are normally kept in the Fonts sub-folder of your System folder in System 7.1, or in the Extensions sub-folder in System 7.0. OzTeX doesn't really care where you keep them, but it does need to be told where to find them.

Some config files do not appear in the Config menu but can be loaded from the `Configs` folder:

Remove Help Menu — Removes OzTeX's Help menu.

Old LaTeX — Configures OzTeX to typeset with the old L^AT_EX 2.09 format; see section 5.1.

New LaTeX — Configures OzTeX to typeset with the new L^AT_EX format; see section 5.1.

You can of course modify any of these config files to suit your own needs. If you've created your own config files and you think other people might find them useful, please e-mail them to me and I'll add them to the standard OzTeX distribution.

4.6 The Help menu

The items appearing in this menu are read from the help list in a configuration file and must correspond to the names of text files in the `Help-files` folder. 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 more items to the Help menu. For example, you might like to create a file of TeX/L^AT_EX commands to refer to while editing an input file. If you want to add more help files, or modify the existing ones, then there are a few things to watch out for. Don't use tabs, and avoid long lines if you have a small screen. If you create a new help file then remember to update the `Default` configuration file.

5 L^AT_EX

OzL^AT_EX is distributed with the latest implementation of L^AT_EX (formerly called L^AT_EX 2e). This new version of L^AT_EX is described in the 2nd edition of *L^AT_EX: A Document Preparation System* by Leslie Lamport and *The L^AT_EX Companion* by Michel Goossens, Frank Mittelbach and Alexander Samarin. Both books are published by Addison-Wesley.

A number of important L^AT_EX packages are provided. The `psnfss` package makes it easy to switch to PostScript text fonts instead of Computer Modern. The `graphics` package makes it easy to include Macintosh graphic files. See the documentation and test files in the `PSNFSS` and `Graphics` sub-folders in `TeX-inputs`. Other common packages are kept in the `Tools` sub-folder. More information about L^AT_EX can be found in the `.tex` files in the `LaTeX` sub-folder.

The L^AT_EX system included with OzL^AT_EX is *not* a full distribution (which is way too big for most OzL^AT_EX users). After installing the base files and the above packages, I deleted all the `.dtx`, `.ins` and `.fdd` files, as well as a few other files that were irrelevant for OzL^AT_EX users. If you need a file or package not provided in OzL^AT_EX, then you can ftp it from any CTAN site or mirror. The following table should help:

Sub-folder in <code>TeX-inputs</code>	Corresponding CTAN directory
<code>LaTeX</code>	<code>/tex-archive/macros/latex/unpacked</code>
<code>Graphics</code>	<code>/tex-archive/macros/latex/packages/graphics</code>
<code>PSNFSS</code>	<code>/tex-archive/macros/latex/packages/psnfss</code>
<code>Tools</code>	<code>/tex-archive/macros/latex/packages/tools</code>

If you only need one or two files then ask me first and I might have a copy I can send by e-mail, but you must be a registered OzL^AT_EX user!

5.1 Using L^AT_EX 2.09

The old L^AT_EX 2.09 system is no longer distributed with OzL^AT_EX. L^AT_EX users should seriously consider switching to the new L^AT_EX; it can typeset L^AT_EX 2.09 documents in “compatibility” mode. If you have the old L^AT_EX system from an earlier version of OzL^AT_EX and you want to use it alongside the new L^AT_EX, then carry out these steps:

1. Delete all log files for your existing L^AT_EX documents so that OzL^AT_EX won’t get the wrong format by looking in a log file.
2. Edit the `Default` config file and insert “New LaTeX” and “Old LaTeX” anywhere in the list of config files.
3. Take the `LaTeX` sub-folder out of your *old* `TeX-inputs` folder, rename it `Old LaTeX` and put it next to (*not* inside) the new `TeX-inputs` folder. If you want to store the `Old LaTeX` folder somewhere else, or use a different name, then you’ll have to change the appropriate path in the `Old LaTeX` config file.
4. Build `OldLaTeX.fmt` (no space in the file name!):
 - (a) Start up OzL^AT_EX and select `Old LaTeX` from the Config menu.
 - (b) Run `INITEX`, type “`lplain\dump`” and hit Return.
 - (c) Save the format as `OldLaTeX.fmt` in the `TeX-formats` folder.

Before typesetting a L^AT_EX 2.09 document you must remember to select `Old LaTeX` from the Config menu, and before typesetting a new L^AT_EX document you must select `New LaTeX`.

6 Building format files

A format file contains a pre-compiled set of macros which $\text{T}_{\text{E}}\text{X}$ can load in very quickly. To build a format file you need to run `INITEX` (in the $\text{T}_{\text{E}}\text{X}$ menu). You may need to rebuild all format files if you edit a configuration file and change certain $\text{T}_{\text{E}}\text{X}$ parameters to suit your typesetting needs or your Mac's memory capacity. Building a format file is quite easy; here are the steps used to create `Plain.fmt`:

1. Run `INITEX`, wait for the “**” prompt, type “`Plain\dump`” and hit the Return key. `INITEX` will begin reading `plain.tex` from the `Plain` sub-folder in `TeX-inputs`.
2. You'll eventually get a standard dialog box allowing you to save the format information in the file and folder of your choice. The default file shown will be `Plain.fmt`. The best location for format files is the `TeX-formats` folder.
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 `latex.ltx` from the `LaTeX` sub-folder in `TeX-inputs`.
2. The dialog box will show `LaTeX.fmt`, so just save this file in the `TeX-formats` folder.
3. Delete `LaTeX.log` if you want to.

If you decide to rename one of the existing formats, or add a completely new format, then remember to update your configuration file(s).

6.1 Building big formats

Some people with very large or complicated documents will need to build big versions of the `Plain` or $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ formats. Here's how:

1. Read the comments at the top of the `Big TeX` configuration file. It sets most of the $\text{T}_{\text{E}}\text{X}$ parameters to very large values, but you may want to increase or reduce them depending on how much memory you have. It also sets the list of formats in the $\text{T}_{\text{E}}\text{X}$ menu to `Big-LATEX` and `Big-Plain`. You can change these to anything you like; just make sure your format names don't contain any spaces.
2. After making any changes to `Big TeX`, click on the `OzTeX` icon and use “Get Info” in the Finder's File menu to set `OzTeX`'s preferred memory size to a suitable value (use 3100K if you didn't make any changes).
3. Restart `OzTeX` and select `Big TeX` from the Config menu.
4. Run `INITEX` and type in the appropriate string for the desired format: “`Plain\dump`” for `Big-Plain` or “`LaTeX.ltx`” for `Big-LATEX`.
5. When the dialog box appears, change the format file name to match the name listed in `Big TeX` (`Big-Plain.fmt` or `Big-LaTeX.fmt`) and save the file in the `TeX-formats` folder.
6. Delete the log files if you don't want to keep them.

When you want to use a big format, select `Big TeX` from the Config menu first, then select the desired format from the $\text{T}_{\text{E}}\text{X}$ menu. If you'll be using big formats all the time then you should copy the parameter settings from `Big TeX` into the `Default` configuration file and remove `Big TeX` from the config file list.

7 Fonts

Oz \TeX 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 sub-sections discuss all these font sources in some detail.

7.1 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 \TeX book* if you want to know more about how \TeX uses fonts, especially Chapter 4 and Appendix F.

Oz \TeX reads TFM files when printing or viewing a DVI file containing any of the fonts listed at the end of your configuration file. The dummy TFM file specified in your configuration file will also be read if a requested PK file can't be found (PK files are discussed in section 7.3).

Oz \TeX will look for a TFM file in the current folder before looking in the sub-folders in **TeX-fonts**. All the TFM files in the **CM** sub-folder belong to the Computer Modern family of fonts (created by Donald Knuth using METAFONT). The TFM files in the **LaTeX** sub-folder are used by \LaTeX . The TFM files in the **PS** sub-folder correspond to PostScript fonts; their names follow a scheme devised by Karl Berry in an attempt to make \TeX input files more portable. See section 7.4 for more information about these PostScript fonts.

7.2 Creating new TFM files

If the Oz \TeX 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 it is a PostScript font then you might have to search a bit harder; using Karl Berry's scheme it might be called something like `pfo.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 at any resolution you like). See section 13.
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. Alan Jeffrey's `fontinst` package can convert an AFM file into a PL or VPL file which OzTools can then convert into a TFM file (see page 30).
4. As a last resort you could even create a PL file from scratch. A very tedious and error-prone job though.

Having found or created a new TFM file, you need to place it in a suitable folder so that Oz \TeX can find it. If you put it in one of the sub-folders in **TeX-fonts** (or create a new sub-folder) then you won't need to change the TFM folder(s) in the **Default** config file.

If the TFM file is for a PostScript font then you must add a suitable entry to the list of fonts at the bottom of the **Default** config file. The examples in the file will show you what to do.

7.3 PK files

PK (packed pixel) files store the character images needed to print or view a DVI file containing fonts *not* listed at the end of your configuration file. For each such font there is usually a number of PK files, each one representing the same font but at a different size. This size (which has no relation to the font's design size) is calculated as follows:

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

where *resolution* is either the printer 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 and should equal (± 1) one of the folder names in `PK-files` if you decide to keep PK files of the same size in separate folders.

Although $\text{T}_{\text{E}}\text{X}$ allows you to request a font at virtually any magnification, it is obviously impossible to provide an infinite number of PK files. The compromise solution is to provide a range of sizes for each font. These sizes are in a geometric ratio based on powers of 1.2 and correspond to $\text{T}_{\text{E}}\text{X}$'s `\magstep` values. See Chapters 4 and 10 of *The $\text{T}_{\text{E}}\text{X}$ book* for more details. For example, assuming a *resolution* of 300 and no document magnification:

$\text{T}_{\text{E}}\text{X}$ input	<i>size</i>	PK file
<code>\font\ra=cmr10</code>	300×1.2^0	<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>	300×1.2^1	<code>cmr10.360pk</code>
<code>\font\rd=cmr10 scaled\magstep2</code>	300×1.2^2	<code>cmr10.432pk</code>
<code>\font\re=cmr10 scaled\magstep3</code>	300×1.2^3	<code>cmr10.518pk</code>
<code>\font\rf=cmr10 scaled\magstep4</code>	300×1.2^4	<code>cmr10.622pk</code>
<code>\font\rg=cmr10 scaled\magstep5</code>	300×1.2^5	<code>cmr10.746pk</code>

$\text{T}_{\text{E}}\text{X}$'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 $\text{OzT}_{\text{E}}\text{X}$ will use `cmr10.518pk`. Note that $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ has no `\magnification` command — use the `11pt` and `12pt` options to increase the size of all fonts in a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ document.

For compatibility with other $\text{T}_{\text{E}}\text{X}$ systems, $\text{OzT}_{\text{E}}\text{X}$ uses a fairly complicated search scheme when looking for a font file. If `foo` is a PostScript font then $\text{OzT}_{\text{E}}\text{X}$ looks for `foo.tfm` in the current folder before looking in `TeX-fonts`. If `foo` is not a PostScript font then $\text{OzT}_{\text{E}}\text{X}$ looks for a PK file at the requested size. For example, if the calculated size is 300 then $\text{OzT}_{\text{E}}\text{X}$ will look for:

1. `foo.300pk` (in the current folder)
2. `:PK-files:300:foo` (relative to the location of $\text{OzT}_{\text{E}}\text{X}$)
3. `:PK-files:300:foo.300pk` (ditto)
4. `:PK-files:foo.300pk` (ditto)

In fact, at each step $\text{OzT}_{\text{E}}\text{X}$ will also add and subtract 1 from 300 because rounding errors can occur in the size calculation. If still not found, $\text{OzT}_{\text{E}}\text{X}$ gives up and displays a warning.

There is still another possibility to consider: a font name can include an explicit location. For example, your $\text{T}_{\text{E}}\text{X}$ input file might contain `\font\xxx=:myfonts:foo`. In this case $\text{T}_{\text{E}}\text{X}$ will look for `:myfonts:foo.tfm` in the current folder; that is, it expects to find a file called `foo.tfm` in a sub-folder called `myfonts`. If not found then it gives up; it does not look for

`foo.tfm` in `TeX-fonts`. Similarly, when you print or view the DVI file, `OzTeX` will only look for `:myfonts:foo.tfm` if `foo` is a PostScript font, or for `:myfonts:foo.300pk` (or `301pk` or `299pk`) if `foo` is a PK font. Note that the use of an explicit font location is not recommended as it decreases the portability of your `TeX` input file.

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 might need to get the correct PK files from some other source. Or build them yourself using `METAFONT` as described in section 13.

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 sub-folders, `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 360×1.2^5 .

7.4 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 at the end of a configuration file (see the `Default` file).

The PostScript TFMs stored in the `PS` sub-folder in `TeX-fonts` are meant to be used in conjunction with the virtual font files stored in the `VF-files` folder (section 7.7 has more information about virtual fonts). These TFMs and VFs were created for the `psnfss` package by Sebastian Rahtz using Alan Jeffrey's `fontinst` macros. 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 T1 (or Cork) encoding is the new `TeX` text font encoding for 256-character fonts. The T1 TFMs and VFs have names ending in "q". See `test0.tex` in the `PSNFSS` sub-folder in `TeX-inputs` for how to select either encoding.

Virtual PostScript fonts like `ptmr` should not appear in a config file's list of PostScript fonts. Only the corresponding "raw" font (`ptmr0`) should be listed. Note that raw TFMs are indicated by names of the form `*0.tfm`. When `OzTeX` is fixing VFs it will replace characters from `ptmrq.tfm` with characters from `ptmr0.tfm` according to the mapping defined in `ptmrq.vf`.

Every PostScript printer has a certain set of resident fonts. To see an alphabetical list of the PostScript fonts available in your current printer, choose "Send PostScript..." from the File menu, open the `PS-files` folder, and send the file `getfonts.ps`. The list should appear in the `OzTeX` window. If you'd prefer to print the list then send `fontlist.ps`.

The font you wish to use may not reside in the current printer, so `OzTeX` provides a mechanism for downloading non-resident PostScript fonts. For example, the following entry appears in the list of PostScript fonts at the bottom of the `Default` file:

```
= putr0    Utopia-Regular <putr.pfa    Utopia    Mac.enc
```

The file to be downloaded appears immediately after the "<" character. When `OzTeX` translates a DVI file into PostScript, it checks if any fonts need to be downloaded. If `putr0` is used on a selected page then the above entry tells `OzTeX` to include `putr.pfa` in the PostScript output. `OzTeX` looks for `putr.pfa` in the current folder first and then in the `PS` folder(s).

`OzTeX` can also download font information stored in a Macintosh-specific PostScript font file. It checks the type of the file specified after "<" and if the type is `LWFN` then it will download the PostScript code stored in the file's `POST` resources. Such files are normally kept somewhere in your System folder (e.g., in the `Fonts` sub-folder under `System 7.1`), so you will need to add a suitable path to the `PS` folder(s) in the `Default` config file.

To print a DVI file that uses PostScript fonts on a non-PostScript printer you should have the corresponding screen fonts installed in your system. For example, if `putr0` appears in the

DVI file then `OzTeX` will try to use the screen font `Utopia`. If `Utopia` is not installed then `OzTeX` will display an error message and use the system font instead (normally `Chicago`). When previewing a DVI file, `OzTeX` will also warn you if it can't find the screen font for a PostScript font; all characters from such a font will be shown using the system font in the missing font colour.

There are some important differences between PostScript fonts and the Computer Modern fonts created specifically for use with `TeX`:

1. A PostScript font can be requested at almost any size. For example, in Plain `TeX` you can do things like “`\font\helv=phvr at 33.3pt`”.
2. When a PostScript font is scaled to the same design size as a CM font it tends to look darker and larger. It is probably not a good idea to use the two font designs in the same document.
3. The standard `TeX` commands for accents and foreign letters need to be redefined for a PostScript text font. For Plain `TeX` users, the file `pstext.tex` in the `Plain` sub-folder in `TeX-inputs` contains the required macro definitions. Another file, `psfonts.tex` in `TeX-docs`, inputs these macros and illustrates the use (and abuse) 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 in the `PSNFSS` sub-folder in `TeX-inputs` 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. Another alternative is to get the `BaKoMa` or `CM/PS` fonts; these are PostScript versions of the Computer Modern fonts. `OzTeX` provides configuration files for switching to `BaKoMa`, `CM/PS`, `Lucida` or `MathTime`.

7.5 Macintosh screen fonts

The fonts listed at the end of a configuration file are not restricted to PostScript fonts. The list is really for any non-PK font, and that could be an outline font in TrueType format, or even some sort of bitmap font in non-PK format. The critical requirement is that a matching TFM file exists so that `TeX` can typeset characters from that font.

When previewing a DVI file containing a non-PK font (or when printing such a file on a non-PostScript printer), `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 system font instead. If you want to use PostScript fonts with `OzTeX` then for best results you should get the TrueType or ATM versions of the screen fonts listed in the `Default` config file.

7.6 Encoding files

Encoding files give you complete control over which Macintosh character is displayed when a particular PostScript character in a DVI file is to be viewed or printed on a non-PostScript printer (encoding files are *not* used during either low-level or standard PostScript printing).

Encoding files are loaded at the time a configuration file is loaded. Every PostScript screen font 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 PS folder(s) and, if found, uses it to build an encoding array of 256 elements. This array defines the mapping of DVI character codes into Macintosh characters. You can see the currently loaded encoding arrays by selecting the appropriate sub-item in the Config menu's "Show" item.

The syntax of an encoding file is described in detail in the standard encoding file called `Mac.enc` which is kept in the `Encodings` sub-folder in `PS-files`. This encoding file maps characters from a standard PostScript text font into equivalent characters in a standard Macintosh text font. Depending on the encoding scheme used in your PostScript TFMs or screen fonts, you might need to create other encoding files. For an interesting example of how such files can be used, see the `View CM using PS` configuration file.

7.7 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 article in TUGboat vol. 11 no. 1. A text version called `knuth-vf` can be found in the `TeX-docs` folder.

OzTeX supports virtual fonts via two items in the TeX menu: "Fix VFs in DVI" can be used to select a DVI file and replace all virtual font characters with appropriate characters from actual fonts (or with the specified rules and other DVI commands allowed in VF files). The code used to perform this task is based on Peter Breitenlohner's `DVIcopy` program.

Another item, "Fix VFs after TeX", can be ticked to tell OzTeX to automatically fix any VFs in the DVI file created at the end of a TeX run. This means you don't have to remember to run "Fix VFs in DVI" before viewing or printing a DVI file that contains virtual fonts.

If you intend to make frequent use of virtual fonts then it would be a good idea to tick "Fix VFs after TeX". Even if you only use virtual fonts rarely, it doesn't hurt to tick this option. OzTeX can quickly check to see if any virtual fonts are used in a DVI file; if none are used then no further processing is carried out. To decide if font `foo` is virtual or not, OzTeX looks for a file called `foo.vf` in the `VF-files` folder.

If one or more virtual fonts are used then OzTeX reads the entire DVI file, creates a new DVI file with no virtual font references, deletes the original DVI file and renames the new file with the original name. If an error is detected, or if you decide to cancel processing, OzTeX will delete the new file and leave the original file untouched.

8 OzTools

OzTools is a companion program for Oz \TeX which includes the following \TeX -related tools:

TFtoPL — Converts a TFM (\TeX font metric) file into a PL (property list) file. The PL file will be created in the current folder; it is a human-readable version of the information stored in the TFM file. Apart from verifying the correctness of a TFM file, **TFtoPL** can also be used to modify the font metrics. For example, you can edit the PL file, change the amount of kerning between two characters, and run **PLtoTF**.

PLtoTF — Converts a PL file into a TFM file. The TFM file will be created in the current folder.

VFtoVP — Converts a VF (virtual font) file into a VPL (virtual property list) file. The VPL file will be created in the current folder; it is a human-readable version of the information stored in the VF file. **VFtoVP** also needs to read one or more TFM files from the folder path specified in the “Set TFM folder” dialog. (If you keep the OzTools application in the same folder as the **TeX-fonts** folder then you won’t need to change the supplied path. Note that multiple search folders are not supported, so it’s a good idea to keep all VF-related TFMs in the one folder. All of Oz \TeX ’s VF-related TFMs are currently kept in the **PS** sub-folder in **TeX-fonts**.)

VPtoVF — Converts a VPL file into a VF file. The VF file will be created in the current folder. **VPtoVF** also creates a new TFM file in the folder specified by the “Set TFM folder” dialog.

OzTools provides some nice features for running these tools in a Mac environment. For example, the standard file dialog has a “Do all files” button which allows you to process all TFM/PL/VF/VPL files in the current folder. Background processing is supported. You can also delete PL/VPL files from within OzTools.

9 Transferring TFM/VF/PK/DVI files

Oz \TeX reads standard TFM/VF/PK files, and reads and writes standard DVI files. If you have access to \TeX on some other computer system, you should be able to move such files to and from 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). When using Kermit to transfer binary files from another computer to your Mac, remember to type “**set file type binary**” before using the **send** command. The same goes for receiving such files from your Mac, but with one exception: to transfer a DVI file created by Oz \TeX to a VAX/VMS host you’ll need to type “**set file type fixed**” before using the **receive** command.

If you have access to the Internet then there is a huge library of TFM/VF/PK files available by anonymous ftp from many archive sites. When using ftp to transfer such files, remember to type the “**binary**” command before using “**get**”.

If you transfer a DVI file to your Mac and a mistake occurs during transmission, or if you forgot to send it as a binary file, then Oz \TeX will display some sort of error message soon after opening the file.

Note that a DVI file contains TFM file names; this can be a cause for concern if you plan to transfer DVI files from one \TeX 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 \TeX input file so you can edit any incompatible font names or **\special** commands.

10 Including graphics

Creating figures and illustrations with T_EX isn't easy. Although it is theoretically possible to place small dots anywhere on a page and build up an arbitrarily complex picture, time and memory limitations make such a scheme impractical. (L^AT_EX provides a `picture` environment, but it is very inefficient and only suitable for small, simple diagrams.)

T_EX does however provide a `\special` command which can be used to pass arbitrary information to the DVI interpreter. OzT_EX's printing and previewing code interprets a `\special` command using a simple syntax that allows inclusion of a given file.

Note that using `\special` can seriously reduce the portability of your T_EX documents because a given DVI-reading program can only understand `\special` commands that obey its own particular syntax. One useful technique to enhance portability is to hide the actual `\special` call inside a macro; then when you move your T_EX file to another computer all you need to do is change the macro definition (assuming the DVI driver supports similar functionality in its handling of `\special`). For an example of this approach see Larry Siebenmann's `boxedeps.doc` and `boxedeps.tex` in the `Plain` sub-folder in `TeX-inputs`. Another alternative is `epsf.tex`; see `epsftest.tex` in the `TeX-docs` folder. Using either of these macro packages will make your input files much more portable.

A `\special` command can appear almost anywhere in your input file. It behaves like an invisible box of zero height and width. T_EX simply stores the given information in the DVI file at the current page position. When previewing a DVI file, OzT_EX draws a small marker indicating the location of a `\special`. The "Page Info" item will display this location in paper coordinates as well as the text of the `\special` command. Similar information is displayed if you choose to show statistics when printing a DVI file.

10.1 Including Macintosh graphic files

OzT_EX allows the inclusion of three common types of Macintosh graphic files:

- PICT files can be generated by almost all drawing/painting programs on the Macintosh. A quick way to get a PICT file if you're using System 7 is to type Shift-Command-3. To include a PICT file, use a command of the form:

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

Note that OzT_EX converts the PICT file into a black and white bitmap for previewing and printing, so you won't see any colours. The bitmap is always at 72 dpi when previewing, but the current printer resolution is used when printing.

- PNTG files are also very common — they're better known as MacPaint files. (Shift-Command-3 on a Mac running System 6 or earlier will create a screen dump in a file of type PNTG. If you have a colour monitor you'll first need to set it to black and white.) To include a PNTG file, use a command of the form:

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

- EPSF files are becoming increasingly common. They contain encapsulated PostScript code in the data fork which OzT_EX uses when PostScript printing and a corresponding PICT image in the resource fork which OzT_EX uses when previewing or non-PostScript printing. To include an EPSF file, use a command of the form:

```
\special{epsf=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, `OzTeX` uses the same search strategy as that for a `TeX` input file. It looks in the current folder first, and then in the `TeX` input folder(s). The file name can also be a 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 sub-folder called `my-pict-files` in the current folder. Spaces and other special characters (like `=`) can be included in a file name or path name by enclosing them in double quotes. For example,

```
\special{pict="my pict file"}
```

will include a file called `my pict file`. Using spaces and other unusual characters in a file name is not recommended, particularly if `TeX`'s `\input` command might be used to read the file (as in the `boxedeps.tex` and `epsf.tex` macros). Also avoid using double quotes in a file name.

The `PICT/PNTG/EP SF \special` commands all allow you to preview graphics in your DVI file, but if you have a choice, use `EP SF` files because:

1. The output on a PostScript printer should be of much higher quality than a similar `PICT/PNTG` image.
2. A coloured `EP SF` picture will print correctly on a PostScript printer (assuming it supports colour) even though `OzTeX` will convert its `PICT` resource into a black and white bitmap when previewing.
3. The `boxedeps.tex` and `epsf.tex` macros can extract the bounding box information in an `EP SF` file and position the image automatically.
4. Encapsulated PostScript is supported on many other computers.

When creating a `PICT/PNTG` file it doesn't really matter where you position the picture; `OzTeX` will convert it to a black and white bitmap, find the smallest bounding box of all black pixels in the bitmap, then place the bounding box so that its bottom left corner coincides with the location of the `\special`.

`OzTeX` does much the same thing for an `EP SF` picture; it places the lower left corner of the bounding box at the location of the `\special`. The only difference is that if the `PICT 256` resource is being used (during preview or non-PostScript printing) then `OzTeX` makes no attempt to determine the smallest bounding box of all black pixels in the picture. This is because the picture's dimensions should match those of the `BoundingBox` comment in the data fork (`OzTeX` will warn you if the dimensions differ by more than 3 bp).

`OzTeX` allows optional `keyword=value` pairs after the file name in a `PICT/PNTG/EP SF \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 ≤ 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 ≤ 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.