OzTeX User Guide

Version 3.1, September 1997

Andrew Trevorrow
`akt@kagi.com`
`http://www.kagi.com/authors/akt/`

OzTeX is a Macintosh implementation of Donald Knuth's TeX typesetting system. This document assumes you know how to use a Macintosh. It also assumes you know how to create a TeX (or LaTeX) input file. If you have access to a PostScript printer then an understanding of PostScript would be useful, but it is not essential.

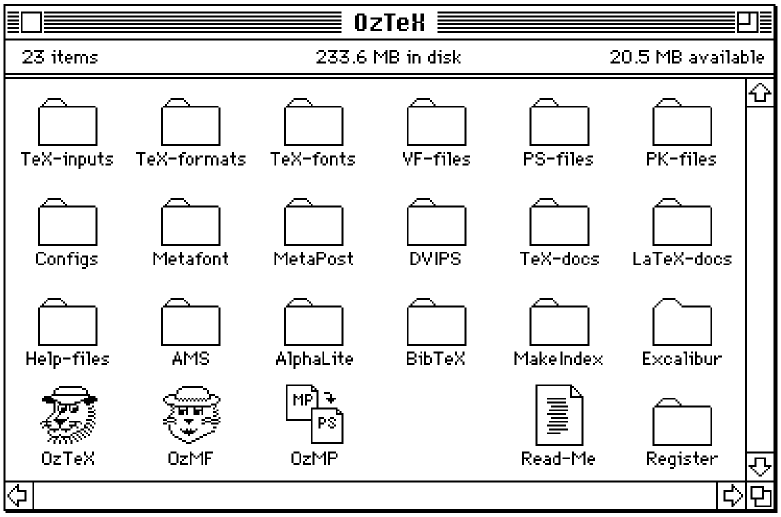
OzTeX aims to provide a standard TeX environment for the Macintosh that is easy to extend or customize to suit your particular needs. If you have used TeX on some other computer then the way OzTeX works shouldn't be too surprising.

OzTeX is shareware, so you are welcome to try it out before paying the registration fee (see the "Shareware Fee" item in OzTeX's Help menu for details about how to pay). Comments, bug reports and suggestions are all welcome, but please note that I only reply to e-mail queries from registered individuals.

Contents

1 Introduction

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



Here's a brief description of what's inside all those folders:

- The `TeX-inputs` folder contains a large number of TeX input files stored in various subfolders. TeX looks in these subfolders if it can't find a given file in your current folder. The supplied subfolders contain the files needed to use Plain TeX, LaTeX and REVTeX. If you have your own library of macro files then the best place for them is in a new subfolder in `TeX-inputs`.

- The `TeX-formats` folder contains the format files read by TeX and the `TeX.pool` file read by `INITEX`. The three most popular formats are supplied: `Plain.fmt`, `LaTeX.fmt` and `AMSTeX.fmt`. Section 6 describes how to create format files.

- The `TeX-fonts` folder contains many TeX font metric files stored in various subfolders. 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` subfolder; see section 7.6.

- The `PK-files` folder contains PK font files stored in various subfolders. The name of each subfolder 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. OzTeX is distributed with a fairly small set of PK files (sufficient to print out the documentation on most printers). New PK files at any desired size can be built on demand; see section 8.

- The `Configs` folder contains OzTeX's configuration files. The most important of these is the `Default` file loaded by OzTeX when it starts up. See section 4 for a detailed discussion of configuration files, including the required syntax.

- The `Metafont` folder contains files needed by OzMF, a Mac implementation of Donald Knuth's METAFONT program and other font related tools. The `Default` config file loaded by OzMF is in the `Configs` subfolder in `Metafont`. See section 8 for more information about OzMF.

- The MetaPost folder contains files needed by OzMP, a Mac implementation of John Hobby's MetaPost program. The Default config file loaded by OzMP is in the Configs subfolder in MetaPost. See section 9 for more information about OzMP.
- The DVIPS folder contains files needed by dvips, Tom Rokicki's popular DVI-to-PostScript translator. The OzTEX application contains an enhanced version of dvips. See section 10 for more information.
- The TeX-docs folder contains examples of Plain TEX files, including nasty.tex which is used in the guided tour of OzTEX. It also contains gentle.tex, the source file for Michael Doob's *A Gentle Introduction to TEX* and a precursor to his book *TEX Starting from 1* [3]. TEX beginners should find this a useful companion to *The TEXbook* by Donald Knuth [9].
- The LaTeX-docs folder contains ozuser.tex, the source file for this user guide, plus hyper.tex which shows how to use the hyperref and color packages. Also provided in LaTeX-docs is the LaTeX Short Course subfolder generously donated by George Grätzer; it contains the source files for Part I of his book *Math into LATEX — An Introduction to LATEX and AMS-LATEX* [7].
- The Help-files folder contains a number of text files used by OzTEX. The information in these files can be displayed by selecting their names from OzTEX's Help menu. (Note that the help files for OzMF and OzMP are stored in the Help subfolders in Metafont and MetaPost respectively.)
- The AMS folder contains all the input files for AMS-TEX and AMS-LATEX. It also contains TFM files and METAFONT source files for the AMS fonts.
- The AlphaLite folder contains Pete Keleher's AlphaLite application for editing TEX/LATEX input files. Note that AlphaLite is shareware and comes with its own Register application. If you prefer to use a different editor then throw away the AlphaLite folder.
- The BibTeX folder contains Vince Darley's port of BibTEX which can be used in conjunction with LATEX to generate a bibliography from information stored in a bibliographic database.
- The MakeIndex folder contains Rick Zaccone's port of MakeIndex for creating a sorted index (like the one at the end of this user guide).
- The Excalibur folder contains Excalibur, a spelling checker written by Rick Zaccone and Robert Gottshall. Excalibur was designed to handle LATEX documents, but also works quite well with Plain TEX files.
- The Register folder contains the Register application for paying OzTEX's shareware fee. Some brief notes on how to use Register are also included.

The OzTEX folder can be called anything you like, and can be moved anywhere on your disk. However, if you decide to rename some of the above folders or applications then you'll probably need to make corresponding changes to the various Default configuration files loaded by OzTEX/OzMF/OzMP. The paths to these Default files are stored as resource strings in the preferences file for each application, so you'll also need to use ResEdit if you want to change the names of the Configs, Metafont or MetaPost folders. If you are new to OzTEX then my advice is to stick to the given folder names!

2 A guided tour of OzTEX

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

2.1 Starting OzTEX

Assuming you've installed everything, double-click on the OzTEX application file to get the program started. After a brief pause you should see a window with the title "OzTEX" and a couple of messages showing the current version of OzTEX and the path name of the `Default` configuration file just loaded. The `Default` file contains references to an encoding file called `Mac8r.enc`, so it will also be loaded.

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

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. Text in the window can be copied to the Clipboard or saved to a file.

You can also start up OzTEX by double-clicking on `foo.dvi`. The advantage of this method is that if `foo.tex` exists then that file name will appear in the TEX menu, ready for typesetting. Note that under System 7 or later you can drop `foo.tex` 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.

2.2 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 open-file dialog to locate and select the application you'll be using to edit your TEX/LATEX 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 3.2 for more details about the items in the Edit menu.

2.3 Editing a TEX input file

A TEX input file is a standard Macintosh text file. We won't create an input file from scratch, instead we'll use one of the sample files provided in the `TeX-docs` folder. There is nothing special about this folder, or `LaTeX-docs`. I simply like to keep my TEX and LATEX 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.

2.4 Running TeX

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

Choose "TeX..." from the TeX menu and open `nasty.tex`. 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.
l.5 \xxx

? █
```

Chapter 6 in *The TeXbook* [9] explains what you can do in such a situation. (Section 3.2 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 open-file dialog. The second item in the TeX menu has also been updated with the name of the most recent input file (`nasty.tex` in this case), so you can easily typeset the same file again by pressing Command-T. The file name also appears in the bottom item of the Edit menu so you can quickly switch back to your editor by pressing Command-E.

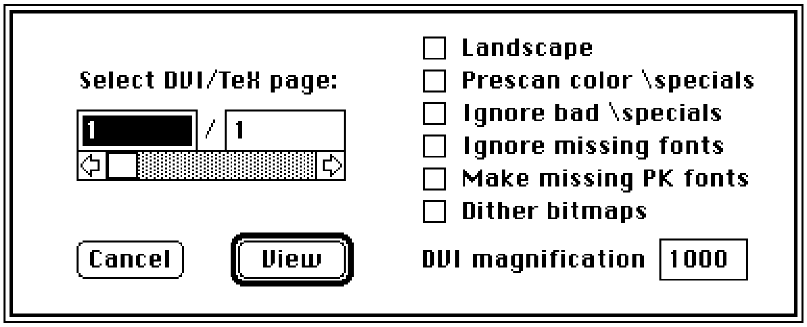
Another way to repeat a TeX run is to type "`t`" at any time during the first run. This is very handy for LaTeX users because two runs are often needed to resolve all cross references. Type "`tt`" to get two extra runs (any more than two "`t`"s will be ignored). Type "`-t`" to cancel the extra run(s).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

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

At the end of every TeX run, OzTeX displays the time taken and the amount of memory allocated by TeX for its large arrays. The total number of bytes depends on the current TeX parameters. The values in the `Default` config file are suitable for small to medium-size documents. For some documents you might need to increase one or more TeX parameters; you'll know this is necessary if TeX quits with a "capacity exceeded" error; see section 12.3. For very large documents, like a book or thesis, you will probably need to increase most of the parameters and create big formats; see section 6.1.

2.5 Viewing a DVI file

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



Select DUI/TeH page:

1 / 1

□ Landscape
□ Prescan color \specials
□ Ignore bad \specials
□ Ignore missing fonts
□ Make missing PK fonts
□ Dither bitmaps

DUI magnification 1000

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

2.5.1 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".

2.5.2 Paper coordinates

An understanding of the coordinate system used by OzTeX to view (and print) a DVI page would be useful. OzTeX's internal units are called "paper pixels". The resolution parameters in a configuration file define the number of paper pixels per inch; the `viewing_resolution` is only used when viewing a DVI file and the `printing_resolution` is only used when printing a DVI file on a non-PostScript printer (for a PostScript printer, `dvips` uses the resolution specified in `config.ps`).

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 (set the units to `px` to see paper pixel values).

2.5.3 Zooming in and out

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

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

When you zoom in or out by clicking, 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.

2.5.4 Scrolling around

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

2.5.5 Checking for errors

It is a good idea to select the “Page Info” item at least once while viewing a DVI file. This item displays its results in the OzTEX window. The display includes a list of all the fonts used in the DVI file and clearly indicates any that are missing. (If virtual fonts are used then “Page Info” won't necessarily detect all missing PK fonts. A missing PK font referenced in a virtual font will only be detected if a character requiring that PK font is used somewhere on the current page.) OzTEX also lists any `\special` commands on the current page, showing their locations and arguments.

2.5.6 Selecting pages

There are a number of interesting pages in `nasty.dvi`. Use “Previous Page”, “Next Page” or “Go to Page...” to have a look at some of them. Note that the current DVI/TEX page numbers are 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-dot if you get bored.

- Page `3` has examples of `\special` commands. OzTEX is able to preview included files of type PICT, PNTG or EPSF; see section 11 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 closed automatically whenever you run TEX, print a DVI file or load a configuration file.

2.5.7 Keyboard short cuts

OzTEX provides a number of keyboard short cuts to make it easier to preview a DVI file:

- Type “`o`” to bypass the view dialog and open the most recently viewed DVI page (or page 1 if you haven't viewed anything yet). OzTEX will also restore the scale factor and location you last used.

- Type “`i`” to see page info.

- Type “`g`” to go to a given page and switch to the default view.

- Type “`<num>g`” to go to DVI page `<num>`. “`-1g`” will go to the last page, “`-2g`” to the 2nd last page, etc. “`99g`” will also go to the last page if there are fewer than 99 pages.

- Type “`<num>t`” to go to TEX page `<num>`. “`99t`” will only go to TEX page 99 if it exists; if not, OzTEX beeps and the page selection dialog appears.

- Type "`b`" to move to the bottom of the previous page without changing the current scale or horizontal position.
- Type "`n`" to move to the top of the next page without changing the current scale or horizontal position.
- Type "`[`" to go back one link.
- Type "`]`" to go forward one link.
- Type "`f`" to change the scale to full view.
- Type "`a`" to change the scale to actual size.
- Type "`h`" (or Home) to move to the first page and switch to the default view.
- Type "`e`" (or End) to move to the last page and switch to the default view.
- Hit the space bar (or Page Down) to scroll forwards through a document.
- Hit the Delete key (or Page Up) to scroll backwards.
- Hit the Return key to bring either the view window or the OzTEX window to the front.
- Type "`d`" to toggle the dither option (System 7 or later is required).
- Type "`w`" to close the view window.

Note that all the Command-key items in the View menu have matching short cuts.

2.6 Printing a DVI file

Before printing a DVI file it is a good idea to select "Page Setup" from the File menu and make sure the paper size and orientation are correct. To print `nasty.dvi` choose either of the top two items in the File menu. Choosing the second item simply avoids the open-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.

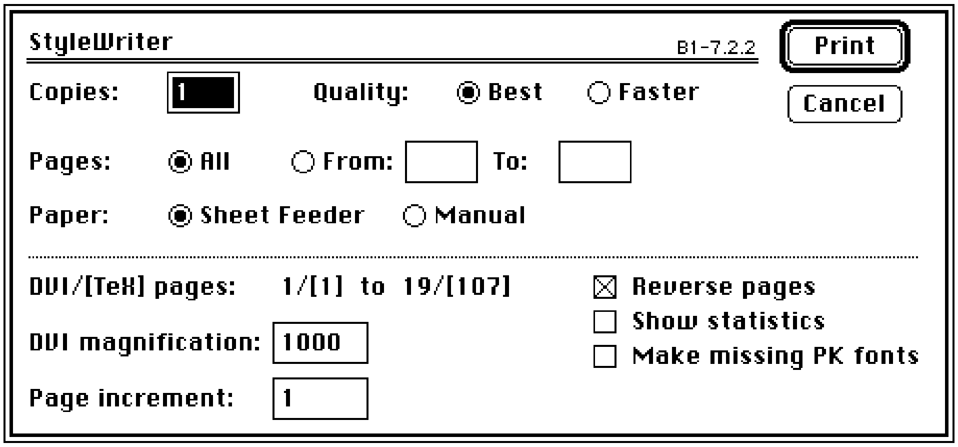
OzTEX supports three methods of printing a DVI file:

1. Non-PostScript printing.
2. Low-level PostScript printing.
3. Standard PostScript printing.

The printing method chosen by OzTEX depends on whether or not the current printer is a PostScript device, and the current state of the File menu's "Use Standard PostScript" flag. For the bottom two methods, OzTEX uses a built-in version of Tom Rokicki's `dvips` program to convert a DVI file into PostScript; see section 10 for details about `dvips`.

2.6.1 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 current `printing_resolution` value does not match the actual resolution of the printer then you'll first be warned about possible scaling problems; see section 12.5.) For example, if the current printer is a StyleWriter then you'll get a dialog like this:



StyleWriter B1-7.2.2

Copies: 1 Quality: ● Best ○ Faster

Pages: ● All ○ From: □ To: □

Paper: ● Sheet Feeder ○ Manual

DVI/[TeX] pages: 1/[1] to 19/[107]

DVI magnification: 1000

Page increment: 1

☒ Reverse pages
□ Show statistics
□ Make missing PK fonts

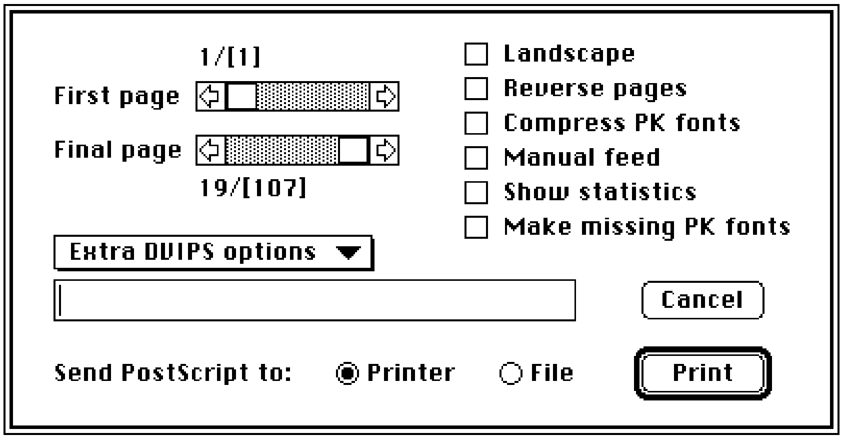
The dialog box lets you change various options specific to the current printer. OzTEX adds some extra items at the bottom of the dialog:

- 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.

2.6.2 Low-level PostScript printing

If the current printer is a PostScript device and "Use Standard PostScript" is *not* ticked then OzTEX will start up `dvips` and display the following dialog:



First page 1/[1]

Final page 19/[107]

□ Landscape
□ Reverse pages
□ Compress PK fonts
□ Manual feed
□ Show statistics
□ Make missing PK fonts

Extra DVIPS options ▼

Send PostScript to: ● Printer ○ File

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

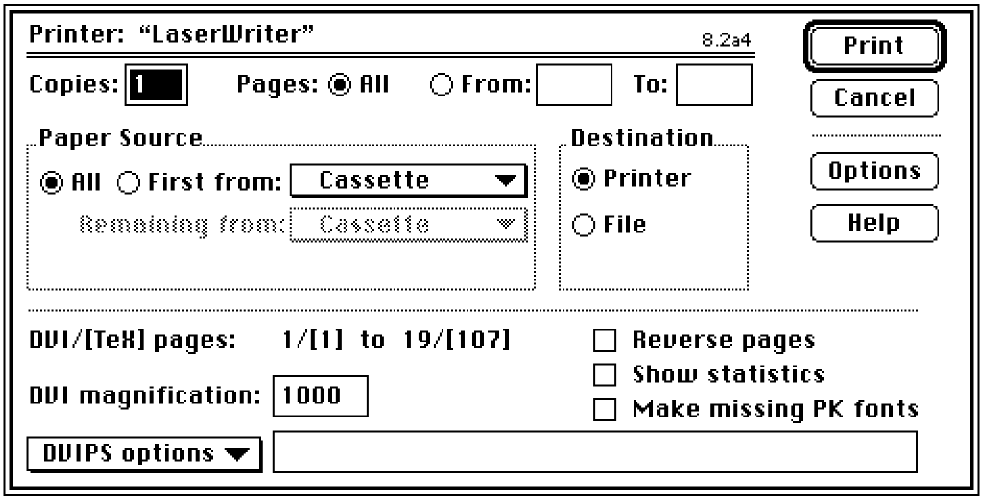
A number of check boxes let you change the paper orientation (landscape or portrait), the page order (reverse or normal), whether or not PK fonts should be compressed (useful at high resolutions or when very large fonts are used), the paper source (manual feed or normal input tray), whether or not to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

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

Two radio buttons let you send the output to the printer or to a file. We'll use the printer, so click on the Print button (or hit Return). 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.

2.6.3 Standard PostScript printing

If the current printer is a PostScript device and "Use Standard PostScript" *is* ticked then OzTeX will start up `dvips` and display the standard dialog for that printer:



Printer: "LaserWriter" 8.2a4

Copies: 1 Pages: ● All ○ From: To:

Paper Source: ● All ○ First from: Cassette

Remaining from: Cassette

Destination: ● Printer ○ File

DVI/[TeX] pages: 1/[1] to 19/[107]

DVI magnification: 1000

□ Reverse pages

□ Show statistics

□ Make missing PK fonts

Below the normal options for the current printer, OzTeX adds some extra items. They are the same ones described in the dialog box for non-PostScript printing in section 2.6.1, except for the page increment which has been replaced by a pop-up menu for entering `dvips` options. The items in this pop-up menu are slightly different to those in the low-level dialog because some options either cannot be used or don't make sense when merging `dvips` output with the PostScript code generated by Apple's LaserWriter drivers.

NOTE: Due to incompatibility problems, the pop-up menu is not available in the print dialog for LaserWriter 8.4.x or later.

Click on the Print button or hit the Return key to start printing. Note that dvips has to go through all selected DVI pages first and determine font and character usage. All the PK character bitmaps and PostScript font files are then downloaded at the beginning, so the first page in a long document might take a while to be printed.

Although there are some advantages to using standard PostScript, such as being able to select printer-specific options like which input tray to use, there are significant disadvantages. A limitation in LaserWriter drivers older than version 8 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, the standard PostScript code takes longer to print.

If you want to use OzTEX's more efficient low-level printing method then leave "Use Standard PostScript" unticked. OzTEX's low-level code works under QuickDraw GX on both AppleTalk and serial printers. Another advantage is that any PostScript error messages should appear in the OzTEX window (on some printers, like the LaserWriter Select 310, you might need to send :PS-files:geterrors.ps first).

2.7 Positioning pages correctly

For your DVI output to appear in precisely the right location on the paper, the TEX origin must be exactly one inch in from the top and left paper edges. Page 2 in nasty.dvi contains a thick rule with its top left corner exactly at the TEX origin.

If the output from a non-PostScript printer shows a significant discrepancy then adjust the horizontal_offset and/or vertical_offset in the Default config file to shift the origin to the correct position. These offset parameters are *not* used by dvips, so to adjust the origin for a PostScript printer you could add a line containing something like "O 3mm,-2mm" to the config.ps file in :DVIPS:Inputs:.

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. Note that the current paper dimensions *are* passed on to dvips.

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.

2.8 Continuing from where you left off

That's the end of the guided tour. One more useful feature to note is that OzTEX remembers most of your settings in a preferences file called OzTeX Prefs. Some of the important settings include window sizes and locations, the 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 subfolder under System 7 or later, or in the System folder itself under 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. Note that OzMF and OzMP have their own Prefs files.

3 OzTEX's menus

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

3.1 The File menu

Print DVI...

Prints a selected DVI file. After selecting the DVI file you'll be presented with a dialog box that depends on the current printer and the setting of "Use Standard PostScript" (described below). Non-PostScript printing is discussed in section 2.6.1. Low-level PostScript printing is discussed in section 2.6.2. Standard PostScript printing is discussed in section 2.6.3. Hit Command-dot at any time to cancel printing, or switch to another application and OzTEX will continue to print in the background.

Print ?.dvi

Prints the indicated DVI file. 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. 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 OzTEX's `paper_width` and `paper_height` 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.

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.

Send ?.ps

Sends the indicated PostScript file to the current printer. This item is initially disabled; the file name changes after a PostScript file is created by `dvips` or one of the PS tools.

Save All Text . . .

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

Delete

Lets you delete files ending in a given string selected from a sub-menu. The open-file dialog has a "Do all files" button that lets you delete all such files in the selected folder. Use this feature with caution! (A note to ResEdit hackers: you can modify the Delete MENU resource to create your own menu of suffixes.)

Quit

Quits OzTEX.

3.2 The Edit menu

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

Copy All Text

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

Font

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

Size

Selects the point size of the font used in the OzTEX window. (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 in front.

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 (see 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 AlphaLite supports the extra parameter.

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).

3.3 The TEX menu

TEX . . .

Typesets a selected input file using the current format (or the format on the first line of the input file; see below). 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* [9] will apply to the version of TEX run by OzTEX, and the same goes for Leslie Lamport's LATEX book [12] if you use that format. A few extra features have been added in OzTEX:

- TEX looks for "`%&format`" on the first line of a selected input file and, if present, uses the given format regardless of the currently ticked format in the TEX menu.
- Input files can have either Return or Linefeed as the end-of-line character. This means that TEX can read Unix text files unmodified and thus avoid spurious `buf_size` overflow errors.
- Most of TEX's capacity parameters are set at run-time in a configuration file rather than at compile-time. If you get a "capacity exceeded" error then you should be able to overcome the problem by increasing the appropriate parameter (see section 12.3).
- TEX will look for input files in the current folder first, then in the folders specified by the `input_folders` parameter. Similarly, when looking for TFM files, TEX will look in the current folder first and then in the folders specified by the `tfm_folders` parameter.
- New text files created by OzTEX will have your chosen editor's signature, so double-clicking on a new `.log/.aux` file will open it in the editor. However, OzTEX won't change the signature of an existing file, so if you want to double-click on a `.log` file to start up OzTEX, use ResEdit or some other utility to change the file's creator signature to "OTEX".

- You can interrupt a TEX session at any time by typing Command-dot. Depending on what it is currently doing, TEX usually responds immediately with a suitable message and the "?" prompt. If you hit 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 on the first line of the input file). OzTEX remembers the most recently typeset file.

INITEX

Runs `INITEX`, a special version of TEX normally used to create format files; see section 6.

List of formats

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

The current format is indicated by a tick and will be the one used when you typeset a file, although this will be overridden if the first line of the file starts with "`%&format`". To change the current format, simply select a different format item. OzTEX automatically ticks the first format found when loading a config file containing `format_files`, so it's a good idea to place your preferred format first.

3.4 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 5) that lets you choose various viewing options:

- The initial page to display. A scroll bar allows you to locate any page in the DVI file. The left and right arrow keys can be used to move the thumb box. You can also type in a specific DVI page or 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.
- Prescanning for color `\specials`. You only need to enable this option if your document uses color; see section 11.5.
- The next two 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 previewer recognizes all `dvips`-specific `\special` commands and silently ignores those it can't handle (like rotation and other commands that use raw PostScript code). Ditto for `tpic` `\specials`. This means that most people should never need to tick the "Ignore bad `\specials`" option.

- The second last check box can be used to make missing PK fonts (but it will be ignored if "Ignore missing fonts" is selected). OzTEX remembers the check box setting. See section 8 for more details. This option is only available under System 7 or later.

- The last check box tells OzTEX whether or not to dither the bitmaps used for PK fonts and `\special` images. If you have a color or gray-scale monitor then dithering should improve the readability of PK fonts. OzTEX remembers the check box setting. Dithering is only available under System 7 or later.

- The DVI magnification (expressed as an integer 1000 times the desired magnification). You should only alter the default value if you plan to print the DVI file at a different magnification. See Chapters 4 and 10 in *The TEXbook* [9] for details. A document using only PostScript fonts can be viewed/printed at almost any magnification (OzTEX allows values from 1 to 10000). 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. The initial scale factor and page location will be set to values determined by the default view (see section 2.5.1). 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. 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 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`".

- A list of all virtual fonts used in the DVI file. Those used on the current page are flagged by a comment.
- The number of rules on the current page.
- The location and argument of each `\special` command on the page. The location is expressed in terms of OzTEX's paper coordinate system; see section 2.5.2. If the `\special` command includes a valid PICT/PNTG/EPSF file then the width and height of the image's bounding box is also displayed.

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

Previous Page

Displays the previous DVI page. This item is disabled if the view window is 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.

Back Link

Takes you back one HyperTEX link; see section 11.6.

Forward Link

Takes you forward one HyperTEX link; see section 11.6.

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 2.5.1). This item is disabled if the view window is closed.

3.5 The Tools menu

NOTE: If the Option key is held down while selecting an item in the Tools menu then OzTEX will skip the open-file dialog and use the current DVI/PS file.

DVIPS...

This item is provided so you can always create a `.ps` file even if the currently chosen printer is non-PostScript. When `dvips` is called from the File menu then the default destination is the current printer, but if called from the Tools menu then the default destination is a file.

Note that Option-clicking on the Save button will skip the save-file dialog and create `foo.ps` in the same folder as `foo.dvi`.

DVIDVI...

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

After a DVI file has been selected, DVIDVI will look in the `Help-files` folder for a file named "`DVIDVI examples`". If found, its contents will be displayed just before the `Options` prompt so that you can cut and paste the examples.

DVIcopy...

Selects a DVI file and replaces all virtual fonts with actual fonts. See section 7.7 for information about virtual fonts. Most people won't need to use DVIcopy because OzTEX automatically resolves all references to virtual fonts. If you decide to use DVIcopy then you need to be aware of a potential problem; see the DVIcopy item in OzTEX's Help menu for details.

PSbook . . .

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

PSnup . . .

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

PSselect . . .

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

PStoPS . . .

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

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

After a PostScript file has been selected, each tool will look in the `Help-files` folder for a file named "`<tool> examples`". If found, the contents will be displayed in the OzTEX window just before the `Options` prompt so that you can cut and paste the examples. Feel free to remove/rename/edit the supplied files.

DVIDVI and the PS tools can be asked to create an output file with the same name as the input file (OzTEX temporarily renames the input file). They will also repeat the `Options` prompt if an error is detected in the given options, so there is no need to select the input and output files again.

3.6 The Config menu

Show

Shows the current values of configuration parameters in the OzTEX window. You can choose to show all parameters or only certain subsets. The output is displayed with the correct syntax for a configuration file; this makes it easy to 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. OzTEX will also load any config files whose names start with "`+`".

Other configuration files

The remaining items in the Config menu are determined by the `config_files` parameter in the `Default` 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:

`StyleWriter` configures OzTEX for the StyleWriter printer. The printing and viewing resolutions are set to 360 and both 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 printing and viewing resolutions are set to 144 and both modes to `imagewriter`.

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

`600 dpi printer` configures OzTEX for a 600 dpi printer. The printing resolution is set to 600 and the mode to `ljfour`. This mode is suitable for a HP LaserJet 4. If you have a different type of 600 dpi printer then you might need to use a different mode to get better looking fonts.

`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`. 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). 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. See the comments in the config file for more details.

`Add CM/PS Fonts` updates the current list of PostScript fonts so that OzTEX can use the CM/PS fonts created by Blue Sky Research. The CM/PS fonts are PostScript versions of the Computer Modern fonts. These are also in the public domain and available from CTAN sites. However, you need to install ATM to get nice-looking screen fonts.

`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.

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.

`LaTeX209.config` and `LaTeX.config` can be used to configure OzTEX to typeset documents with either the old LATEX 2.09 format or the new LATEX format; see section 5.2.

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.

3.7 The Help menu

The items appearing in this menu are set by the `help_files` parameter 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/LATEX 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 add it to the `help_files` list in the `Default` file.

4 Configuration files

OzTEX, OzMF and OzMP can all be customized by changing parameter values in configuration files. The config files for OzTEX are kept in the `Configs` folder. The config files for OzMF and OzMP are kept in the `Configs` subfolders inside the `Metafont` and `MetaPost` folders respectively.

4.1 The required syntax

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

- Blank lines are ignored. Lines starting with "`%`" are also ignored and can be used to include comments. This is also true for lines starting with "`@`", but in this case the rest of the line is displayed in the application's main window.

- All other lines are parameters of the form "keyword = value". Keywords are case insensitive. Spaces or tabs around a keyword are ignored. The same keyword can appear more than once in a config file; later values override earlier ones.

- The "`=`" character is optional, but I recommend you put it in. Any spaces or tabs around "`=`" are ignored. The parameter value consists of one or more strings. Multiple strings are separated by spaces or tabs, so use double quotes around a string that includes spaces. If only one string is required then the rest of the line is ignored.

- To avoid very long lines, use "`|`" to split lines anywhere you like. Upon reading a "`|`", the application will ignore the rest of the current line and continue reading from the next line. The best place to use "`|`" is where a space is allowed; that way you can safely indent the next line to make the file more readable.

- A line starting with "`!`" tells the application to stop reading the config file. This lets you add comments without having to start each line with "`%`".

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

4.2 Path names

Some parameters are used to tell each application where to find certain files and folders. The parameter values consist of one or more strings in the form of 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. Path names must be enclosed in double quotes if they contain any spaces; e.g., `":My PK files:"`.

NOTE: Because OzTEX's path names are also used by `dvips`, you cannot use a comma in those paths; if you do you'll get an error message when the config file is loaded. This is because commas are used to separate `dvips` search paths (the standard version of `dvips` uses colons, but they can't be used on the Mac).

Partial path names begin with a colon. The first folder in such a path is assumed to be in the same location as the 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:VF-files:`.

Each application will generate a warning message if a given path name does not specify a known folder or file. Note that under System 7 or later, aliased files are supported, but not aliased folders.

4.3 Default configuration files

The Oz... applications all read a config file called `Default` when starting up. The supplied `Default` files contain examples of all possible keywords. They also contain detailed notes (at the end of each file) discussing how the important parameters are used and other handy tips. Please read these notes before making changes to a `Default` file.

After loading a `Default` file, each application will also load any files in the `config_files` list whose names start with "`+`". So, rather than modify the `Default` file, you could create a new config file called "`+Changes`" which contains all your changes to the default settings. (You still need to edit the `Default` file to add "`+Changes`" to the `config_files` list!) This might make life easier when upgrading to future versions of OzTEX.

Another good idea when modifying a `Default` config file is to isolate your changes in the one spot. Because later use of a keyword overrides an earlier use, the best location for all your changes is just before the "`!`" character at the start of the notes.

4.4 Other configuration files

The Oz... applications all have a Config menu that lets you switch rapidly from one configuration file to another. In OzTEX's case, this is especially useful if you have access to more than one printer because each printer will probably require a separate config file. There are many other uses for config files; see section 3.6 for a brief description of the various OzTEX config files. It is a simple matter to create your own files and add their names to the `config_files` list in the `Default` file.

5 LaTeX

OzTeX is distributed with the latest implementation of LaTeX (formerly called LaTeX 2e). This new version of LaTeX is described in the 2nd edition of *LaTeX: A Document Preparation System* by Leslie Lamport [12] and *The LaTeX Companion* by Michel Goossens, Frank Mittelbach and Alexander Samarin [5].

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

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

Subfolder in `TeX-inputs`	Corresponding CTAN directory
`LaTeX`	`/tex-archive/macros/latex/unpacked`
`Graphics`	`/tex-archive/macros/latex/packages/graphics`
`PSNFSS`	`/tex-archive/macros/latex/packages/psnfss`
`Tools`	`/tex-archive/macros/latex/packages/tools`

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 OzTeX user!

5.1 Installing a new package

A LaTeX package usually consists of a `.ins` file, plus one or more `.dtx` files, and possibly some `.fdd` files. To install a package so that OzTeX can use it, run LaTeX on the `.ins` file. This should create one or more files with extensions of the form `.sty`, `.cfg` or `.fd`. Move these files into a folder where TeX can find them; a good idea is to put them in a new subfolder in `TeX-inputs`.

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

5.2 Using LaTeX 2.09

The old LaTeX 2.09 system is no longer distributed with OzTeX. LaTeX users should seriously consider switching to the new LaTeX; it can typeset LaTeX 2.09 documents in "compatibility" mode. If you have the old LaTeX system from an earlier version of OzTeX and you want to use it alongside the new LaTeX, then the best approach is to edit the `LaTeX209.config` and `LaTeX.config` files in the `Configs` folder and make any necessary changes to the `input_folders` list.

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

6 Building format files

A format file contains a pre-compiled set of macros which TEX can load in very quickly. To build a format file you need to run `INITEX` (in the TEX menu). Here are the steps needed to create `Plain.fmt`:

1. Run `INITEX`, wait for the "`**`" prompt, type "`Plain\dump`" and hit the Return key. `INITEX` will begin reading `plain.tex` from the `Plain` subfolder in `TeX-inputs`.
2. You'll eventually see the save-file dialog 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` subfolder 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.

Creating `AMSTeX.fmt` is just as easy: run `INITEX`, type "`AMSTeX.ini`" and save `AMSTeX.fmt` in the `TeX-formats` folder. If you decide to rename any of the existing formats, or add a completely new format, then remember to update the `format_files` list in your config file(s).

6.1 Building big formats

Some people with very large or complicated documents will need to build big formats. Here's how:

1. Edit the `Default` configuration file and locate the lines showing large TEX parameters; these are in the notes near the end of the file. Copy these lines to a spot anywhere after the supplied TEX parameters (a good spot would be just before the first "`!`").
2. Click on the OzTEX icon and use "Get Info" in the Finder's File menu to set OzTEX's preferred memory size to at least 3500K.
3. Start up OzTEX and follow the steps in the above section to rebuild all your formats. Remember to save the `.fmt` files in the `TeX-formats` folder and thus replace all existing files.

6.2 Building custom formats

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

7 Fonts

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

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 TEXbook* [9] if you want to know more about how TEX uses fonts, especially Chapter 4 and Appendix F.

OzTEX reads TFM files when printing or viewing a DVI file containing any of the non-PK fonts listed in your 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).

OzTEX will look for a TFM file in the current folder before looking in the subfolders in `TeX-fonts`. All the TFM files in the `CM` subfolder belong to the Computer Modern family of fonts (created by Donald Knuth using METAFONT). The TFM files in the `LaTeX` subfolder are used by LATEX. The TFM files in the `PS` subfolder 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 OzTEX distribution doesn't provide a TFM file for a font you would like to use then there are a number of things you can do:

1. If you have access to the Internet then look for the TFM file in the nearest CTAN site (see the "Related Software" item in the Help menu). If the font is called `foo` then look for `foo.tfm`. If it is a PostScript font then you might have to search a bit harder; using Karl Berry's naming 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). See section 8.2.

3. If `foo` is a PostScript font then look for `foo.afm`. An AFM (Adobe Font Metric) file contains very similar information to a TFM file. Use `AFM2TFM` in OzMF to convert `foo.afm` to `foo.tfm`; see the `dvips` manual for detailed information about `AFM2TFM`. Alan Jeffrey's `fontinst` package can also convert an AFM file into a PL or VPL file which OzMF can then convert into a TFM file using `PLtoTF` or `VPtoVF`.

4. As a last resort you could even create a PL file from scratch. A very tedious job.

Having found or created a new TFM file, you need to place it in a suitable folder so that OzTEX can find it. If you put it in one of the subfolders in `TeX-fonts` (or create a new subfolder) then you won't need to change the `tfm_folders` list 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. If you have a PostScript printer then `dvips` needs a similar entry added to `psfonts.map` in `:DVIPS:Inputs:`. The examples in those files 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 bitmap fonts; these are all the fonts *not* listed in your configuration file (or `psfonts.map` in the case of `dvips`). Each bitmap font usually has a number of PK files, each one representing the same font but at a different size. This size is calculated as follows:

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

where *resolution* is either the printing resolution or the viewing resolution, and *magnification* is the overall font magnification (i.e., the DVI magnification times the individual font scaling). The *size* is then rounded up to the nearest integer and should equal (± 1) one of the subfolder names in `PK-files` if you decide to keep PK files of the same size in separate folders.

Although TeX 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 TeX's `\magstep` values. See Chapters 4 and 10 of *The TeXbook* [9] for more details. For example, assuming a *resolution* of 300 and no document magnification:

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

TeX's `\magnification` command has a cumulative effect on font scaling:

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

The font size is now $300 \times 1.2 \times 1.44 = 518.4$, so OzTeX will use `cmr10.518pk`. Note that LaTeX has no `\magnification` command — use the `11pt` and `12pt` options to increase the size of all fonts in a LaTeX document.

OzTeX uses a very flexible search scheme when looking for fonts. If `foo` is a PostScript font then OzTeX looks for `foo.tfm` in the current folder before looking in `TeX-fonts`. If `foo` is not a PostScript font then OzTeX looks for a PK file using the list of templates specified by the `pk_files` parameter. For example, if the calculated size is 300 then OzTeX will look for:

1. `foo.300pk` (in the current folder)
2. `:PK-files:foo.300pk` (relative to the location of OzTeX)
3. `:PK-files:300:foo.300pk` (ditto)

In fact, at each step OzTeX will also add and subtract 1 from 300 because rounding errors can occur in the size calculation. If still not found, OzTeX either gives up or calls OzMF to make the missing PK file (see section 8.1).

There is still another possibility to consider: a font name can include an explicit location. For example, your TeX input file might contain `\font\xxx=:myfonts:foo`. In this case TeX will look for `:myfonts:foo.tfm` in the current folder; that is, it expects to find a file called `foo.tfm` in a subfolder 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 can build new PK files using OzMF as described in section 8.

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

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 in a configuration file on lines between `begin_fonts` and `end_fonts`; see the `Default` file. For `dvips` however, information about PostScript fonts is obtained from the `psfonts.map` file in `:DVIPS:Inputs:`.

The PostScript TFMs stored in the `PS` subfolder in `TeX-fonts` are meant to be used in conjunction with 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 OT1 TFM/VF files have names ending in `7t`. The T1 (or Cork) encoding is the new TeX text font encoding for 256-character fonts. The T1 TFM/VF files have names ending in `8t`. See `test0.tex` in the `PSNFSS` subfolder in `TeX-inputs` for how to select either encoding.

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

7.4.1 Reencoding PostScript fonts

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

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

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

This tells OzTeX to map character codes from `ptmr8r.tfm` to characters in the Times screen font using a mapping defined in `Mac8r.enc`. OzTeX looks for encoding files in the `ps_folders` list (it does *not* look in the current folder first). For more information about encoding files, see section 7.6. Note that OzTeX ignores the `Times-Roman` field; it only serves as a reminder of which printer font is represented by the rather cryptic TFM name.

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

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

This tells dvips to reencode ptmr8r.tfm using an encoding vector called TeXBase1Encoding. The vector is downloaded in a file called 8r.enc; dvips looks for 8r.enc in the current folder first, then in the ps_folders list.

NOTE: Screen font reencoding is not as reliable as printer font reencoding because some characters simply don't exist in Mac screen fonts. (Font suitcases normally use the standard Mac text encoding called "Standard Roman Encoding".) This means that OzTEX's preview might not match the output from dvips. Even worse, people with non-PostScript printers won't be able to print the full range of reencoded characters. The solution to this problem is to use a font editor/creator like Fontographer to create suitable Mac fonts, or to purchase such fonts from a supplier like Y&Y. (Y&Y also sell a font manipulation package that lets you create new font suitcases to match PostScript fonts, but it is only available for DOS machines.)

7.4.2 Downloading PostScript fonts

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 dvips provides a mechanism for downloading non-resident PostScript fonts. For example, the following line appears in the supplied psfonts.map file:

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

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

Note that dvips can download font information stored in Macintosh PostScript fonts (i.e., files of type LWFN). 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. You can keep PostScript fonts anywhere you like, but you do need to include the path in the ps_folders list. Such fonts are normally kept in the Fonts subfolder of your System folder. The special $f entry at the end of the ps_folders list in the Default config file will expand to the correct path.

Partial downloading of PostScript fonts is available via the -j option. Another feature supported by dvips is the downloading of PostScript fonts used (but not included) in EPSF files. Such fonts will only be downloaded if they appear in psfonts.map along with a "<" entry specifying the pfa/pfb/LWFN file to be downloaded. See the bottom of psfonts.map in :DVIPS:Inputs: for some examples.

7.4.3 PostScript fonts on a non-PostScript printer

To print a DVI file that uses PostScript fonts on a non-PostScript printer, you must have the corresponding screen fonts installed in your system. For example, if ptmr8r appears in a DVI file then OzTEX will try to use the screen font Times. If Times 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 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_fonts color.

7.4.4 PostScript or Computer Modern?

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

1. A PostScript font can be requested at virtually any size. For example, you can do things like "`\font\helv=ptmr8r at 33.3pt`". You can also ask for `cmr10` at any size, but a separate PK file is needed for each size. (However, you can avoid PK files altogether by getting PostScript versions of the Computer Modern fonts; see below.)

2. When a PostScript font like Times-Roman is scaled to the same design size as a CMR font it tends to look darker and larger. It is not a good idea to mix the two font designs in the same document.

3. For Plain TeX users, the standard TeX commands for accents and foreign letters need to be redefined for a PostScript text font. The file `pstext.tex` in the `Plain` subfolder in `TeX-inputs` contains the required macro definitions. Another file, `psfonts.tex` in `TeX-docs`, inputs these macros and illustrates the use of PostScript fonts in a Plain TeX document.

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

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

will switch the default text font to Times-Roman. See the documentation in the `PSNFSS` subfolder 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.

If you like the look of Computer Modern fonts, but you also want the flexibility of PostScript fonts, then you can have the best of both worlds. PostScript versions of Computer Modern are available at CTAN sites. Choose either the BaKoMa fonts created by Basil Malyshev, or the CM/PS fonts created by Blue Sky Research.

OzTeX provides configuration files for switching to BaKoMa, CM/PS, Lucida or MathTime. The corresponding `config.*` files for `dvips` are in the `:DVIPS:Inputs:` folder.

7.5 Macintosh screen fonts

The font entries listed between `begin_fonts` and `end_fonts` in a configuration file are not restricted to PostScript fonts. The entries are 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 install 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. (The `.enc` files downloaded by `dvips` during PostScript printing serve a similar purpose to OzTEX's encoding files, but the syntax is completely different.)

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_folders` list 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 "Show" item in the Config menu.

The syntax of an encoding file is described in detail in the standard encoding file called `Mac.enc` which is kept in the `Encodings` subfolder 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 TUGboat article [8]. A text version called "Knuth on VFs" can be seen in OzMF's Help menu.

OzTEX and `dvips` process virtual fonts automatically, so most people never need to worry about whether or not a DVI file uses such fonts. To decide if font `foo` is virtual, OzTEX and `dvips` look for a file called `foo.vf` in the current folder, then in the list of folders specified by `vf_folders`.

8 OzMF

OzMF is a companion program for OzTEX which includes various font related tools. The most important of these is METAFONT, a program written by Donald Knuth and used to create TEX-compatible fonts. For more information, see *The METAFONTbook* by Knuth [10]. The other font related tools included in OzMF are:

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

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

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

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

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

`TFtoPL` converts a TFM (TEX font metric) file into a PL (property list) file.

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

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

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

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

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

The OzMF application should be kept in the same folder as OzTEX, along with the `Metafont` folder which contains these subfolders:

`Configs` contains the configuration files used by OzMF, especially the `Default` file which is loaded when OzMF starts up. Like OzTEX, OzMF can be customized by selecting config files from a Config menu.

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

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

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

8.1 Making PK fonts on demand

All of OzTEX's dialog boxes for printing or viewing a DVI file have a check box called "Make missing PK fonts" (only available in System 7 or later). If this box is checked and OzTEX detects a missing PK font, then OzMF will be called to build the required PK file.

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

The location of the PK file is determined by the pk_file template. The template in OzMF's Default file will create PK files in the PK-files folder. If you want MakeTeXPK to create PK files in appropriate subfolders of PK-files then set pk_file to ":PK-files:%d:%f.%dpk". See the comments at the end of the Default file for details about the pk_file template. If you decide to change the template then make sure it matches one of the templates in the pk_files list used by OzTEX so that it can find PK files created by MakeTeXPK!

It is possible that the font creation process will fail. For example, METAFONT might not find the required .mf input file. Or you might decide to cancel the process. In such cases OzTEX will report a missing PK font and append a line of information to a file called missfont.make in the PK-files folder.

If "Make missing PK fonts" is *not* checked then OzTEX won't attempt to create a missing PK font, but it will append a line to missfont.make. This file can be used later by MakeTeXPK to build all the listed PK fonts. For many people this might be more convenient than having OzTEX build PK fonts on the fly — on a slow Mac it can take a while to create a font.

If you don't have a Power Mac, I recommend leaving "Make missing PK fonts" unchecked, at least initially. Preview all your DVI files and select "Page Info" from the View menu. If there are any missing PK fonts, OzTEX will create a missfont.make file. Edit this file and remove any duplicate lines. Start up OzMF, select MakeTeXPK from the Tools menu, open missfont.make and go to bed! When you wake up, all the PK files should have been created.

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

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

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

8.2 Making TFM files

In general, you can't simply open a .mf file in METAFONT to create a TFM file. Most .mf files require a suitable mode and magnification to be set up first, and METAFONT's default mode does not create a TFM file. Assuming you have a file called foo.mf, the solution is to create a foo.make file with a line like this:

```
MakeTeXPK foo 300 300 1.0 laserwriter
```

Before running MakeTeXPK on foo.make, select "Keep TFM files" from OzMF's Config menu. Put the new TFM file into a subfolder in TeX-fonts so that OzTEX can find it, then let OzTEX call OzMF to build new PK files as and when required.

8.3 Setting the Metafont mode

To be able to build a PK file, METAFONT needs to know what mode to use. OzTEX gets this information from either the printing_mode or viewing_mode parameter and simply passes the string on to METAFONT. For dvips, the mode string is specified in config.ps.

WARNING: The given strings must exactly match known mode_def values defined in either modes.mf or macmodes.mf; these files are kept in :Metafont:Inputs:. It is also important to make sure the current printing/viewing resolution matches the pixels_per_inch value in the corresponding mode_def, otherwise METAFONT won't create a suitable PK file.

OzTEX's Default config file sets both modes to laserwriter. You might need to change these modes, either by selecting one of the supplied config files (e.g., StyleWriter), or by editing the Default config file. If a mode string is empty ("") then METAFONT will use the localfont mode which is currently equivalent to the laserwriter mode.

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

NOTE: I've set the stylewriter mode_def parameters in macmodes.mf to be the the same as the laserwriter parameters (except for pixels_per_inch of course). This was done so that both modes can share the same set of PK files. People who are fussy about their pixels (or with better eyesight than me) might want to optimize the stylewriter parameters for that printer.

9 OzMP

OzMP is a Mac implementation of John Hobby's MetaPost program for producing PostScript pictures using a METAFONT-like language. The PostScript files created by MetaPost don't contain a PICT 256 resource, so you'll need a PostScript printer to print a DVI file that includes such files. The OzMP application should be kept in the same folder as OzTEX, along with the MetaPost folder which contains these subfolders:

Configs contains the configuration files used by OzMP, especially the Default file which is loaded when OzMP starts up. Before using MetaPost, examine the supplied parameters in the Default file and make sure the search paths for TFM/VF files match those used by OzTEX. Also read the comments at the bottom of the file.

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

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

Inputs contains *.mp files.

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

OzMP includes two supporting tools, MPtoTeX and DVItoMP. If you are using System 7 or later then you won't need to run these tools manually because OzMP will call them automatically if your .mp file contains btex ... etex or verbatimtex ... etex sections. Such sections contain TEX/LATEX code, so OzMP will also call OzTEX in between MPtoTeX and DVItoMP.

10 DVIPS

OzTEX uses Tom Rokicki's dvips program to convert DVI files into PostScript. The DVIPS folder contains all the necessary supporting files in a couple of subfolders:

Inputs contains various input files required by dvips, such as config.ps, psfonts.map and the *.pro files.

Docs contains the dvips documentation and some test files. To print off a copy of the dvips manual, run dvips.tex through TEX and print/view dvips.dvi.

The version of dvips included in OzTEX contains a number of enhancements:

- All the OzTEX-specific \special commands for including PICT/PNTG/EPSF files are supported. I've also added support for OzTEX's method of including raw PostScript files followed by arbitrary PostScript code, although this is not something I recommend (EPSF files are much safer). Because of conflicts with dvips-specific \special commands, there are some rare situations where inclusion of a raw PostScript file will no longer work. In such cases you will need to add a rawps keyword; e.g., \special{rawps=foo.ps ...}.

- dvips can download Mac PostScript font files (of type LWFN) as well as the usual pfa/pfb files. This is very handy if you want to use the BaKoMa or CM/PS fonts.

- A number of config.* files have been added to :DVIPS:Inputs: so that you can pass an extra option like "-P CMPS" to tell dvips to load config.CMPS after config.ps; config.CMPS contains the line "p +psfonts.CMPS" which tells dvips to append the info in psfonts.CMPS to psfonts.map. This is useful if you don't want to make changes to the latter file.

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

- The dvips output can be sent directly to the current printer.

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

TEXCONFIG	= ps_folders	(to find config.ps, psfonts.map)
DVIPSHEADERS	= ps_folders	(to find .pro/.enc/.pfa/.pfb/LWFN files)
TEXINPUTS	= input_folders	(to find \special files)
TEXFONTS	= tfm_folders	(to find TFM files)
VFFONTS	= vf_folders	(to find VF files)
TEXPKS	= pk_files	(to find PK files)

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

In most operating systems, dvips uses the above environment variables to override the settings in config.ps. In OzTEX's case, config.ps can be used to override the above parameters, but it is not something I'd recommend doing unless you are an experienced dvips hacker. If you need to do this, note that Mac path names use colons to separate folders, so you must use commas to separate path names.

NOTE: dvips does *not* use OzTEX's printing_resolution or printing_mode parameters. It uses the resolution and mode specified in config.ps. This approach allows more flexibility. For example, a person with two printers, one QuickDraw and one PostScript, could set up OzTEX's printing parameters for the QuickDraw printer and have quite different settings in config.ps for the PostScript printer.

11 Graphics

Creating illustrations with standard TEX commands 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. (LATEX provides a `picture` environment, but it is very inefficient and only suitable for small, simple diagrams.)

TEX does however provide a `\special` command that can be used to pass arbitrary information to the DVI reader. For example, OzTEX can interpret a `\special` command containing the name of a graphic file and then include that file in the preview or printed output.

Note that using `\special` can seriously reduce the portability of your TEX 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 TEX file to another computer all you need to do is change the macro definition (assuming the DVI translator supports similar functionality in its handling of `\special`).

LATEX users should never need to use explicit `\special` commands. The `graphics` and `graphicx` packages make it easy to include graphic files. Plain TEX users can also use macro packages; see Larry Siebenmann's `boxedeps.doc` and `boxedeps.tex` in the `Plain` subfolder in `TeX-inputs`. Another alternative is `epsf.tex`; see `epsftest.tex` in the `TeX-docs` folder. Using a macro package will make your input files much more portable.

11.1 Including Macintosh graphic files

A `\special` command can appear almost anywhere in an input file. It behaves like an invisible box of zero height and width. TEX simply stores the given information in the DVI file at the current page position. When previewing a DVI file, OzTEX 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.

OzTEX 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 OzTEX converts the PICT file into a black and white bitmap for previewing and printing, so you won't see any colors. The bitmap is always at 72 dpi when previewing, but the current printing 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 color 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 OzTEX uses when PostScript printing, and a corresponding PICT resource (ID = 256) which OzTEX 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=FOO}` 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 `input_folders` list. 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 subfolder 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/EPSF `\special` commands all allow you to preview graphics in your DVI file, but if you have a choice, use EPSF files because:

1. The output on a PostScript printer should be of much higher quality than a similar PICT/PNTG image.

2. A colored EPSF picture will print correctly on a PostScript printer (assuming it supports color) even though OzTEX will convert its PICT resource into a black and white bitmap when previewing.

3. Macro packages like `boxedeps.tex` and `epsf.tex` can extract the bounding box information in an EPSF file and position the image automatically.

4. Encapsulated PostScript is supported on many other computers.

Some EPSF files, especially those created on other computers or by programs like MetaPost, do not contain a PICT 256 resource. When previewing a DVI file that includes such files, OzTEX will draw a rectangle showing the bounding box, along with the name of the file and the message "no PICT" near the `\special` marker. If you try to print the DVI file on a non-PostScript printer then OzTEX will generate error messages.

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 EPSF 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 available (for 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/EPSF `\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 bp = 1 in). If no previous `height` keyword has been used then the vertical scale is set to the new horizontal scale to preserve the aspect ratio.
- `height=`*bp* – sets the vertical scale to get a picture at the desired height. If no previous `width` keyword has been used then the horizontal scale is set to the new vertical scale to preserve the aspect ratio.

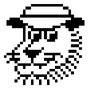
If a particular keyword is used more than once in a single `\special` command then the last value is used. An illegal keyword or value will produce an error message when previewing or printing. Synonyms exist for most keywords:

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

As mentioned earlier, it is not advisable to use raw `\special` commands. It is much better to let a high-level macro package generate them. Here are some examples of the `\includegraphics` command available in LaTeX's `graphicx` package:

1. To include a picture at "normal" size:

   ```
   \includegraphics{ozicon.eps}
   ```

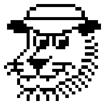


2. To halve the size:

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

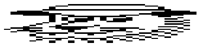
3. To set the height to 0.5 in and preserve the aspect ratio:

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



4. To set the height to 0.2 in and the width to 1 in:

   ```
   \includegraphics[height=0.2in,width=1in]{ozicon.eps}.
   ```



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

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

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

11.2 Compatibility with other DVI translators

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

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

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

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

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

11.3 Including PostScript files

OzTeX allows you to include a "raw" PostScript file using TeX's `\special` command. However, if you have a choice then any of the methods described in the above sections are preferable because OzTeX cannot preview such a file, nor can it print the file on a non-PostScript printer.

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

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

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

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

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

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

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

11.4 Rotating TEX material

If you have a PostScript printer then rotating arbitrary TEX material is quite simple. LATEX users can use the `\rotatebox` command from the `graphics/graphicx` packages; see the examples in `:TeX-inputs:Graphics:grfguide.tex`. Plain TEX users should check out the examples in `:DVIPS:Docs:rotsample.tex`.

11.5 Color

OzTEX's previewer supports the color `\special` commands used by `dvips` and generated by LATEX's `color` package. The `hyper.tex` file in `LaTeX-docs` shows how the `color` package can be used. For more details, LATEX users should read `:TeX-inputs:Graphics:grfguide.tex` and Plain TEX users should read the final section in the `dvips` manual.

If your DVI file uses color `\special`s then you should enable the "Prescan color `\special`s" option in the view dialog. Prescanning might be needed because OzTEX allows you to jump to any page, but there might be a `\special` on an earlier page that sets the text color (or the background) for all future pages. The prescan option tells OzTEX to do a quick scan of `\special`s on all earlier (unseen) pages to make sure that the colors on the current page are correct. Although this scan is quite fast — a mid-range Mac takes about 2 seconds to scan 50 pages — it is still slow enough that you should keep the prescan option disabled if you never use color. OzTEX remembers the state of the prescan option.

When OzTEX sees a page for the first time, the color `\special`s on that page are remembered. As more pages are seen, the prescan will take less time. In fact, if you preview pages 1, 2, 3, etc., the prescan isn't done at all.

NOTE: The color `\special`s are not yet supported for QuickDraw printers; that will have to wait until I buy a color printer.

11.6 HyperTEX

OzTEX's previewer supports the HyperTEX `\special` commands. These `\special`s let you jump to another place in a document by simply clicking on a link, such as a reference to a section number.

The `hyperref` package (in `:TeX-inputs:HyperRef:`) makes it very easy for LATEX users to add HyperTEX links to their documents. Just add the line `\usepackage{hyperref}` before `\begin{document}` and run LATEX a couple of times.

To see how HyperTEX links work, preview the DVI files in the `LaTeX-docs` folder. All links are underlined and, on a color monitor, are colored green. If the mouse moves over a link then the cursor changes to a pointing finger. Clicking on a link will take you to the target.

The "Back Link" and "Forward Link" items in the View menu make it easy to move to previously visited links; they work just like the Back/Forward buttons in a Web browser. Their keyboard short cuts are "`[`" and "`]`" respectively.

OzTEX also supports links to other DVI files on your Mac and to URLs anywhere on the Internet; see page 3 in `hyper.dvi`. Links to URLs require Internet Config (version 1.1 or later) and appropriate helper applications to be installed. Internet Config is free and available for downloading from `ftp://ftp.share.com/internet-configuration/`.

The HyperTEX `\special` commands are also supported by `dvips`. If the `-z` option is given then `dvips` will generate "HyperPostScript"; this can be run through Adobe Distiller to produce a PDF file with hypertext links. For more information, see the TUGboat article by Mark Doyle [4], or visit `http://xxx.lanl.gov/hypertex/`.

12 Recovering from errors

OzTEX reports errors by displaying a message in the OzTEX window or in a dialog box. In the latter case there are two types of error dialogs:

1. Warning messages appear in a box with two buttons: Cancel and Continue. The former lets you cancel a whole heap of `\special` errors when viewing a DVI file (try page 16 in `nasty.dvi` for example).

2. Fatal messages appear in a box with a Quit button (which will quit OzTEX).

Most errors, particularly the serious ones, are accompanied by a beep. Let's now look at the most likely errors you'll encounter and discuss what you can do to prevent them recurring.

12.1 Errors while loading a configuration file

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

12.2 Errors while loading an encoding file

These errors can occur when OzTEX is loading a configuration file containing a list of PostScript fonts. If OzTEX complains that it can't find an encoding file, check the `ps_folders` parameter in the configuration file. The encoding files supplied with OzTEX are all stored in the `Encodings` subfolder in `PS-files`. OzTEX does *not* look in the current folder for an encoding file.

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

12.3 Errors while running TEX

Nearly all the errors you'll see when running TEX will be the result of mistakes in your input file. *The TEXbook* [9] is the best place to look for solutions, especially Chapters 6 and 27. Lamport's LATEX manual [12] also has a section devoted to errors. The following errors are more likely to occur due to OzTEX's particular implementation of TEX.

Can't find file

There are a number of likely reasons why TEX can't find a file:

1. The file does not exist in either the current folder or in any of the subfolders within `TeX-inputs`. If TEX was looking for a format file then make sure it is in the `TeX-formats` folder.

2. You could have spelt the file name incorrectly.

3. The file name might contain an illegal character. A Macintosh file name can contain just about any character, but TEX is not so flexible. Don't use spaces and avoid non-alphabetic characters in all file names that might be seen by TEX.

If you see such an error then the best thing to do is hit Command-dot to abort TEX and either rename the file or move it to the correct folder.

Not enough memory

The most likely time you could run out of memory is when using `INITEX` to create a format file. `INITEX` typically requires more memory than TEX — just how much more depends on the TEX parameters in your configuration file. If you do run out of memory then quit OzTEX, select the OzTEX application icon, choose "Get Info" from the Finder's File menu, and increase the preferred size.

Fatal format error

You probably changed one of the critical TEX parameters in your configuration file and forgot to rebuild the format. Run `INITEX` and rebuild the offending format file (see section 6).

TEX capacity exceeded

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

Look at the list of TEX parameters in your configuration file and check if the exceeded parameter appears there; if it is listed then you might be able to solve the problem by simply increasing the parameter's value and reloading the configuration file. Note that if you change the value of `mem_top`, `hash_size`, `hash_prime` or `hyph_size` then you will also need to rebuild all format files (see section 6).

12.4 Errors while viewing a DVI file

Missing fonts

During interpretation of a DVI page you may get a warning message if OzTEX can't find a PK file, or if a PostScript font's matching screen font is not installed. Such warnings will only appear once for each missing font (but not at all if the "Ignore missing fonts" check box was selected). The "Page Info" item indicates all missing fonts. METAFONT can be used to build missing PK files; see section 8.

Page off paper

If OzTEX detects that any part of a DVI page is off the paper then it will beep and show a full view. TEX will often catch these errors (e.g., overfull `\hbox` or `\vbox`) but there are quite a few situations where it can't. TEX really has no idea about what paper size you intend to print its pages on.

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

`\special` errors

When interpreting a DVI page OzTEX will check any `\special` commands and warn you about the following problems:

- An unknown keyword.
- A file that can't be opened.
- A file type that does not match the first keyword (PICT/PNTG/EPSF).

- A bad value after an optional keyword.

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

12.5 Errors while printing a DVI file

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

To avoid wasting paper you should always view a DVI file before printing it and choose "Page Info" to check for any missing fonts (but don't worry about missing screen fonts if you have a PostScript printer). Another way to check for errors is to send PostScript output to a file instead of the printer, although this won't detect certain errors that can only occur with a non-PostScript printer.

Non-PostScript scaling errors

Before OzTeX starts non-PostScript printing, it checks to see that the `printing_resolution` parameter matches the actual resolution of your printer. If the resolutions don't match then OzTeX will warn you that scaling problems may occur. This warning appears just before the print dialog in case you want to Cancel printing (you might have forgotten to switch to the correct configuration file).

If you decide to go ahead and print the DVI file, you might see problems such as horizontal gaps at regular intervals down each page. OzTeX prints a page by sending a sequence of horizontal bands to the printer. The number of bands depends on how much memory is available (the band height is displayed when showing statistics). Scaling problems can result in gaps or overlapping at band boundaries, but they can be avoided by bumping up OzTeX's memory to a suitably large value so that only one large band is needed to print each page.

Printer errors

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

Can't open `\special` file

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

Bad `\special` command

If OzTeX detects any sort of error in a `\special` command then a suitable message will be displayed in the OzTeX window. Note that you will get an error message if you try to print a DVI file that includes a "raw" PostScript file on a non-PostScript printer. See section 11 for the `\special` syntax and semantics supported by OzTeX.

PostScript errors

You shouldn't see any errors in the PostScript code generated by `dvips`, but there may well be problems in a PostScript file included by `\special`. (You can always tell which is the culprit by commenting out the `\special` command.) If you are sending the output to a printer rather than spooling software then any PostScript error should cause a message to appear in the OzTEX window. Any further output will be flushed and printing cancelled.

So what can you do? For starters, get the *PostScript Language Reference Manual* by Adobe Systems [1]. Debugging a PostScript program can be a frustrating experience. Try downloading the error handler stored in `errhandler.ps`. If the problem isn't obvious then use `print` and `flush` commands to isolate just where the error is occurring. The Ghostscript program mentioned in the "Related Software" help file can also be useful.

13 Changing resource data

This section is for people who know how to use a resource editor like ResEdit. Most user preferences for OzTEX are stored as resources in a file called `OzTeX Prefs`. You might like to change the following resources in this file:

- STR 200 and 201 specify the location and name of the default configuration file.
- STR 240 and 241 specify the dimensions of the magnified rectangle that appears after clicking in the view window. If you want the magnified area to fill the view window then set both numbers to huge values (8000 should do). Setting either number to a value less than 5 will prevent any magnification.
- STR 300 to 316 specify the keyboard short cuts used when previewing (see section 2.5.7). All the resources are named, so it is easy to see which string corresponds to which short cut. OzTEX uses the first character in each string. You might need to open a string resource with the hex editor if you want to insert a non-displayable character like Delete or Return.

Some read-only information is stored as resources in the OzTEX application. For example, STR 12345 specifies the name of the preferences file. The MENU resources can also be modified. You might want to change the name of a menu item or add a new Command-key equivalent.

Note that OzMF and OzMP have their own preferences files.

14 Transferring TFM/VF/PK/DVI files

OzTEX 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 fro without any further processing.

For the purposes of data transmission a TFM/VF/PK/DVI file must be treated as a binary file (a stream of arbitrary 8-bit bytes). 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 OzTEX 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 OzTEX 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.

15 Getting more information

Many TEX-related books are available through TUG, the TEX Users Group. They also produce an excellent magazine called TUGboat. Anybody planning to use TEX a lot is strongly advised to join. The "TEX Users Group" item in the Help menu explains how to get in touch.

If you have access to the Internet then another good source of help and information about TEX-related topics is the newsgroup `comp.text.tex`. The following World Wide Web sites are also worth visiting:

- TEX Users Group: `http://www.tug.org/`
- Donald Knuth's home page: `http://www-cs-staff.Stanford.EDU/~knuth/`
- Gary Gray's TEX/LATEX software for the Mac: `http://www.esm.psu.edu/mac-tex/`
- Tom Scavo's LATEX info: `http://web.syr.edu/~trscavo/latex.html`
- AMS list of TEX references: `http://e-math.ams.org/tex/publications.html`
- Text processing with TEX: `http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/`
- (La)TEX Navigator — a (La)TEX encyclopaedia: `http://www.loria.fr/tex/english/`
- OzTEX's home page: `http://www.kagi.com/authors/akt/oztex.html`

The next page has a list of recommended reading material. For Plain TEX users and all serious TEX hackers, [9] is essential. For LATEX users, [12] and [5] should be adequate for all but the largest typesetting jobs. Also included are a number of other books recommended by OzTEX users; see [6], [11], [3], [14], [2] and [13].

* * *

References

[1] Adobe Systems Incorporated. *Postscript Language Reference Manual.* Addison-Wesley, Reading, MA, USA, 1985.

[2] Wolfgang Appelt. *TEX für Fortgeschrittene.* Addison-Wesley, Reading, MA, USA, second edition, 1994.

[3] Michael Doob. *TEX: Starting from 1.* Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1993.

[4] Mark D. Doyle. The Los Alamos e-print archives: HyperTEX in action. *TUGboat*, 16(2):154–159, June 1995.

[5] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LATEX Companion.* Addison-Wesley, Reading, MA, USA, second edition, 1994.

[6] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LATEX Graphics Companion: Illustrating Documents with TEX and PostScript.* Tools and Techniques for Computer Typesetting. Addison-Wesley, Reading, MA, USA, 1997.

[7] George Grätzer. *Math into LATEX: an introduction to LATEX2e and AMS-LATEX 1.2.* Birkhäuser, Basel, Switzerland, 1996.

[8] Donald Knuth. Virtual fonts: More fun for grand wizards. *TUGBoat*, 11(1):13–23, April 1990.

[9] Donald E. Knuth. *The TEXbook.* Addison-Wesley, Reading, MA, USA, 1984.

[10] Donald E. Knuth. *The* METAFONT *book*, volume C of *Computers and Typesetting.* Addison-Wesley, Reading, MA, USA, 1986.

[11] Helmut Kopka and Patrick W. Daly. *A Guide to LATEX2e: Document Preparation for Beginners and Advanced Users.* Addison-Wesley, Reading, MA, USA, second edition, 1995.

[12] Leslie Lamport. *LATEX: A Document Preparation System: User's Guide and Reference Manual.* Addison-Wesley, Reading, MA, USA, second edition, 1994.

[13] Norbert Schwarz. *Einführung in TEX.* Addison-Wesley, Reading, MA, USA, second edition, 1988. Available in English and Dutch translations.

[14] Michael D. Spivak. *The Joy of TEX — A Gourmet Guide to Typesetting with the AMS-TEX macro package.* Addison-Wesley, Reading, MA, USA, 2nd revised edition, 1990.

Index