User Guide

Version 1.4, January 1992

Andrew Trevorrow

c/- Research School of Earth Sciences

Australian National University

GPO Box 4, Canberra 2601, Australia

E-mail: akt150@huxley.anu.edu.au

is a public domain version of TEX for the Macintosh. This document assumes you know how to use a Macintosh. It also assumes you are familiar with TEX, the typesetting system developed by Donald Knuth at Stanford University. An understanding of PostScript, the page description language developed by Adobe Systems, is not essential but would be useful.

aims to provide a standard TEX environment for the Macintosh that can easily be extended or customized to suit your particular needs. If you have had experience with TEX on some other computer then the way works shouldnjt be too surprising.

Comments, bug reports and suggestions are all welcome. I have plans for further development of , but at this stage such work can only be carried out in my spare time. If anybody out there would like to sponsor such work then please get in touch.

Contents

1 Introduction

Assuming you correctly carried out the installation instructions in the `Read-Me` file supplied with ,
your screen should display something like this:

to 3.4in pntg=folders.pntg scale=0.72

Here is a brief description of the various folders:

The `Configs` folder contains a number of configuration files.

The `TeX-fonts` folder contains a standard set of TFM files. These files are required by
TEX to carry out typesetting; see section ? for more information about TFM files.

The `TeX-formats` folder contains some format files used by TEX and the `TeX.pool` file
read by `INITEX`. Section ? describes how to create format files.

The `TeX-inputs` folder contains a large number of TEX input files. TEX looks in this
folder if it canjt find a given file in your current folder.

The `Help-files` folder contains a number of text files.

The `PS-files` folder contains various PostScript files.

The `TeX-docs` folder contains examples of TEX input files, including `nasty.tex` which
is used in the quick tour of .

The `LaTeX-docs` folder contains the source files for this User Guide.

The `?Edit` folder contains the *Edit desk accessory and its documentation.

The `DVIM72-Mac` folder contains the `DVIM72-Mac` program and its documentation.

The `PK-files` folder contains PK files stored in various sub-folders. The name of a sub-
folder indicates the ksizel of the PK files kept within it; see section ? for how this size is
calculated and for more information about PK files.

1.1 Changing the default configuration file

reads a configuration file called `Default` when starting up. This file is kept in the `Configs` folder
along with other configuration files. (The name and location of the default configuration file are stored
as STR resources in the application file, so you can easily change them using a resource editor like
ResEdit.)

A configuration file is a simple text file which you can edit and change various parameters that
control js behaviour. Some of the more important parameters are:

A list of the configuration files that will appear in the Config menu. This list should only
appear in the `Default` file; it will be ignored in every other configuration file.

A list of the formats that will appear in the TEX menu. Place your preferred format first.

A list of the help files that will appear in the Help menu. Feel free to add your own help files
or remove them entirely (the Help menu wonjt appear if the list is empty).

The printer resolution and viewing resolution. The former is used to print a DVI file and the
latter for previewing. If you have a set of 72 dpi PK files then you could set the viewing resolution to 72; this value matches the resolution of the
standard Mac screen, resulting in a more readable display (itjs also faster and uses much less memory).

The paper dimensions used to detect page-off-paper errors when printing/viewing a DVI file. If the paper width is greater than the height then will print/view in landscape mode. The
`Default` values specify A4 paper in portrait mode.

The names of certain special folders: `TeX-formats`, `TeX-inputs`, `TeX-fonts`, `Help-files`, `PS-files` and `PK-files`. The supplied strings are partial path names relative to
the location of the application, but you might prefer to store certain files elsewhere. For example, you could tell to look for PK files on a different volume by changing the path name of the
PK folder to kMyDisk:PK-files:l.

The names and locations of special files: `TEXTtoPS.ps`, `DVItoPS.ps`, `TeX.pool`, etc.

The colours of various elements seen when viewing a DVI file (paper edges, `\special` markers and missing fonts). The allowed colours are black, red, green, blue, cyan, magenta,
yellow and white (white is allowed in case you want to do something like make `\special` markers invisible).

The values of important TEX parameters that determine how much memory will be allocated for certain arrays. The `Default` values are suitable for a Mac with 1MB of memory. If you
have more memory then you may want to increase many of the parameters. If you do then youjll need to increase js application memory size for use with MultiFinder or System 7 (see the
kGet Infol item in the Finderjs File menu). Note also that a change to mem_top, hash_size, hash_prime or hyf_size will require the format file(s) you use to be rebuilt (see section
?).

A list of all PostScript TFM file names and their corresponding printer-resident font names and screen font names. Each screen font has a particular encoding scheme (Adobe or Macintosh)
and an optional style.

The `Default` configuration file contains many comments describing the purpose of each parameter and how to go about changing them. Before doing so it might be a good idea to save a copy first. In
your new `Default` file you can then remove all the comments so that starts up a little faster.

1.2 Creating other configuration files

js Config menu lets you switch rapidly from one configuration file to another. This is especially useful if you have access to more than one PostScript printer, since each printer might require a separate
configuration file (they might have different resolutions or support a different set of resident fonts).

There are other uses for multiple configuration files. See the examples supplied in the Configs folder. It is a simple matter to create your own and add them to the list in the `Default` file. To help
keep configuration files short and simple, a couple of special characters can be used:

1. If k?l is the first character of a parameterjs value then the parameter is not changed and the rest of the line is ignored.

2. k!l is similar but in addition all remaining parameters are left unchanged (the rest of the file is ignored).

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

2 A quick tour of

The aim of this section is to get you quickly acquainted with js major features.

2.1 Starting

Assuming youjve installed everything, double-click on the application file to get the program started. After a brief pause you should see a window with the title kl and a message indicating the name of the default configuration file. (If you made a mistake during installation, especially when editing the `Default` file, then you might also be looking at a dialog box from which you can only quit. The message displayed will hopefully give you some idea of how to fix the problem.) The window is always present; note that it has no close box in the upper left corner. This window provides a terminal-like interface for .

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.) Letjs go through this cycle step by step.

2.2 Editing a TEX input file

A TEX input file is a standard Macintosh text file. We wonjt create an input file from scratch, instead wejll 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 wejll 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 youjll encounter when using in the future.

Use any text editor to open `nasty.tex` in the `TeX-docs` folder. The only change wejll 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 .

2.3 Running TEX

Letjs 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 see which format item is ticked. It should be kPlainl, unless you changed the format list in the `Default` configuration file. This is the format required by `nasty.tex` so we donjt need to change it. (To change the current format you simply select a different one.)

Choose kTEX ?l from the TEX menu and open `nasty.tex`. The window will be cleared and brought to the front (if it isnjt already) and after a short pause TEX will start up, load the Plain format, and begin reading the given input file.

When it sees the unknown commmand, TEX will print a suitable error message, beep, and wait for you to type something. Note that a solid block cursor sits next to TEXjs k?l prompt. This block cursor always appears when TEX is waiting for you to type something. The window should look like this:

```
        This is TeX, Version 3.14 (no format preloaded)
**&Plain nasty.tex
(nasty.tex
! Undefined control sequence.
l.5 \xxx

? width .4em height .7em depth .15em
```

Chapter 6 in *The* TEX*book* explains what you can do in such a situation. What we will do is simply ignore the error and continue by hitting the Return key. Although `nasty.tex` is not very nice, you shouldnjt see any more TEX errors. A 19-page DVI file called `nasty.dvi` should be created.

Have a look at the second item in both the File and View menus. Whenever a new DVI file is created its name will automatically appear in these items so you can quickly print or view the DVI file without going through the standard file dialog. The second item in the TEX menu has also been updated with the name of the most recent input file (`nasty.tex` in this case) so you can easily typeset the same file many times.

2.4 Viewing a DVI file

After generating a DVI file, the next step is to preview it and check for problems that TEX may have missed, like missing fonts, bad page breaks, spelling mistakes, etc.

Letjs have a look at `nasty.dvi` by choosing the second item in the View menu. creates a new window, called the kviewl window, and sets its title to the selected DVI file. A dialog box has also appeared. It has a scroll bar that allows you to select any page in the DVI file. (The dialog box also allows you to change a couple of viewing options, but wejll ignore them at this stage; see the description of kView DVI ?l in section ? for details.) Play around with the scroll bar if you like but make sure it shows the 2nd DVI page before proceeding. To view this page simply click in the default 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. chooses the initial view based on the current size and shape of the view window and the current paper dimensions. It will try to show as much of the paper as possible. The top, left and right paper edges should all be visible (and maybe the bottom edge as well if you have a large screen).

Now is a good time to adjust the size and location of the and view windows to suit your screen. When you quit, will remember the current settings and use them the next time it starts up.

2.4.1 Paper coordinates and the scale factor

An understanding of the coordinate system used by to display (and print) a DVI page would be useful. You may have noticed the kpxl units mentioned in the `Default` configuration file. These are kpaper pixelsl and are js internal units. The resolution parameters in a configuration file define the number of paper pixels per inch; one parameter is only used when printing a DVI file and the other is only used when viewing a DVI file.

 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 kTEX originl since it is also the origin of the coordinate system used in DVI files. Every rule/character/`\special` on a page 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.

A kscale factorl 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.

2.4.2 Zooming in and out

Certain View menu items change the current scale factor:

=0pt

kFull Viewl sets the scale factor to its maximum value and displays the entire page and paper edges in the middle of the view window.

kActual Sizel sets the scale factor so that the new view will show the page at roughly the right size (itjs only approximate because the scale factor is an integer value).

kZoom Inl halves the current scale factor.

kZoom Outl doubles the current scale factor.

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

=0pt

Click-and-drag within the viewing area to zoom in to the selected rectangle. Cancel the operation by making the rectangle very thin (not small, because that may be interpreted as a simple click).

A simple click will zoom in by halving the scale factor.

Command-click will zoom in by decrementing the scale factor.

Option-click will zoom out by incrementing the scale factor.

Shift-click will zoom out by doubling the scale factor.

A beep occurs if you canjt zoom in or out any further.

When you zoom in or out by clicking, will try to make the point you clicked the middle of the new view. However, when you choose a menu item, 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 . Experiment with the various ways of zooming to see which methods you prefer.

The most precise display occurs when the scale factor is 1 since each screen pixel corresponds to exactly one paper pixel. doesnjt allow you to zoom in any further than this.

2.4.3 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. wonjt let you get too far away from the page/paper boundaries. If you do manage to get lost just choose kFull Viewl.

2.4.4 Checking for errors

It is a good idea to select the kPage Infol item at least once while viewing a DVI file. This item displays its results in the window. The display includes a list of all the fonts used in the DVI file and clearly indicates any that are missing. It also lists any `\special` commands on the current page, showing their locations and arguments.

2.4.5 Selecting pages

There are a number of interesting pages in `nasty.dvi`. Use kPrevious Pagel, kNext Pagel or kSelect Page ?l to have a look at some of them. Note that the current DVI/TEX page numbers are always displayed in the view windowjs title. Here are some points of interest:

The current scale factor and page location will only change if the new page is off the paper, in which case 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, reverts back to the default scale and location.

 may take a little while to interpret and display a selected page, depending on how complicated it is. Hit Command-C or Command-Dot if you get bored.

The view window is updated in the following manner. Visible paper edges are drawn first, then `\special` bitmaps and markers, then rules, then characters on a font by font basis, starting with the font that has the least number of characters on the page. Try a full view of page 19.

Page 3 has examples of `\special` commands. is able to preview included files of type PICT, PNTG or EPSF. The location of each `\special` is indicated with 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.

All the fonts used on page 12 are deliberately missing.

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

2.5 Printing a DVI file

To print `nasty.dvi` choose either of the top two items in the File menu. Choosing the second item simply avoids the standard file dialog.

2.5.1 Changing print options

will present a dialog box that lets you to change a variety of print options. Most of them should be fairly obvious n details can be found in the description of kPrint DVI ?l in section ?. The option youjll most often want to change is the page range. (If you find yourself changing a check box option nearly every time then you should probably change its default setting in the `Default` configuration file.) The two scroll bars control the first and final pages. prevents you from choosing a first page greater than the final page. Play around with the scroll bars but restore their values so that every page will be printed. Click in the Print button or hit Return.

2.5.2 The printing process

will look for the printer most recently selected by the Chooser. If this isnjt a PostScript printer then you wonjt get very far. If the printer is found then a status box will appear and keep you up-to-date on your jobjs progress and the state of the printer. As each page is translated into PostScript its DVI/TEX page numbers are displayed in the window.

You canjt 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 youjre using A4 paper) but the rest will generate nearly every type of error message youjre 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. (may start translating page 18 before the error is seen.)

2.5.3 Saving output in a file

The print dialog box also lets you send the PostScript output to a file instead of the printer. This file is an ordinary text file with a default name of `Oz.ps`. A warning: the PostScript code generated by is resolution-dependent. If you send this code to a PostScript device with a resolution different to that of the current printer resolution parameter then the output will look a little strange! (Use kShow Configl to see the current value of the printer resolution parameter.)

2.5.4 Positioning pages correctly

When printing a DVI file, changes the PostScript coordinate system to match the paper coordinate system described in section ?. For your output to appear in precisely the right location on the paper the TEX origin must be exactly one inch (2.54 cm) in from the top and left paper edges. canjt always set the TEX origin automatically to the correct position because some PostScript printers donjt provide an accurate way of of finding out the exact paper dimensions and the relative location of the PostScript origin.

Page 2 in `nasty.dvi` contains a thick rule with its top left corner exactly at the TEX origin. If the printed page shows a significant discrepancy then change the horizontal and vertical offsets in a configuration file to shift the origin to the correct position. If you use more than one printer then you may need a separate configuration file for each.

Choose offset values based on the normal portrait paper orientation. The offsets are adjusted automatically so that the landscape origin should also be correct (if it isnjt then youjll have to create a separate configuration file for landscape printing, or if you know a bit of PostScript then alter some numbers in `DVItoPS.ps`; search for the string kTeX originl in that file).

3 Fonts

uses the following sources of font information in the process of typesetting, previewing and printing a document: TFM files (usually stored in `TeX-fonts`), PK files (usually stored in various folders nested within `PK-files`), and Macintosh screen fonts (normally found in the System file).

3.1 TFM files

A TEX Font Metric file contains the crucial typesetting information about a font, such as each characterjs height, depth and width. The actual character images are not stored in a TFM file (that information is kept in PK files, or, in the case of a PostScript font, inside the printer). TFM files are the only source of font information used when running TEX, since it doesnjt need to know anything about character images to be able to create a DVI file. See *The* TEX*book* by Donald Knuth if you want to know more about how TEX uses fonts, especially Chapter 4 and Appendix F.

also reads TFM files when printing or viewing a DVI file containing PostScript fonts (based on the list of TFM names at the end of your configuration file), or if a requested PK file canjt be found (in which case the dummy TFM file specified in your configuration file will be used).

will look for a TFM file in the current folder before looking in the default `TeX-fonts` folder. Most of the TFM file names in `TeX-fonts` begin with kcml. These letters stand for Computer Modern, a family of fonts created by Donald Knuth using METAFONT. All the PostScript TFM files supplied with have names beginning with kps-l; this is not required but does make them easy to recognize. If you want to change their names then remember to update your configuration file(s).

3.2 PK files

PK (packed pixel) files store the character images needed to print or view a DVI file containing non-PostScript fonts (those NOT appearing in the list at the end of your configuration file). For each such font there is usually a number of PK files, each one representing the same font but at a different size. This size, which has no relation to the fontjs design size, is calculated as follows:

$$size = resolution * magnification$$

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

Although 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 TEXjs `\magstep` values. See Chapters 4 and 10 of *The* TEX*book* for more details. For example, assuming a *resolution* of 300 and no document magnification:

TEX input	size	PK file
[6pt] `\font\ra=cmr10`		