System Guide

Version 1.3, September 1990

Andrew Trevorrow

1-3-37 5th St., Habsiguda

Hyderabad 500007, India

This document is aimed at programmers who would like to modify or carry out further development. It assumes you have a strong background in TEX and programming experience on the Mac, especially with Modula-2 under MPW (Macintosh Programmer's Workshop). If you are planning to make changes to then please get in touch so that we can coordinate our efforts.

Contents

1 What you need

1.1 Modula-2 and MPW

was developed using TML Modula-2 v1.10 which runs as a tool under MPW (version 2 or 3). TML Systems no longer distribute the Modula-2 compiler, but a much newer and presumably improved version is now available from Metrowerks:

(in the United States) Metrowerks, Inc. The Trimex Building, Route 11 Mooers, NY 12958 U.S.A.

phone: (514) 458-2018 (Canada and international) Metrowerks, Inc. 548 Main, m/s 95 Hudson Heights, J0P 1J0 Canada

MPW is available through APDA (AAPDA in Australia or ADG in the UK). It's worth very cent. I've found it to be the best programming environment I've come across.

1.2 Contents of `Sources` disk

All the source code for is packed into a StuffIt archive on the `Sources` disk. It contains the following files and folders:

`Build-OzTeX`

This is the MPW worksheet used to build and carry out other miscellaneous tasks. All the MPW commands for compiling resources, compiling source modules and linking the final application are kept in this file.

`Sources`

This folder contains Modula-2 definition modules (`*.DEF` files), their matching implementation modules (`*.MOD` files), and various Rez input files (`*.r`).

`Symbols`

An empty folder. `Build-OzTeX` uses this folder to store all the `*.SBM` files created by compiling `*.DEF` files.

`TeX-sources`

This folder contains all the TEX-related modules (`TeX*.DEF` and `TeX*.MOD` files).

`TeX-objects`

An empty folder. `Build-OzTeX` uses this folder to store all the `TeX*.MOD.o` files created by compiling `TeX*.MOD` files. This scheme simplifies the MPW `link` command.

`Objects`

An empty folder. `Build-OzTeX` uses this folder to store all other `*.MOD.o` files.

`Trip`

This folder contains the files used to carry out Knuth's torture test for TEX 3.0.

1.3 Contents of WEBtoMOD disk

The WEBtoMOD disk has a StuffIt archive containing all the MPW tools and scripts used to convert version 3.0 of tex.web into Modula-2. tex.web itself is not included; at over 1MB in size I decided it was too big to add to the distribution. TEX hackers please note: the latest versions of TANGLE, TFtoPL and PLtoTF are included as MPW tools.

2 Translation of TEX into Modula-2

The translation of the WEB source for TEX 3.0 into Modula-2 for the Macintosh has been redone from scratch. The resulting code is much nicer to look at and far easier to modify than the TEX code used in previous versions of . The main reason for the improvement is that nearly all the WEB numeric macros have been changed to symbolic constants.

The Build-TeX file in the WEBtoMOD disk contains the MPW commands used to translate tex.web into a set of Modula-2 source files. Here is a brief summary of the translation process:

1. Before I could start on TEX I had to get version 4 of TANGLE working. The starting point was tangle.web and an implementation of TANGLE version 3 by Wayne Sewell (much thanks to Walter Carlip for sending me all this stuff). I also had to write PAStoMOD, a Pascal-to-Modula-2 converter. The steps required to create TANGLE provided a useful preparation for the much harder task of translating TEX.

2. A special-purpose change file, tex.ch, was created. This was TANGLEd with tex.web to produce tex.pas.

3. tex.pas was passed through PAStoMOD to create tex.mod.

4. The output of PAStoMOD is a nicely formatted Modula-2 program, but with quite a few syntax errors. It was easier to fix these problems using the pattern matching facilities in MPW, so an MPW script called FixTeX was created to fix tex.mod.

5. tex.mod was then decomposed by SplitTeX into a number of smaller modules suitable for separate compilation. The resulting files were stored in a sub-folder called TeX-sources.

6. After tracking down a few bugs, the final outcome was an MPW tool version of TEX that passed Knuth's trip test.

The resulting code was then merged into and many more changes made. These changes were *not* recorded in MPW scripts, mainly because this is a very boring way of writing code (and I got lazy). It would be easy to reproduce the changes by repeating the above steps and then running compare on all the files in the two different TeX-sources folders.

Until major changes are again made to TEX, I'd advise anybody wanting to modify the TEX part of to forget the WEBtoMOD disk and go straight to the TeX-sources folder on the Sources disk. And get the latest printing of TEX*: The Program*.

3 An overview of the code

is essentially the merging of three programs: TEX, PSPRINT and DVIto VDU. The latter two are public domain programs I wrote while working at Adelaide University and are available for VAX/VMS

and UNIX systems. Both programs come with System Guides that might provide a useful introduction to the inner workings of . There have been significant changes though, most of them improvements!

Here is a brief description of each module in top-down order. I'll concentrate on the main purpose of each module and occasionally mention the most significant routines and data structures:

`OzTeX.MOD`

The main module; note that there is no corresponding `.DEF` file. It handles most of the high-level, Macintosh-specific aspects of 's user interface. The key routine is `MainEventLoop`; this is the best place to start tracing exactly what will do in response to most user-initiated events.

`TeX.DEF/MOD`

's interface to TEX. `OzTeX.MOD` uses just two routines: `RunTeX` and `UnloadTeXSegs`.

`TeX*.DEF/MOD`

All the TEX-related modules. They are used only by `TeX.MOD` and other TEX-related modules. When modifying and testing parts of that have nothing to do with TEX, you can speed up linking considerably by commenting out all references to the TEX-related modules — see the `link` command in `Build-OzTeX`.

`DVItoPS.DEF/MOD`

Used by `OzTeX.MOD` to translate a DVI file into PostScript. The important routines are `TranslateDVIFile` and `DoPage`. Much of the code is based on PSPRINT 3.0.

`DVItoMac.DEF/MOD`

Used by `OzTeX.MOD` to preview a DVI file. The important routines are `ViewDVIFile`, `ProcessPage` and `DisplayPage`. Some of the code is based on DVIto VDU 3.0.

`DVIReader.DEF/MOD`

Used by `DVItoPS.MOD` and `DVItoMac.MOD` to read and interpret DVI files. The definition module exports most of the critical data structures, such as the various lists containing information about all the fonts, characters, rules and `\special`s appearing on a DVI page.

`FontReader.DEF/MOD`

Used by `DVItoPS.MOD` and `DVItoMac.MOD` to read and interpret TFM and PK files.

`PStoPrinter.DEF/MOD`

Used by `DVItoPS.MOD` and `OzTeX.MOD` to send PostScript code to the current printer. Note that the implementation module uses low-level PAP code (rather than Print Manager routines) to communicate with the printer; this gives us complete control over what gets sent to the printer and any messages that come back (such as PostScript errors).

`Options.DEF/MOD`

Reads configuration files and exports various global parameters.

`FileIO.DEF/MOD`

Handles all text file input/output.

`TermIO.DEF/MOD`

Provides with a terminal-like window, mostly for output. The reading routines are only used in a few TEX-related modules.

`LowLevel.DEF/MOD`

The lowest-level module. It exports routines used by nearly all the above modules to handle errors and carry out other miscellaneous tasks. The implementation block is the first piece of code executed and does a number of important things:

- Checks that we are running on a 128K ROM Mac or newer model.
- Checks that the Hierarchical File System is running.
- Installs the VBL task used to detect Command-C/Dot interrupts.
- Installs the `FatalGrowZone` function used to trap out-of-memory errors.
- Initializes a number of global parameters and procedure variables.

4 History of changes

1.0 was released in April, 1989.

4.1 Changes to version 1.0

- Added `Changes` to the Help menu.
- Added support for MultiFinder. This included adding a SIZE −1 resource, adding code to handle Suspend and Resume events in 's main event loop, and changing the implementation of `OpenIn` in `FileIO.MOD` so that a text file currently open by another application (such as an editor) could also be opened by .
- Fixed bug that prevented reading or writing files on other disks. [Reported by Chris Meaney.]
- Fixed bug that caused a spectacular crash if couldn't open the current printer resource file. [Reported by Chris Meaney.]
- Certain fatal errors ("not enough memory" or "disk full") no longer leave files open.
- Added a new "Include Laser Prep" check box option to the print dialogue. This simplifies the inclusion of a modified Laser Prep when printing a DVI file that uses `\specials` to include Mac-generated PostScript files. If this option is selected then appends `LaserPrep.ps` to `DVItoPS.ps`. The option's default setting has been added to `Oz.config`.
- The names of 's special folders and files are now defined in `Oz.config` rather than in STR resources. [Suggested by Leigh Hume.]
- Modified `DVItoPS.MOD` so that the PostScript output generated by starts with `DVItoPS.ps`. This and other PostScript files now begin with "`%!`" as recommended in Appendix C of Adobe's *PostScript Language Reference Manual*. [Suggested by Nick Nei.]
- Added MENU resources to the application so people can use a resource editor to change the Command-key equivalents to suit themselves. Note that I changed the location of Command-P, Command-T and Command-W.
- Shift-Command-W will bypass the view dialogue box and display the most recently viewed DVI page (or page 1 if used first time). [Suggested by Leigh Hume.]
- You can now print or preview an DVI file from the Finder. Use "Print" in the Finder's File menu to print a selected DVI file or "Open" (or double-click) to preview it. [Suggested by Leigh Hume.]
- `SystemTask` is called once every DVI page when typesetting or printing so that DAs get a chance to execute. [Suggested by Leigh Hume.]
- Modified `TEXTtoPS.ps` so that Macintosh characters (like Σ) print correctly.
- Updated the User Guide and completed the System Guide.

1.1 was released in June, 1989.

4.2 Changes to version 1.1

- 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. Users can click in this box to change units.

- will now look for PK file names of the form `:PK-files:300:cmr10.300pk` after first looking for `:PK-files:300:cmr10`. This makes it easy for people to Kermit down their own PK files without the need to do any renaming. [Suggested by Sam Gassel.]
- Some of TEX's capacity parameters are now specified in `Oz.config`.
- PostScript text fonts can now be previewed correctly by specifying the encoding scheme for the corresponding screen fonts in `Oz.config`.
- The location and size of 's windows are now saved upon quitting (and used the next time starts up).
- TEX can now input 8-bit characters. This allows TEX input files to be much more readable for people that don't use English. [Suggested by Jörgen Pind and Martti Nikunen.]
- Nearly all the files in `TeX-inputs` have been updated. The format files in `TeX-formats` have been rebuilt.
- A Times-LATEX format has been provided for people that prefer to use PostScript text fonts instead of Computer Modern. See the `Times-*.*` files in `TeX-inputs`.
- The `@newfont` procedure in `DVItoPS.ps` is now passed the maximum character code in the PK font. People no longer have to edit the `@newfont` code to be able to use PK fonts with more than 128 characters.
- When printing a DVI file, will look for a file called `global.ps` in the current folder and include its contents at the end of the PostScript prologue.
- A number of modified versions of Laser Prep are supplied in `PS-files` in case the default file specified in `Oz.config` doesn't work with the PostScript code generated by a particular Mac.
- Any `\specials` on a page are now processed in the order they appear in the DVI file (instead of in reverse order).
- A Help file is displayed much more quickly and with its top line showing.
- `ErrHandler.DEF/MOD` renamed to `LowLevel.DEF/MOD`.
- After switching back to from MultiFinder we now make sure the cursor is reset to the normal arrow pointer.
- Modified `TEXTtoPS.ps` so that Macintosh characters (like ç) print correctly.
- No longer display the "`Loading format file...`" message when running TEX.
- The application now has FOND and FONT resources for 9 pt Monaco to avoid relying on the resources in the System file (which may have been modified). [Problem reported by Larry Siebenmann.]
- now looks for the printer resource file only in the System folder of the startup disk. It no longer relies on the "poor man's search path". [Problem reported by Bharat Jayaraman.]
- Renamed `fmt` files so they match exactly the case of the names used in `Oz.config`. This avoids problems with a UNIX-based file-server. [Suggested by John Zic.]
- Created `psnames.tex` in `TeX-inputs` for people that use PostScript fonts and want to be able to run the same TEX input file on a variety of systems even if they have different naming

conventions for PostScript TFM files. See `psfonts.tex` in `TeX-docs` for how to use `psnames.tex`.

- Added Bob Campbell's Modula-2 compiler to the package. To squeeze everything into ten 800K disks I used Raymond Lau's excellent StuffIt utility.
- 1.2 is built using MPW 3.0. If you still have MPW 2.x then the only change you'll need to make is to `:Sources:SIZE.r`. See the comments there.

1.2 was released in October, 1989.

4.3 Changes to version 1.2

- Implemented TEX 3.0. (Actually, a couple of bugs have been fixed, so the correct version is probably more like 3.02.) Typesetting is about 5% faster.
- Since TEX 3.0 allows input of 8-bit characters, there is no need for the STR# 201 resource. `8bit.tex` in `TeX-docs` has been updated to use `8bitdefs.tex`, a new file in `TeX-inputs`.
- There is a new Config menu to support rapid switching between multiple configuration files. `Oz.config` has been replaced by `Default` which is kept in the `Configs` folder along with other configuration files.
- To help keep non-default configuration files short and simple, a couple of special characters (`?` and `!`) can be used. See the example files in the `Configs` folder.
- Instead of a single resolution parameter in the configuration file there are now two: one for printing a DVI file and the other for previewing.
- The horizontal and vertical offsets specified in the configuration file no longer affect the positioning of pages when previewing a DVI file.
- Some of the special files defined in the configuration file are no longer assumed to reside in certain folders; their locations must be specified.
- People with colour monitors can now see coloured paper edges, `\special` markers and missing fonts when previewing a DVI file. The colours are specified in the configuration file.
- There is no longer a STR# 200 resource stored in the application. `INITEX` now reads pool strings from a text file; its name and location are specified in the configuration file.
- More of TEX's important parameters are specified in the configuration file. The format files supplied with have all been rebuilt with the parameter settings in the `Default` configuration file.
- To help people choose TEX parameter values that suit their Mac's memory, the amount of memory allocated is now displayed at the end of a TEX session. The elapsed time is also displayed.
- For compatibility with older versions of TEX and LATEX, the following lines have been added to `plain.tex` and `lplain.tex`:

```
\lefthyphenmin=2
\righthyphenmin=3
\errorcontextlines=5
```

- Menus that contain items obtained from the configuration file now appear correct if given a string that contains meta-characters (like /) or is empty (the item will be “NUL”).
- Previous versions of could lose track of the current folder under certain situations. You could run `INITEX` and it would find a given file in the current folder, but if you then ran a DA or switched to another application under Multifinder and ran `INITEX` again it would fail to find the file even though the current folder had not apparently changed. This problem has been fixed.
- Viewing a DVI file on a colour monitor is now faster, especially on a large screen. used to save an off-screen copy of any new view so that a later update could be done rapidly. On a large colour monitor the copying step caused an uncomfortable delay. Since updating occurs less often than changing the view, the copying step is no longer done.
- A silly bug in `DVItoPS.ps` has been fixed (it could cause a `rangecheck` error on non-LaserWriter printers).
- A return is added at the end of an included `\special` file to avoid the possibility of a PostScript error.
- The way checks for out-of-memory errors has been improved. Most `NewPtr` calls are bracketed by code that disables and enables the fatal grow-zone function installed by `LowLevel.MOD`.
- Ignored a pair of inactive/active events after “About ” that caused scroll bars to flash unnecessarily.
- The way searches for TFM and PK files has been extended. See `Changes` in the `Help-files` folder for details.
- A DVI file is now opened for shared reading. This allows two copies of under MultiFinder to open the same DVI file and view side-by-side pages. (A clumsy way of doing it, but better than nothing.)
- The maximum line length in the window now has a minimum value of 256, so the output of “Show Config” is unlikely to be auto-wrapped. When typing input to TEX the cursor will no longer move beyond the current right hand edge of the window.
- *AmS*-TEX is no longer distributed with as the version I have is way out of date.
- James Walker’s `DVIM72-Mac` program is distributed with .
- The Modula-2 compiler is no longer distributed as the author has withdrawn permission. The rights to the compiler have been taken over by Metrowerks. The disk with the Modula-2 compiler has been replaced by the `WEBtoMOD` disk.

1.3 was released in September, 1990.

5 Future development

Here is a list of what I think are the most important areas for future development of :

- Provide a big version of TEX that allows `mem_max` to exceed 64K. The key step is to redefine a halfword to be of type LONGCARD and examine all the implications of such a change. Given Modula-2’s strict type checking, it doesn’t seem possible to make it a run-time option set in the configuration file.
- Add support for virtual fonts (see Knuth’s article in TUGboat 11, no. 1). Most of the changes would be restricted to `DVIReader.MOD` and perhaps `FontReader.MOD`. A list of virtual fonts may need to be specified in the configuration file.

- Support other types of printers, especially the ImageWriter. (Note that James Walker's `DVIM72-Mac` program can't print `\specials`.) The obvious approach would be to create a new module based on `DVItoMac.MOD` and add the necessary Print Manager calls to print page bitmaps. (If PK files at the correct resolution aren't found, would scaled 300 dpi characters look good enough?)
- Add support for inclusion of PICT/PNTG/EPSF files in `\special` commands:

```
\special{pict: foo.pict}    (PICT format)
\special{pntg: foo.paint}   (MacPaint format)
\special{epsf: foo.eps}     (encapsulated PostScript)
\special{ps:   foo.ps}      (raw PostScript)
```

  For upward compatibility with existing `\specials`, "`ps:`" is assumed if no keyword is present (note that Mac file names can't end in a colon). The PICT resource in an encapsulated PostScript file could be used when previewing a DVI file. For PICT/PNTG files, printing could be done by drawing into an off-screen bitmap (or PixelMap at printer resolution?) and then converting to PostScript code using `imagemask` and hex data.
- When translating a DVI file, perhaps 's PostScript output should optionally conform to the structuring conventions defined in Appendix C of Adobe's *PostScript Language Reference Manual*. 's output is currently nonconforming, and deliberately so, as it aims to generate highly efficient, compact PostScript code. A conforming PostScript program must have pages that are independent entities, but sends the bitmap data for a PK character only *once per document*, just before the page that first uses it. For this scheme to work the bitmap data must be sent outside any `save`/`restore` brackets surrounding each page. (This is not done when the "conserve VM" option is selected, so perhaps we should change this option to "generate conforming PS" and use a different prologue with the required structuring comments.)
- Allow previewing of more than one DVI page at a time by (optionally) creating a new view window for each page. Add a "Tile Pages" item to the View menu? Significant changes to the data structures in `DVIReader.DEF` would be needed — perhaps the lists of rules/chars/`\specials` for a page should be moved into the `pagetable`. Can we avoid having to close an open DVI file before printing it?
- Specify in the configuration file the point size of the font to be used for all text in the window. If we also specify the type of font then major changes to `TermIO.MOD` would be needed, since we could no longer assume a fixed-width font.

6 Further reading

PostScript Language Reference Manual, by Adobe Systems Inc.

PostScript Language Program Design, by Adobe Systems Inc.

Macintosh Programmer's Workshop documentation, by Apple Computer Inc.

TEX*: The Program*, by Donald E. Knuth.

A Test file for TEX, by Donald E. Knuth.

DVIto VDU: A TEX *Page Previewer*, by Andrew Trevorrow. TUGboat 7, no. 1.

DVIto VDU 1.7 and PSPRINT 1.1, by Andrew Trevorrow. TUGboat 8, no. 1.

DVIto VDU 3.0 and PSPRINT 3.0, by Peter Abbott & Andrew Trevorrow. TUGboat 9, no. 3.

Programming in Modula-2 (3rd ed.), by Niklaus Wirth.