Installing TeXShop

Richard Koch
Mathematics Department, University of Oregon, Eugene, Oregon, USA
koch@math.uoregon.edu
http://www.uoregon.edu/~koch

Abstract

TeXShop is a TEX previewer for MacOS X, Apple's new Unix-based operating system for the Macintosh. It uses teTeX as an engine, typesetting primarily with pdftex and pdflatex. The pdf output is displayed using Apple's internal pdf display code.

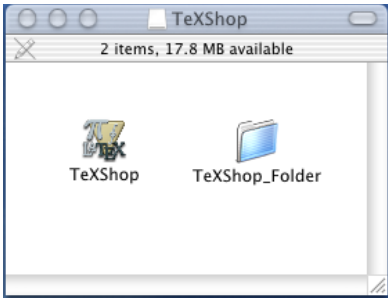
TEX Implementations for MacOS X

MacOS X is Apple's new operating system for the Macintosh. Four features of this operating system interest us. First, the operating system is Unix at the ground level, so all standard TEX programs run. Second, the command line interface is optional throughout the system, so programmers cannot assume that users understand Unix. Third, the system has a wonderful programming environment inherited from NeXT. This environment, OpenStep on the NeXT, is called Cocoa on the Mac. Finally, the graphic system on MacOS X is based squarely on pdf. For example, the Cocoa environment includes classes to display pdf images; low-level graphic primitives on the system correspond directly to pdf commands.

There are several ways to run TEX on MacOS X. The operating system has a mode called "classic" which runs old Macintosh programs; Textures runs under Classic. The X-windows environment runs on MacOS X, so emacs and xdvi can be used. To obtain X, go to http://www.apple.com/x11. Go to http://fink.sourceforge.net for emacs and xdvi. Several native systems run directly on the new operating system, including CMacTex by Tom Kiffe, OzTex by Andrew Trevorrow, and TeXShop by my group. A number of these systems use the same teTeX distribution, which was compiled and configured for MacOS X by Gerben Wierda. The site http://www.esu.psu.edu/mac-tex lists all of these systems, with links to appropriate web sites. See http://www.uoregon.edu/~koch/texshop to obtain TeXShop and teTeX. TeXShop is free.

Installing TeXShop

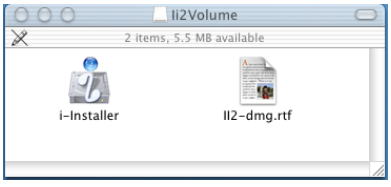
In the TUG conference program, 45 minutes were allotted to installing TeXShop; we'll see if that was enough time. To install, double click on a file named **texshop.dmg** downloaded from my web site.



A window opens containing a TeXShop icon. Drag the TeXShop icon to the Applications folder. Done. TeXShop is installed.

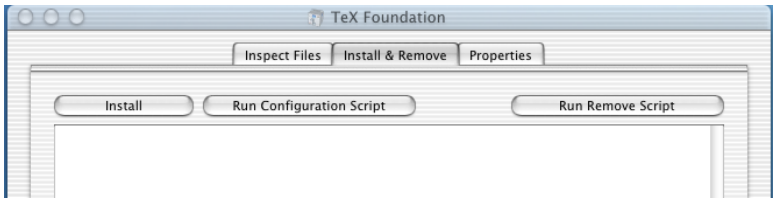
Installing teTeX

We also need to install teTeX. Portions of TeXShop use ghostscript, so we install it at the same time. Double click on the file named **II2.dmg**.



A window opens containing the teTeX installer. Double click on this installer. A panel opens, as shown on the next page.

If we click "Install," the teTeX distribution is installed in /usr/local/teTeX without further input from us. We must also install two other packages; the process takes about five minutes, not counting download time. I refused to install teTeX at the conference because I once watched an Apple Demo



at the University of Oregon in which the demonstrator decided on the spur of the moment to install MacOS-X right before our eyes. Installation was almost complete when the two hour demo ended.

In all, three packages must be installed: TeX Foundation, TeX Programs, and Ghostscript 7. Optional packages are available through i-Installer for graphic conversion, cm-super fonts, etc.

You may have heard that Apple's installer has a bug which sometimes causes it to erase all programs in /Applications while installing new material. For that reason, Gerben Wierda wrote his own installer. You can find details, including full source code for teTeX and for Wierda's installer, on his site at http://www.rna.nl/tex.html.

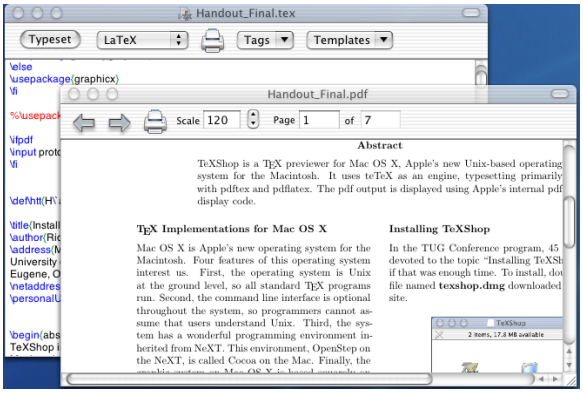
Running TeXShop

Let's see what we've got. To the right, TeXShop is shown running. The window in the background is a source editor; the window on the front shows the resulting typeset document. The editing window has features we certainly expect: parenthesis matching, syntax coloring, tags. The preview window has buttons which increase or decrease magnification and page forward or backward.

Notice the button named "Typeset" at the extreme left of the editing window. Hitting this button saves the document and calls latex (actually pdflatex) to typeset. If the source code has an error, it can be fixed and typeset again without quitting the original TEX job. The popup menu labeled "LaTeX" next to the "Typeset" button changes the behavior of the first button, allowing users to call TEX, LATEX, BIBTEX, MakeIndex, METAPOST, ConTeXt, or METAFONT.

TeXShop can open multiple documents at the same time. Some of these might be TEX documents and others might be LATEX documents. When a document is first opened, the button labeled "LaTeX" will default to a choice set in preferences. If a user usually works with LATEX but happens to open a TEX document, it is easy to switch to TEX for that particular file.

By default, TeXShop calls pdftex and pdflatex to typeset, but pdfetex, pdfelatex and other programs can be chosen in the preferences dialog.



Notice the "Templates" menu. Choosing an item from this menu inserts appropriate text into the document. For instance, I use standard templates to begin LATEX documents and to insert graphic commands. The menu is configurable. Templates are stored in `~/Library/TeXShop/Templates`. When TeXShop starts, the names of files in this directory are listed in the Templates menu; the contents of a file will be inserted when the item is chosen. To modify the menu, simply change the programs in this folder.

Behind the Scenes

In normal operation, TeXShop does not produce dvi files at all. Instead it calls Hàn Thế Thành's wonderful pdftex and pdflatex to directly output pdf files, which are rasterized by Apple's pdf software. The pdf file is also used in printing. MacOS X's printing architecture always prints using a two pass process: in the first pass the printing job is converted to a pdf file, and in the second pass this pdf file is processed for a particular printer. In TeXShop, the first pass has already been done, so the same pdf file is used for displaying to the screen and printing to the printer. TeXShop has no special code for particular printers.

The basic philosophy behind TeXShop is simple: if software already exists to do a job, use that software rather than inventing your own. TeXShop carries this philosophy to extremes — it doesn't do any hard work itself. Editing is done by the

text object written at NeXT, typesetting is done by the software invented by Knuth and many others, and pdf display is done by Apple's pdf rendering routines.

Here's how TeXShop came about. I bought a NeXT when that machine was introduced because I thought it would be fun to demonstrate to others. Wrong. Friends who used Unix pushed the NeXT windows aside and opened a large terminal window. Mac friends said, "I've got that stuff on my Mac and why are the scroll bars on the left?"

But the NeXT was a wonderful working machine, and I came to rely on Tomas Rokicki's TexView, a TEX previewer which made use of the NeXT's Display PostScript. When Apple bought NeXT, I celebrated and began pestering my Apple representative to make sure TexView was ported. The representative said "we understand that TEX is important and we're working on it." But later he said "who's this Tomas Rokicki anyway."

Around that time someone ported teTeX (minus xdvi) to the MacOS X server. Shortly after that I learned about pdftex and pdflatex and discovered that they were present in the teTeX port. What a revelation. Thank you, Hàn Thế Thành!

But would Apple's software display the pdf? According to the html documentation for Cocoa's NSPDFImageRep class, the class had only a few methods, and none to change the page displayed. "Obviously," I thought, "Apple is using a very primitive form of pdf for graphic display." Then on a fluke I looked at the header files and found other undocumented methods. Done!

Constructing TeXShop was easy. Cocoa contains a powerful text editing object. When I wanted to run LATEX, I looked at the html list of classes and found NSTask, an object which runs other Unix processes. Later I wanted to obtain LATEX errors through some sort of pipe, so I glanced through the html list of classes. "Aha. Here's one called NSPipe." It was all about that simple.

During the beta period for MacOS X, Apple's pdf routines refused to recognize embedded fonts, so TeXShop had the following mild restrictions: you couldn't use symbols and your font had to be Times Roman! It was a thrill to run TeXShop on the release version of MacOS X and see TEX fonts for the first time.

The Cocoa classes provide so much functionality that it sometimes seems they work automatically. After using TeXShop for a while, I needed to find something and hit command-f, but no Find panel appeared. "What sort of lousy software is this Cocoa?" I thought. "It didn't even give me a Find panel." TeXShop has one now.

A user wrote asking if we could implement a specific emacs key binding. I replied that TeXShop had no key bindings, but we might add some in the future. "What do you mean?" he wrote back. "Most emacs bindings are there." Sure enough ...

The Team

TeXShop is a free program, released under the GPL. Source code is available on my web site. Shortly after TeXShop was released, Dirk Olmes offered to contribute to the program. At least that's how I remember it; perhaps he really wrote "buster, you need help."

When Olmes began, the program code was in a single source file! Olmes broke the code into pieces, rationalized it, and completely rewrote the preferences system while teaching me about the defaults database on OS X. He is now part of our team. Shortly afterward, Jérôme Laurens contributed the TeXShop icon, which I like a lot. Jérôme's signature reads "Dijon, France, where the mustard grows," so I have a romantic image of his home.

Gerben Wierda has become a more and more crucial player in the team. He carefully crafted a version of teTeX which can be used by Mac folks who don't understand Unix. Several scripts used by TeXShop were written by Wierda. He fixed BIBTEX so it can deal with files with Mac line feeds. I'll have more to say about his teTeX in a moment.

Above all, Wierda constructed the teTeX installer; Unix no longer confronts a user who wants to install TeXShop.

It is interesting that the members of this team come from different countries and speak different languages; I have never met any of them. I don't know how old they are, or how they talk, or where they work. For all I know, one of them is a ten-year old kid.

A great many other users have suggested features which are now in TeXShop. Their names and addresses can be found on my web site under the "Credits" link.

Advanced TeXShop Features

Of course eps illustrations must be converted to pdf form for pdflatex. One hopes that future Macintosh programs will create pdf directly, but that may not happen soon; the MacOS X version of Mathematica offers to "Save Selection" as eps, but not pdf.

A few months ago, Sean Luke suggested that TeXShop should be able to typeset using the sequence latex → dvips → ps2pdf to convert source to dvi, and then to PostScript, and finally to pdf. He sent a shell script to do that. The advantage of this typesetting approach is that old projects with eps illustrations and PostScript special commands typeset directly without modification. TeXShop now includes an option to typeset that way; the option calls an expanded version of Luke's script carefully constructed by Gerben Wierda. The teTeX distribution has been configured to make the resulting pdf file use Type 1 fonts when possibly so it will display rapidly and crisply with Apple's software. Notice that the option requires ghostscript for the final conversion.

The user interface for this option has confused some users. TeXShop is designed so users can have several documents open at once. We imagine that some of these documents will contain pdf illustrations and be typeset by pdflatex, while others will contain eps illustrations and be typeset with TEX and ghostscript. There is a default typesetting command set in preferences; when documents are first opened they are assigned the default typesetting method. But a command in the typesetting menu allows users to change the typesetting method for a particular file after it is open. Users who don't understand this design tend to change the typesetting method in preferences and then discover that the change doesn't affect documents already open.

The latest version of TeXShop has a feature stolen from Rokicki's TexView to simplify choosing a typesetting method. If the first line of a document is `%&pdflatex`, then pdflatex is used. If the first line is `%&latex`, then latex and ghostscript are used. Similar remarks hold for `%&pdftex` and `%&tex`. If no such line exists, TeXShop reverts to the previous method to determine the typesetting engine.

When working with TEX projects, users often need to examine other files briefly. TeXShop can open many such files itself. This includes text files; log files can be opened and inspected, BIBTEX files can be opened and edited, etc. TeXShop can also open `jpg`, `tiff`, and `pdf` illustrations; indeed it can open any `pdf` file. The latest version of TeXShop is also able to open `dvi` files, PostScript files, and `eps` files. Obviously this is done by converting the file to `pdf` and displaying the `pdf`.

In particular, TeXShop can be used to convert `eps` illustrations to `pdf` format, since the process of opening an `eps` also converts it. The conversion indirectly uses epstopdf, but since Macintosh `eps` files sometimes have Macintosh line feed conventions, the conversion script is able to deal with `eps` files regardless of the line feed used. In particular, TeXShop users can now convert `eps` illustrations from Mathematica to `pdf` without first modifying the file.

When TEX is used to write a book, it is common to organize the project using a controlling "root" file. The chapters of the book and their illustrations are put in separate directories inside the project directory. TeXShop can deal with this organization. In the TeXShop "File" menu, there is an item named "Set Root File." This item allows users to assign a root file to any particular TEX source file; when the source file is typeset, the file is saved, but TeXShop typesets and displays the root file. This root file can be in any directory.

I'm not certain this design is optimal because a root file must be set for each separate book chapter. It would be nice to configure the project just once. So the root file syntax may be modified at some later time.

Configuring teTeX

Although teTeX is the engine underlying TeXShop, users may choose to ignore it completely; teTeX will do the hard work silently.

In MacOS X, Unix directories are hidden from the user. In particular, the directory where teTeX is installed, `/usr/local/teTeX`, is not visible from the finder. Gerben Wierda automatically adds a symbolic link to this directory to the system Library folder, which is visible. To inspect teTeX in the Finder, go to `/Library/teTeX`.

The teTeX documentation can be found in `/Library/teTeX/share/texmf/doc/tetex`. In particular, see the files `TETEXDOC.pdf` and `teTeX-FAQ`. More documentation for Wierda's MacOS X compilation is in `/Library/teTeX/README.macosx`.

It is possible to store additional style files, fonts, etc., in teTeX itself. But that is not a good idea; it is better to keep the teTeX directory pure so it can be upgraded when later versions are released. Instead, users should construct a mirror image of the teTeX directory structure inside their `~/Library` folder and store personal files there.

Gerben Wierda has configured teTeX to look for TEX files in four locations, in the following order:

- `~/Library/texmf`
- `/usr/local/teTeX/share/texmf.local`
- `/usr/local/teTeX/share/texmf.macosx`
- `/usr/local/teTeX/share/texmf`

The first of these spots is where individual users will put extra files. The second is where the system

administrator will put files for all users on the computer. The third is where Gerben Wierda will put his own configuration files for teTeX. The final spot is the original texmf tree maintained by the TEX community and kept as default as possible.

When new upgrades to teTeX are introduced, the installer will overwrite files in the last two directories. But files in the other directories will be left alone.

The teTeX system uses a directory structure invented by the TUG working group. When users want to add files to the system, they need to inspect `/Library/teTeX/share/texmf` to find a likely spot for the file, and create a mirror image directory in `~/Library/texmf`.

I'll show how this works by answering three common email questions about TeXShop. As you'll see, we are only partway along the path leading to Mac-like ways to administer TEX. The answer to the first question below is satisfactory, but the remaining answers require more Unix than we'd like.

How Do I Install REVTEX?

At the time of the conference, REVTEXwas not included in Gerben Wierda's distribution. It is now, but I'll still use it to explain how to install extra packages in the system. I'll simulate a typical installation, including an error which must be debugged.

Visit http://publish.aps.org/revtex4/. At the bottom of this page our package is available as `revtex4.tar.gz`. Unpack this package.

The folder contains seven input files:

```
10pt.rtx 11pt.rtx 12pt.rtx aps.rtx
revsymb.sty revtex4.cls rmp.rtx
```

It also contains two BIBTEX files: `apsrev.bst` and `apsrmp.bst`. We look inside the teTeX directories and find that most LATEX include files are in folders inside `tetex/tex/latex`. Therefore, we create the folder `~/Library/texmf/tex/latex/revtex` and place all nine REVTEX files in that folder.

Next we try REVTEX on some sample files. Everything works until we use BIBTEX; the system complains that it cannot find apsrev.bst. We conclude that BIBTEX include files belong in a different directory.

Looking again at teTeX's tetex directory, we find a subdirectory named `bibtex`. Inside this directory is a directory named `bst`. Probably BIBTEX include files go into this directory. So we create `~/Library/texmf/bibtex/bst/` and place the two `bst` files from REVTEX inside. Then everything works.

How Do I Typeset TEX Files Created by Mathematica?

Mathematica can save notebooks in TEX format. This TEX source requires special include files and fonts, which we must install. The required files are packaged with Mathematica. The instructions which follow refer to the MacOS X version of Mathematica. Click on the Mathematica icon while holding down the control key. A dialog allows us to open Mathematica as a folder.

Inside, we find many files, including TEX files. Go to `SystemFiles/IncludeFiles/TeX` and copy files as follows:

- `texmf/tex/latex/wolfram`
  → `~/Library/texmf/tex/latex/wolfram`
- `texmf/fonts/tfm/wolfram`
  → `~/Library/texmf/fonts/tfm/wolfram`
- `texmf/fonts/vf/wolfram`
  → `~/Library/texmf/fonts/vf/wolfram`
- `texmf/dvips/init/wolfram.map`
  → `~/Library/texmf/dvips/config/`
- `/SystemFiles/Fonts/Type1`
  → `~/Library/texmf/fonts/type1/wolfram`

So far the instructions are Mac-like, if somewhat complicated. Unfortunately, Wolfram supplies type 1 fonts in pfa format, and teTeX requires fonts in pfb format. So we must convert each Wolfram font from one format to the other. The program required to do this is called t1binary; luckily it exists in teTeX. To convert, say, Mathematica1.pfa to Mathematica1.pfb, type the following command

```
t1binary --output=Mathematica1.pfb
   Mathematica1.pfa
```

Convert all forty-six fonts in this way.

When Mathematica outputs notebooks in TEX format, it writes graphic content as a series of eps files. Hence the notebook should be typeset using TeXShop's "TeX and Ghostscript" option.

How Do I Use the Lucida Bright and MathTime Fonts?

These are commercial fonts, so teTeX does not contain them. But teTeX has all required supporting files, so it is only necessary to obtain the Type 1 fonts themselves and add them to the system.

Many users already own these fonts for earlier Mac operating systems. I'll describe how to use them for users who bought the fonts from Blue Sky Research for Textures. There is just one problem. The Textures fonts are in a Mac format used by Adobe Type Manager. But teTeX uses the pfb format required in the Windows world and elsewhere.

LucidBri	lbr	LucidSan	lsr	LucidSanTyp	lstr
LucidBriDem	lbd	LucidSanDem	lsd	LucidSanTypBol	lstb
LucidBriDemIta	lbdi	LucidSanDemIta	lsdi	LucidSanTypBolObl	lsbo
LucidBriIta	lbi	LucidSanIta	lsi	LucidSanTypObl	lsto

LucidFax	lfr	LucidNewMatAltIta	lbmo	LucidCalIta	lbc
LucidFaxDem	lfd	LucidNewMatArr	lbma	LucidHanIta	lbh
LucidFaxDemIta	lfdi	LucidNewMatExt	lbme	LucidBla	lbl
LucidFaxIta	lfi	LucidNewMatIta	lbmi	LucidBriObl	lbsl
		LucidNewMatSym	lbms		

Luckily, free conversion software is available on the internet. Running this software is the unpleasant part.

The software needed is called t1unmac. At the conference I had to explain how to retrieve this software and compile it. Luckily, the software is now part of Gerben Wierda's teTeX distribution.

The t1unmac Unix program which will do the conversion. But Unix does not understand the resource fork used by Macintosh fonts, so we must first pack these fonts into a flat file using the old Mac utility BinHex4. As an example, let us convert the Lucida fonts. Run classic, run BinHex4, and use the menu Application → Upload to convert LucidBri to LucidBri.Hqx. Convert the other font files similarly.

Move these files to your home directory. Open Terminal and execute the command

```
t1unmac --binhex --pfb
  --output LucidBri.pfb LucidBri.Hqx
```

to convert the first font to pfb format. Convert the other files the same way.

The pfb files must now renamed; for example, LucidBri.pfb to lbr.pfb (see the table at the top of the page). Place the renamed pfb files in ~/Library/tetex/fonts/type1/yandy/Lucida. To use these fonts instead of the cm fonts, add the following line to your LaTeX document:

```
\usepackage{lucidbry}
```

Coexisting with Fink

MacOS X uses its own graphic system rather than the X Window System. But X-windows can run on MacOS X. When it does, the user has two views of the same file system and can switch between these views with a keystroke.

It is still necessary to obtain and compile X-windows software like ghostview, xdvi, and xpdf. A remarkable system called Fink has been invented to ease this task. Suppose Fink has been installed and suppose you want to obtain ghostview. Type "fink install gv" in the Terminal. Fink will consult its database and discover that gv requires ghostscript and ghostscript-fonts. It will ask permission to install these as well, and will then retrieve the source files, retrieve patches for MacOS X, compile the code, and install it, all automatically.

To avoid overwriting files, Fink installs software in its own location, /sw/bin. So far, so good. But when Fink is setup, it modifies the PATH variable to search /sw/bin first. Thus Fink versions of software will be used even if other versions are also installed.

At the time of the conference, this caused problems. Most of these problems have since been fixed, but I'll discuss them briefly for amusement.

Originally TeXShop used ghostscript 6 rather than ghostscript 7 because Apple's pdf rendering software could not interpret TeX fonts in pdf files created by ps2pdf in ghostscript 7. So it was important that TeXShop call the version of ps2pdf in /usr/local/bin rather than the Fink version in /sw/bin. Unhappily, ps2pdf is a script which ultimately calls ghostscript, so even if we called the correct version, the wrong version of ghostscript would run.

Just before the conference, we released new versions of TeXShop and teTeX which were inoculated against Fink. TeXShop now calls scripts which temporarily reset the PATH variable before calling ps2pdf. Even if users installed the fink version of ghostscript, TeXShop will run with no problems. We also modified TeXShop so it will use Gerben Wierda's distribution even if a second version of teTeX is installed.

Later Apple fixed the pdf rendering bug, and our system now uses ghostscript 7.

Since the conference, the Fink distribution of teTeX was modified. The fink installer now asks if it should install its own version or make symbolic links to a version already installed in /usr/local/bin. Users should choose the "symbolic link" option.

Those who install the fink version of teTeX are asking for trouble, because any TeX program called from the command line will use the Fink version,

which is not configured to look for extra files in `~/Library`. Thus, include files will mysteriously be found sometimes and not at other times. There is little reason to install the Fink version of teTeX because X-windows software should work fine with Gerben Wierda's distribution. Certainly that is true of the xdvi installed by Fink.

Missing Features

If you are using TeXShop, please report bugs and request extra features. Our email addresses are listed at the end of this article.

Four requests were often made at the time of the conference. We have responded to some of these requests since then. Here are the requests:

Better spell checking: Recent versions of TeXShop set the Macintosh type of files to TEXT. Therefore, these files can be opened with Excalibur, a wonderful LaTeX spell-checker. The spell checker cocoaAspell by Anton Leuski was introduced after the conference and also works with TeXShop.

Magnifying glass in the preview window: This was introduced in version 1.27, using code provided by Mitsuhiro Shishikura. Shishikura added other important features in 1.27.

Edit using other editors: Some users want to edit their TeX files with BBEdit or xemacs. This can be done with versions introduced after the conference.

Coordinate the source and pdf windows: Textures has a feature called Synchronicity. We are often asked to provide it for TeXShop. But the Apple pdf classes have no methods for searching pdf files or clicking in the image and finding the corresponding spot in the file. So providing this feature would be hard work.

Samuel Eilenberg was a great topologist. One day a student said to him "Professor Eilenberg, I have decided to work in algebraic topology." Eilenberg replied "Young man, don't let me stand in your way."

Similarly, the TeXShop source code is available at my web site. "Young man, . . . "

Email Addresses

Richard Koch koch@math.uoregon.edu
Dirk Olmes dirk@xanthippe.ping.de
Gerben Wierda Gerben_Wierda@rna.nl