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.

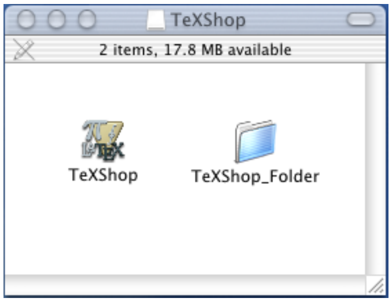
TEX Implementations for Mac OS X

Mac OS 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 Mac OS 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.

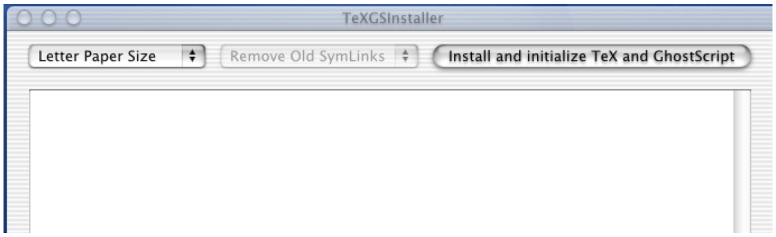
There are several ways to run TEX on Mac OS X. The operating system has a mode called "classic" which runs old Macintosh programs; Textures, OzTeX and CMacTex run under Classic. The XFree86 X-windows environment runs on Mac OS X, so x-emacs and xdvi can be used. To obtain xfree86, go to http://www.xfree86.org. To obtain xdvi and xemacs, go to http://fink.sourceforge.net. Finally, there are two native programs running directly on the new operating system, CMacTex by Tom Kiffe and TeXShop by folks listed below. Both of these systems use the same teTeX distribution, which was compiled and configured for Mac OS X by Gerben Wierda. CMacTex can be obtained at http://www.kiffe.com/tex/cmactex.html and both TeXShop and Wierda's teTeX can be obtained at http://www.uoregon.edu/$\sim$koch/texshop.
CMacTeX contains several programs: MacGhost-View, macdvi, Macps2pdf, Metapost, and others. TeXShop is a single program. TeXShop is free.

Installing TeXShop

In the TUG Conference program, 45 minutes were devoted 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.

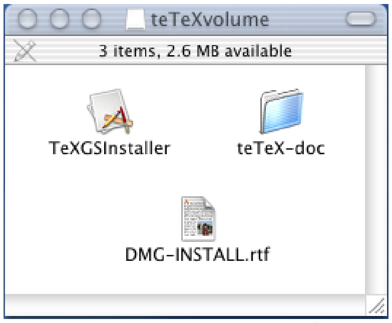


A window opens containing a TeXShop icon. Drag the TeXShop icon to the Applications folder. Done. TeXShop is installed.



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 **teTeX.dmg**.



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

If we click "Install and initialize TeX" the entire distribution is installed in /usr/local/teTeX without further input from us. This takes about two minutes. 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 Mac OS X right before our eyes. Installation was almost complete when the two hour demo ended.

When teTeX was first released by us, symbolic links were created in /usr/local/bin to the TEX binaries. Later Wierda decided to add the location of these binaries to the PATH variable, so an optional install menu item asks users if they wish to remove the old symbolic links. The installer also asks whether dvips should be configured for Letter Paper Size or A4 Paper Size. When installation of teTeX is complete, we also install Ghostscript.

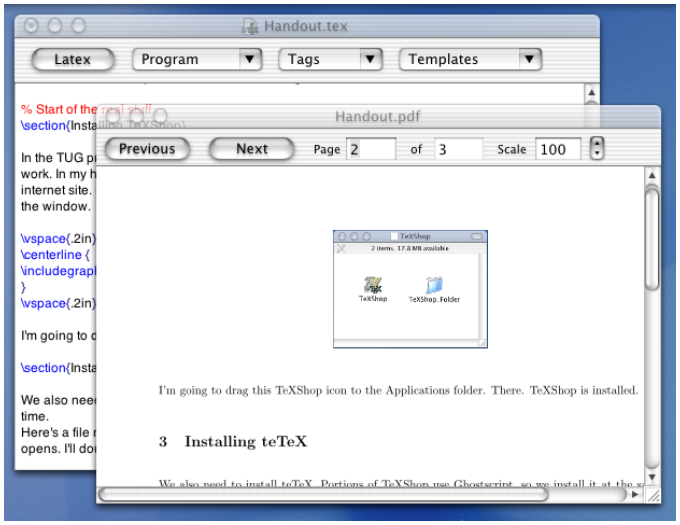
You may have heard that Apple's installer has a bug which sometimes causes it to erase all programs in /Applications while installing new material. For that reason, Gerben Wierda wrote his own installer. You can find the full source code for teTeX, and for Wierda's installer, on his site at ftp://ftp.nluug.nl/pub/comp/macosx/tex-gs/.

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 "Latex" 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 and typeset again without quitting the original TEX job. The pulldown menu labeled "Program" next to the "Latex" button changes the behavior and title of the first button, allowing users to call Tex, Latex, Bibtex, or MakeIndex.

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 on the left 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.



```
% Start of the real stuff
\section{Installing TeXShop}

In the TUG p
work. In my h
internet site.
the window.

\vspace{.2in}
\centerline {
\includegrap
}
\vspace{.2in}

I'm going to d

\section{Insta

We also nee
time.
Here's a file 
opens. I'll do
```

I'm going to drag this TeXShop icon to the Applications folder. There. TeXShop is installed.

3 Installing teTeX

We also need to install teTeX. Portions of TeXShop use Ghostscript, so we install it at the s

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 template 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. Mac OS 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 Tex-View, a TeX previewer which made use of 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 Mac OS 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 Mac OS 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 Mac OS 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."

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 joined us; among other things, he made 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.

Other users have suggested features which are now in TeXShop. Rudolf Blazek from USC has just constructed a version of TeXShop which typesets continuously; his version will soon be on my web site.

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 Mac OS 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. At some later time, the root file syntax may be modified.

Configuring teTeX

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

In Mac OS 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, examine the files TETEXDOC.pdf and teTeX-FAQ. Additional documentation for Gerben Wierda's Mac OS X compilation is in `/Library/teTeX/` in the file 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.

LucidBri	lbr	LucidSan	lsr	LucidSanTyp	lstr	LucidFax	lfr
LucidBriDem	lbd	LucidSanDem	lsd	LucidSanTypBol	lstb	LucidFaxDem	lfd
LucidBriDemIta	lbdi	LucidSanDemIta	lsdi	LucidSanTypBolObl	lsbo	LucidFaxDemIta	lfdi
LucidBriIta	lbi	LucidSanIta	lsi	LucidSanTypObl	lsto	LucidFaxIta	lfi

LucidNewMatAltIta	lbmo	LucidCalIta	lbc
LucidNewMatArr	lbma	LucidHanIta	lbh
LucidNewMatExt	lbme	LucidBla	lbl
LucidNewMatIta	lbmi	LucidBriObl	lbsl
LucidNewMatSym	lbms		

The teTeX system uses a directory structure invented by the tex 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?

Rather than giving mechanical instructions, I'll simulate a typical installation of a package, including an error which must be debugged.

Go to `http://publish.aps.org/revtex4/`. At the bottom of this page our package is available as revtex4.tar.gz. OmniWeb automatically unpacks this file into a folder named revtex4. Internet Explorer downloads; then we must unpack by hand.

The folder contains seven input files: 10pt.rtx, 11pt.rtx, 12pt.rtx, aps.rtx, revsymb.sty, revtex4.cls, and 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.

By the way, the version of teTeX on the web site in August is slightly misconfigured, so texhash must be run after adding files to `~/Library/texmf`. This will be fixed as soon as possible.

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 Mac OS 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`. Copy

```
texmf/tex/latex/wolfram
```

to

```
~/Library/texmf/tex/latex/wolfram
```

and copy

```
texmf/fonts/tfm/wolfram
```

to

```
~/Library/texmf/fonts/tfm/wolfram
```

and copy

```
texmf/fonts/vf/wolfram
```

to

```
~/Library/texmf/fonts/vf/wolfram
```

and copy the file

```
texmf/dvips/init/wolfram.map
```

to

```
~/Library/texmf/dvips/config/
```

Finally copy `/SystemFiles/Fonts/Type1` to

```
~/Library/texmf/fonts/type1/wolfram
```

So far the instructions are Mac-like, if somewhat complicated. But now we need to tell teTeX about this map file. Before I knew better, I told users to go to `texmf/dvips/config` and edit and run updmap. But Gerben Wierda complained that this violated his rule about keeping the original texmf tree pure. The currently approved rule works differently for pdflatex and for tex and ghostscript.

To make latex and ghostscript work, go to preferences and add the following text to the “altpdflatex” entry:

```
--extradviops "-u +wolfram.map"
```

To make pdflatex work, find the file “pdftex.cfg” in `/usr/local/teTeX/share/texmf.local/pdftex/config/`
and add the following line to the end of the file:

```
add +wolfram.map
```

Sadly, root permission is needed to edit this file.

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

Go to http://ftp.uci.agh.edu.pl/. In the directory `pub/tex/fonts/utilities/t1utils` find and get the file t1utils-1.22.tar.gz. Download this file with OmniWeb to unpack automatically, or unpack by hand. Open the Terminal application and type the following commands

- cd
- cd t1utils-1.122
- ./configure
- make
- sudo tcsh
- make install

You now have Unix files 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 should be 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

Mac OS X uses its own graphic system rather than X-windows. But X-windows can run on Mac OS 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 Mac OS 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.

This causes two problems for TeXShop. The first is, we hope, temporary. TeXShop uses ghost-script 6, rather than ghostscript 7, because Apple’s pdf rendering software cannot interpret TEX fonts in pdf files created by ps2pdf in ghostscript 7. Apple knows about this problem, but we do not know when it will be fixed. So it is 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 call the correct version, the wrong version of ghostscript will run.

Just before this conference, we released new versions of TeXShop and teTeX which are inoculated against Fink. TeXShop now calls scripts which temporarily reset the PATH variable before calling ps2pdf. If you install the fink version of ghostscript, TeXShop will run with no problems. It is important to be aware that calling ps2pdf from the Terminal may create pdf files which Apple software will not render correctly.

Fink is happy to install xdvi without installing teTeX. But if a user asks to install lyx, then fink insists on installing teTeX as well. Users who do this will have two copies of teTeX. The versions of TeXShop and teTeX released just before the conference have been protected against this intrusion as well. TeXShop will call Gerben Wierda’s distribution even if these is a second distribution around.

However, users 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 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 are often made:

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.

It is possible to obtain tighter integration with Excalibur, but there is a purely psychological barrier that stands in the way. Excalibur is a Carbon application and TeXShop is a Cocoa application. In the Cocoa world, spell checkers work automatically without special code in the receiving application because NeXT and Apple provide a special Cocoa class for spell checkers. If a TEX spell checker were written in Cocoa, then all the work could be done by the author of the spell checker. Given the dream of a Cocoa spell checker, it is difficult to work up the enthusiasm to add code for Excalibur.

Magnifying glass for the preview window: We expect to add this soon.

Edit using other editors: Some users want to edit their TEX files with BBEdit or xemacs. The difficulty here is trying to find the correct TeXShop design to allow this without having to write special code for each editor. We have a few ideas, but suggested designs from others are welcome.

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; Jérôme Laurens jerome.laurens@u-bourgogne.fr;
Dirk Olmes dirk@xanthippe.ping.de; Gerben Wierda Gerben_Wierda@rna.nl.