

## About TeXShop 2.15 - 2.18

### Changes in TeXShop 2.15

TeXShop 2.15 was an experimental release. It lived for a long time on my personal web page with a promise to migrate it to the usual TeXShop site. After the promise didn't materialize for several months, the link on my personal site was noticed by Version Tracker, and for several months that system pointed to the experimental 2.15 as the latest release. At last 2.15 has become "official" with the release of TeXShop 2.18.

Here is a list of new features:

- (1) Fixed severe memory leakage on Leopard; said another way, the hidden preference item `ReleaseDocumentClasses` is now automatically set correctly on Leopard. This very significant bug, which I listed for a long time as the most significant bug in TeXShop, was caused by an Apple bug in PDFKit. This bug is fixed in Leopard. Consequently, users must be running System 10.5 to obtain the fix.
- (2) Fixed pdf drag and drop on Leopard. In earlier versions on Leopard, a dragged pdf selection had a white background, making the feature problematic for Keynote and other programs. Now a dragged pdf selection has a transparent background, as it always did on Tiger.
- (3) Support for the Hiero package, used to typeset hieroglyphics. See <http://www.filipvervoese.be/hierotexmac/> and the detailed explanation on <http://www.filipvervoese.be/hierotexmac/hierotex.html>.
- (4) Support for Sketch, used to construct 3D images in pdf files. See [www.frontiernet.net/~eugene.ressler/](http://www.frontiernet.net/~eugene.ressler/).
- (5) Generalizing the last two items, support for adding additional extensions to the list of filetypes which can be processed via TeXShop engines.
- (6) The default templates now include a template for Beamer taken directly from the beamer web page. To see it, move the folder `/Library/TeXShop/Templates` to the desktop and restart TeXShop. The default Templates folder will be created. Merge your old Templates folder in with the new one.

Here are more details on each of these items.

- (1) When TeXShop was first released on Tiger, users ran into an annoying bug which caused the program to gradually slow to a crawl after several typesetting actions. The problem occurred when a new pdf file was loaded into PdfKitView for the Preview window. According to Apple documentation, this should have released the previous data from memory. The release did occur, but it caused the program slowdown.

To fix this problem, I added code to TeXShop which tricked the system into believing that the old data structures were still being used so the system didn't try to release them. This meant that TeXShop gradually used more and more memory over time, and it caused other problems as well. In notes to collaborators, I called this "the single most important bug in the program."

This was really a PDFKit bug. But although I have reported several bugs to Apple (and they have been very good about fixing them), I didn't report this problem because I needed to make a small demo program illustrated the bug, and never got around to it.

When system 10.4.3 was released, it looked to me like the problem was resolved, and I modified the TeXShop code to release memory on 10.4.3 and higher. Unfortunately, it soon became apparent that the problem remained, particularly for large pdf files. Luckily, I had added a hidden preference to TeXShop called "ReleaseDocumentClasses"; the value of this preference could be

- 0 to release the data on system 10.4.3 or higher
- 1 to never release the data
- 2 to always release the data

Initially the default value of this preference was 0, but I soon changed it to 1, restoring the old "kludge behavior."

But when Leopard came out, several users reported that this preference can safely be set to 2. Further testing showed that the PDFKit bug was fixed in Leopard. Therefore, in TeXShop 2.15 the data is always released on Leopard, regardless of the value of ReleaseDocumentClasses. The old behavior still applies on system 10.4.11 and earlier.

But just in case, there is another hidden preference called ReleaseDocumentOnLeopard. The default value of this preference is YES. If it is set to NO, the old preference ReleaseDocumentClasses becomes active and behaves as before.

- (2) TeXShop has the ability to select a portion of the pdf image, copy the selection, and paste it into another program. Users can also drag the selection directly to another program. Before the program LaTeXiT was introduced, this feature was often used, but now LaTeXiT is usually the better way to proceed.

The file format of the copy is controlled by TeXShop preferences; the default value is to copy as pdf with a transparent background, making it easy to use the result in Keynote and similar programs.

This feature broke in the beta version of Leopard which Apple released at the 2007 Developer Conference. Later I managed to modify my code and fix the problem. But in the release version of Leopard, my fix also broke.

At the developer conference I spoke to the author of PDFKit, who recommended a different fix. That fix is now in TeXShop 2.15.

The old code used the `NSView` method `dataWithPDFInsideRect` directly in the PDFKit View. Before calling this method, it set the background color of the image in PDFKit to be transparent, and it also modified the PDFKit `drawPage` method to skip drawing a background when drawing for a selection. However, PDFKit in Leopard seems to have additional drawing layers which make the individual pages of an image stand out, and these layers add their own backgrounds.

The new method uses PDFKit's page object and the routine `dataRepresentation`, which I learned at the developer conference does not include background information. This data is then placed in a `NSPDFImageRep` object, imaged in an offscreen `NSView` object, and captured with the object's `dataWithPDFInsideRect` method.

There is a slight change when copying and dragging selections. Earlier, a selection could span more than one page. Now the copy will only include the portion of the selection which is on the page under the cursor.

By the way, this fixes the last TeXShop Leopard bug known to me.

- (3) TeXShop now contains an engine file in `.../Library/TeXShop/Engines/Inactive` for use with the Hiero package. An additional README in the folder explains how to install Hiero. The Hiero `.htx` extension is now recognized by TeXShop, and an icon for such files is provided.

To obtain this engine, it is necessary to move the folder `/Library/TeXShop/Engines` elsewhere, say to the desktop. Then restart TeXShop. The program will create a new Engines folder, containing the new inactive items. Then merge the Engines folder on the desktop into this new default Engines folder.

- (4) The same directory contains an engine file for Sketch. An additional README in the folder explains how to install Sketch. The `.sk` extension is now recognized by TeXShop.
- (5) The previous two items are special cases of a general problem. A number of users have tried to use TeXShop's engine mechanism to process a source file and preview the resulting pdf output in cases where the processing engine isn't TeX at all, but rather some other Unix binary program. In many of these cases the source file has a new extension; for instance, `.htx` in the Hiero case, and `.sk` in the Sketch case.

There are two problems with this technique, one minor and one major. The minor problem is that when TeXShop creates a file, it always adds an appropriate extension, usually `.tex`. In the Save dialog there is a pulldown menu listing all extensions known to TeXShop. By using this menu, files can be created with other extensions like `.ltx`, `.ctx`, and so forth.

But if an extension is not in this list, creating it within TeXShop is tricky. Users often try to directly type an extension, saving for example a file with name “myfile.htx”. But actually TeXShop will then create “myfile.htx.tex” and even worse, the Finder may then hide the “.tex” extension.

Luckily, there is a solution. One of the file types which TeXShop can save is named “Plain Text Document”. Such a file has no extension. So if the user saves “myfile.htx” after selecting the “Plain Text Document” dropdown menu item, they actually will get “myfile.htx”.

The good news is that when TeXShop opens a file with an unexpected extension, say by dragging the file to the TeXShop icon, it will preserve the correct extension when saving. So this first problem is a minor problem during file creation, but it doesn’t interfere with later processing the file.

The major problem is that TeXShop deactivates the “Typeset” button when a file is opened with an unknown extension, or with an extension which is not used by source files. For example, TeXShop can open pdf files and jpg files, but it doesn’t allow the user to typeset such files! Users who wanted to process “.htx” and “.sk” files with an engine found that they could not use the engine because of this behavior.

TeXShop 2.15 has a new mechanism for such users. A hidden preference allows users to add extensions to the list of legal extensions which activate the Typeset button. For example, the command

```
defaults TeXShop OtherTeXExtensions -array-add "htx"
```

adds “.htx” to the list. To clear the list and start over with the built-in defaults, use

```
defaults TeXShop OtherTeXExtensions -array
```

Actually, “.htx” and “.sk” are now built-in, so these users do not need the new mechanism. But it is available for future users.

- (6) The Beamer template can certainly be improved. I’d appreciate help from experts.

### Changes in TeXShop 2.16 through 2.18

Versions 2.16 and 2.17 of TeXShop were constructed for test versions of MacTeX-2008, and released only to a few people testing that install package. Version 2.18 is now officially released on the TeXShop site. Here are the changes:

- (1) Added a new Preference item under the Typesetting tab. The preference is called “After Typesetting” and the choices are “Bring Preview Forward” and “Continue Editing.” The first item activates the Preview window after typesetting and brings it forward, as TeXShop has behaved in the past. When the “Continue Editing” item is activated, typesetting updates the contents of the Preview window but leaves

the Edit window active with the editing cursor blinking. This makes it possible to typeset, glance over to check that things look right, and continue editing. This option is particularly useful on large monitors which can show the full contents of both windows.

- (2) Added support for SyncTeX. SyncTeX is new synchronization technology by Jerome Laurens which is included in TeX Live 2008 (but not earlier TeX distributions). The technology is built directly into pdfTeX and XeTeX. In modern TeX distributions, most typesetting actually calls pdftex (even if using latex to output a dvi file), and thus the technology is available for such typesetting.

To use the technology, add the flag

```
--synctex=1
```

to the command line call to TeX. For example, in TeXShop Preferences under the Engine tab, the lines specifying pdfTeX for Tex and pdfLaTeX should read

```
pdftex --shell-escape --synctex=1
pdflatex --shell-escape --synctex=1
```

A similar flag should be added to other engine scripts. The TeXShop defaults for these preferences already include this flag for pdftex and pdflatex, but not for other engine files. If the flag is included and an older version of TeX is called, typesetting will still occur (TeX will report to the console that it does not understand the flag.)

The flag causes TeX to output an additional “synctex” file during typesetting, containing information linking the TeX source file(s) to the TeX pdf file. This file is similar to the old pdfsync file generated by the older PdfSync technology, but with the very significant difference that line and page breaks are no longer changed when outputting the data.

Laurens also wrote a command line program named “synctex” which is included in TeX Live 2008; when this program is called with a request for appropriate synchronization data, the program parses the synctex file and outputs appropriate data.

To activate SyncTeX support in TeXShop, go to TeXShop Preferences under the Misc tab and select “SyncTeX” as the “Sync Method.” This is the default value if you are installing TeXShop for the first time.

If SyncTeX synchronization is chosen, user interaction is exactly the same as in the old Search method. Hold down the command key while clicking at a spot in the source document. The Preview window will become active and the corresponding spot will be circled in red. Or hold down the command key while clicking at a spot in the Preview window. The source window will become active and the corresponding TeX input commands will be highlighted in yellow.

When these commands are used, TeXShop will fall back on the old Search method if SyncTeX does not find an appropriate synchronization. The most common cause for SyncTeX failure is the absence of a synctex file, which will certainly happen when the file is typeset with an older distribution. Thus users can switch between TeX Live 2007 and TeX Live 2008 without changing their synchronization preference.

A few users might like to test TeXShop's SyncTeX support without being confused by calls to the old Search synchronization method. To simplify this test, there is a new hidden preference which forces synchronization to use only SyncTeX:

defaults write TeXShop SyncTeXOnly YES

Support for SyncTeX was added to TeXShop 2.18 at the last moment. Synctex is now used in both directions, and is generally much better than the old search synchronization. But many improvements should be possible as I digest what Jerome has provided.

Many thanks to Jerome Laurens for this wonderful work. I think you will notice an immediate improvement.

- (3) Added encodings ISOLatinGreek, MacCentralEurRoman, and WindowsLatin2 (i.e., WindowsCentralEurRoman).

The names of the encodings are

Greek (ISO 8859-7) Mac Central European Roman Central European (ISO Latin 2) Central European Roman (Windows)

and the

IsoLatinGreek Mac Central European Roman ISOLatin2 WindowsCentralEurRoman

- (4) In  
allow a space between
- (5) Added code by Adam Maxwell to permit BibDesk Completions; this is controlled by a new Preference Setting. By default, it is not activated.
- (6) Fixed French nib file so Typeset menu works again (there are no longer simultaneous checks on "PdfTeX" and "TeX & Ghostscript").
- (7) Fixed tab settings, so now changing tab spacing in Preferences works correctly.
- (8) Command-T now typesets even when the console window is active.
- (9) Fixed pdf updating. If a user has an external editor and uses TeXShop to display pdf output, and if the "Automatic Preview Update" preference is set, then whenever the pdf file is updated, TeXShop reads and displays the latest version, activating the pdf window when it does so. There is a hidden preference item

named "BringPdfFrontOnAutomaticUpdate". If this is set to "NO", then the preview window is supposed to be updated but not brought forward. Recently, the preview window did this, but didn't display the new content until the user manually activated it. This is fixed.

- (10) Instead of separate commands to display each of George Gratzler's symbol tables, there is one command to display all of the tables in one document. It is much easier to find a symbol if you can scroll through tables rather than guessing which table has the desired symbol.
- (11) Modified default LaTeX macros to bring them up to date.