

About TeXShop 2.15

TeXShop 2.15 is an experimental release. It will migrate to the usual TeXShop site in a week or two. Many requested TeXShop features aren't yet in this version; I have a folder of requests from users and have just begun incorporating them into the program. So if a promised feature is not present, give me a few weeks.

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.
- (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.filipvervloesem.be/hierotexmac/> and the detailed explanation on <http://www.filipvervloesem.be/hierotexmac/hierotex.html>.
- (4) Support for Sketch, used to construct 3D images in pdf files. See 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."

I suspect that 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 now in Leopard, several users have reported that this preference can safely be set to 2. Some of these users have very large projects, so I believe that the problem is 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.