A Fix for the Corrupt Font Cache Bug

The Announcement

The "corrupt font cache bug" on OS X has been traced by Melissa O'Neill to a bug in pdftex, present in that program since 2004. The incorrect code is also used in other programs, notably dvips, luatex, and metapost.

Following the initial discovery, intense debugging by Melissa O'Neill, Thanh The Han, and Jin-Hwan Cho, with the help of Jonathan Kew and Karl Berry, let to the discovery of related problems in a number of TeX fonts and one utility in TeX Live.

It turns out that tlmgr (and hence TeX Live Utility) do not update TeX binaries in TeX Live 2008. Consequently you cannot fix the bug by updating your TeX Live distribution with TeX Live Utility. We have created a small install package which will update pdftex and dvips in TeX Live 2008. The install package is available at

> `http://www.tug.org/mactex/fix-font-cache-bug`

This package will only update TeX Live 2008, but the page also provides the pdftex and dvips binaries for users who want to try upgrading other distributions on their own. For example, BasicTeX-2008 can certainly be fixed with these binaries.

A fuller explanation of the bug is given below. The MacTeX web site also contains a detailed description of the debugging process (see the document Debugging on that site). I urge everyone to read the material below because it answers several important questions, namely:

Q1) Why did a bug in pdftex affect Mac OS X but no other computer platform?

Q2) Why might I still encounter the bug after installing the fix?

Q3) Why doesn't the install package fix luatex and metapost?

Q4) Why doesn't the install package fix broken fonts?

Q5) Why doesn't tlmgr upgrade binaries?

Q6) Why is it important to install TeX Live 2009 when it is released?

The Detailed Explanation

Many of you are aware of the "corrupt font cache bug" on Mac OS X. While typesetting a document, the display of the output pdf suddenly develops problems. Exact symptoms vary — sometimes half of the mathematical characters are missing, and sometimes the text mysteriously changes from Computer Modern to Helvetica. When the problem occurs, it is visible in many programs at once: Preview, TeXShop, LaTeXiT, TeXniscope, etc. But the pdf file displays correctly in Adobe Acrobat. Quitting and restarting programs does

not help, but rebooting the machine usually solves the problem. An even better way to rebuild the cache is to issue the following commands in Terminal:

> atsutil databases -removeUser
> atsutil server -shutdown

The seriousness of the bug depends on usage patterns. Many of us see the problem once in six months or less, reboot, and merrily resume typesetting. But others encounter it much more often.

This problem is caused by a corrupt font cache. When GUI programs display pdf files, they call Apple's internal pdf routines. In turn, these routines call Apple's font routines to construct bitmaps for the outline fonts. These Apple routines store the bitmaps in a font cache to speed up future display. For reasons unknown until recently, this font cache can become corrupt; this is "triggered" by displaying a defective pdf file. After that trigger, display problems persist for all files. To fix the problem, the font cache needs to be rebuilt, as happens when the machine is rebooted. Adobe Acrobat does not use Apple's pdf display routines, so it is immune to the bug.

The bug doesn't occur when running TeX on other platforms. This caused many of us to conclude that the problem was due to a bug in OS X, and several developers contacted Apple about the bug. But most developers had no reliable trigger file.

In a heroic debugging campaign a couple of weeks ago, Melissa O'Neill, the author of the program "PDF to Keynote", traced the bug to a particular line of code in pdftex. The details will be given in a subsequent email; suffice it to say that pdftex source code contained the symbol > in a spot where >= was required. This bug has been present in pdftex since 2004.

Despite the evidence, the bug is in pdftex and dvips, not in OS X. Without providing complete details, let me explain how that could happen. When pdf files are created, the fonts are added to the document but only the subset of characters actually used are placed in the pdf file. Outline font files often contain subroutines which draw commonly used portions of characters. When creating subsets of these fonts, some of these subroutines will be "orphaned" and no longer called by the remaining characters. Such orphaned routines should be replaced by RETURN in the font subset, but instead they were being replaced with "meaningful junk" in certain circumstances.

Since we do not have access to Apple's code, the rest of the explanation is a conjecture. We conjecture that Apple optimized its font routines to call subroutines once at the start rather than many times during expansion of individual characters. This optimization then called the incorrect orphaned routines, even though the characters required for the document didn't need them. The problem didn't occur on other platforms because their font routines had not been optimized in this way.

Once you install the repaired pdftex and dvips binaries, pdf files created by these programs will not contain damaged fonts and thus won't trigger the bug. But unfortunately, your

old pdf files may still be damaged. In addition, users on other platforms won't have the repaired binaries for some time, so pdf files sent by colleagues may still be damaged and trigger the bug. That is why you may still experience the bug from time to time until the TeX world catches up.

It isn't even necessary to display an old damaged pdf file to trigger the bug, because Apple's QuickLook software and the software in Finder which gives a miniature version of the first page in the file browser can trigger it. If you keep experiencing the bug after applying the fix, you may need to retypeset files which you often open.

After the initial bug was discovered, an informal debugging team of Melissa O'Neill, Thanh The Han, the author of pdftex, and Jin-Hwan Cho, the author of dvipdfmx, began systematically examining other programs and files in TeX. I summarize some of their work in the Debugging document mentioned earlier. Suffice it to say that they discovered a large number of broken font files in TeX. These fonts contained subroutines which did not end with RETURN, even though the Adobe font specification requires this. Knuth's Computer Modern files and their modern Type 1 forms don't have this problem. Files created with the defective fonts could also trigger the bug. Thanh then patched pdftex to make it automatically add the missing RETURN when a defective font is used. Thus if a broken font is used by the patched pdftex or dvips, it will not cause problems.

Later debugging discovered the font utility in TeX which created these incorrect fonts; that utility has been patched. But still later, a number of other TeX fonts were discovered which did not pass consistency tests for correctness. We don't know if these fonts trigger the bug.

It is possible that other fonts not associated with TeX corrupt the font cache. The debugging team doesn't want to let Apple off the hook, stating "the right response to invalid data isn't to behave in arbitrary ways." You can be sure that Apple will be contacted with definitive information about our TeX experience.

The incorrect fonts can be updated by tlmgr and TeX Live Utility, so gradually these font problems will go away. If you do not have TeX Live Utility, you can obtain it at `http://code.google.com/p/mactlmgr/`. All of the programs, utilities, and fonts will be correct in TeX Live 2009 when it appears, so we urge users to upgrade to it.

The code containing the pdftex bug was used with little change in modern versions of dvips, so fixing the bug in dvips was not difficult. However, it wasn't so easy to find the corresponding code in luatex and metapost. Instead of waiting for decisive fixes in these programs, we decided to just release the fixed pdftex and dvips. This should fix the bug for most users. Notice that pdftex is used when typesetting in "pdftex" mode and dvips is used when typesetting in "TeX and DVI" mode. XeTeX does not have the bug.

Finally, some of you may wonder why tlmgr does not upgrade binary files. TeX Live binaries are made by a team of many volunteers, most responsible for a single platform. These people work over the course of months as TeX Live goes into final production. If TeX

Live were to upgrade binaries continuously, the team would have to operate throughout the year, which is not a realistic requirement.

In addition, the authors of TeX Live use the interval between releases to revise the build scheme and make sure that all of the binary components of TeX Live remain compatible. TeX Live does not depend on specific library files in your operating system. Instead it contains its own version of these libraries, so the entire TeX system operates in a sort of "box" immune from changes in your system. Keeping this box internally consistent is a difficult task, and is the main reason that the core of TeX Live is released once a year (with upgrades via tlmgr for fonts, style files, etc.) rather than dribbling out over time.

Dick Koch