What is new in XƎTEX 0.9999?

Khaled Hosny

One of the strengths of XƎTEX is the use of external libraries from the underlying system as well as from third parties to do the heavy lifting of supporting modern font technologies and text layout requirements, as well as various aspects of Unicode support. Unicode and modern fonts support can be hard to get right and requires a great deal of time and effort; by using those external libraries we can build upon already existing work in these areas. For OpenType layout we were using the ICU Layout Engine and SilGraphite for Graphite layout. On Mac OS X we were using Apple's Font Manager API for locating fonts, ATSUI for AAT layout and QuickTime for loading images. On other systems we were using FreeType and FontConfig for loading and locating fonts, and Poppler for PDF images.

But all this is not without a cost; depending on external libraries requires continuous maintenance effort to catch up with changes in these external dependencies, if we want to remain relevant. So version 0.9999 of XƎTEX saw a long overdue update to the underlying libraries.

When OpenType support in XƎTEX was first introduced in 2006, the ICU Layout Engine was the best, cross-platform, free software choice at the time, though it had a limited API and was also missing several features that are vital for XƎTEX, so we were using (and maintaining) a locally patched version for our own use.

However, over the past few years the ICU Layout Engine has become unmaintained and many bugs have crept into it, while in the meantime the new and more widely-supported HarfBuzz library has emerged and reached a mature stage. So, for 0.9999 I worked on porting XƎTEX to HarfBuzz, which gives us a maintained, more complete and more widely supported layout engine[1] with less maintenance burden and simpler code on our side.

The switch to HarfBuzz also fixed some long-standing OpenType-related bugs in XƎTEX, such as support for version 2 Indic OpenType specifications, or the ability to activate and deactivate any OpenType feature for any script (previously limited to scripts that did not require any specific shaping behaviour, such as Latin or CJK), and many other smaller issues, thanks to the first tier OpenType support provided by HarfBuzz and its versatile API.

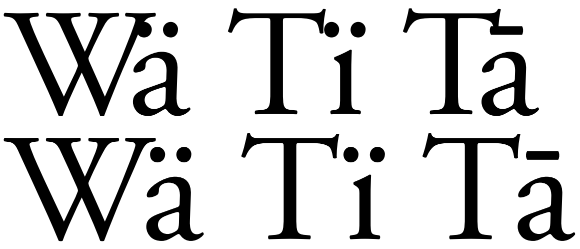

Figure 1: Some kerning bugs with Adobe fonts that were fixed after the switch to HarfBuzz.

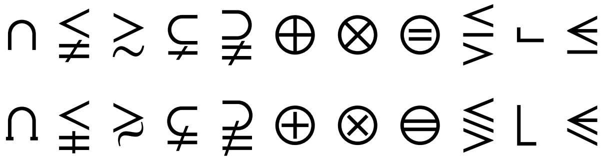

Figure 2: Some variants of mathematical symbols in the XITS Math font (in text mode).

For example, some kerning improvements are shown in fig. 1.

Another benefit we get from using HarfBuzz is support for Unicode Variation Selectors. Variation selectors are a way to represent variant glyphs for certain Unicode code points without encoding them separately or relying only on higher level protocols, like OpenType alternates, to represent them; unlike font alternates, valid variants are defined by Unicode. Variation selectors are used to encode CJK glyph variants (called Ideographic Variation Selectors), certain Mongolian contextual forms that can't be inferred from surrounding characters alone, and even stylistic variants of some mathematical symbols (however, XƎTEX does not currently support variation selectors in math mode), as shown in fig. 2.

ICU is, however, a general library for Unicode support, so we still use it for other features like encoding conversion and the optional locale-aware line breaking.

On the Graphite front, the old SilGraphite engine has been rewritten as Graphite2 to provide a more robust implementation that is optimized for the actual use cases of Graphite than what was envisioned when the original engine was written. So, in 0.9999 we moved away from the old SilGraphite engine; the layout is now done by HarfBuzz (which in turn uses Graphite2), so we have a more unified interface at the code level, but we still call Graphite2 directly for Graphite-specific line breaking support, as well as for the primitives that query Graphite features. We now also support 4-character feature tags in Graphite fonts.

[1] Coincidentally, after we finished the HarfBuzz port, ICU developers issued a statement that "*users of ICU Layout are strongly encouraged to consider the HarfBuzz project as a replacement for the ICU Layout Engine.*"

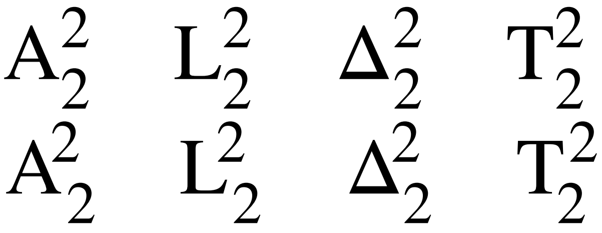

Figure 3: Before and after support for OpenType math cut-ins. (This example was artificially created to show the extremes of cut-ins.)

The situation on Mac OS X was even worse. All the Mac-specific APIs that we were using have been deprecated for several OS releases; furthermore, when Apple moved to 64-bit (`x86_64`) architecture, they didn't port the deprecated frameworks to the new architecture, forcing XƎTEX to always be built as a 32-bit application, causing all sorts of build problems and complications, and raising uncertainty about XƎTEX's future on Mac (the platform it was originally written for!) in the event of Apple dropping support for 32-bit architecture — which is not unexpected.

Fortunately, Jiang Jiang kindly offered to port XƎTEX to Core Text and other new and supported frameworks, and did all the essential work for this port. Thanks to him we no longer depend on any deprecated Mac OS X libraries and XƎTEX future on Mac OS X has been secured.

One regrettable side effect of the Mac OS X updates is that the new XƎTEX is no longer compatible with `xdv2pdf` output driver and thus that driver has been dropped from TEX Live. Remaining XƎTEX features that require `xdv2pdf` will be dropped in the next version.

Another small but important update in this release is the support for OpenType math cut-ins (see fig. 3) which provide for finer control of the horizontal placement of sub- and super-scripts than can be achieved with the (ab)use of italic corrections in traditional TEX math fonts. With this implemented, XƎTEX now supports all major features of OpenType's `MATH` table, putting it on par with the Microsoft Office and LuaTEX implementations.

One more notable change in this release is that the Unicode math primitives have been renamed to use the `\U`-prefix instead of the `\XeTeX`-prefix, for better compatibility with LuaTEX that has the same primitives with the `\U`-prefix. Old names are still allowed, but might be removed in the future.

Other miscellaneous changes in this release include: preferring OpenType or TrueType over Type 1 fonts with the same name when FontConfig is used; proper printing of multi-byte characters to the log and terminal; and proper handling for characters outside Unicode's basic multilingual plane (BMP) in `\show`, `\meaning` and `\showlists` primitives as well as in tracing output.

One last important bug that was fixed (actually in version 0.9998) is the occasional mismatch between the font used by XƎTEX and the output driver `xdvipdfmx` when multiple versions of the same font are installed, which would often result in garbage output or prevent the driver from producing any output at all.

My work on XƎTEX has been supported by the general TUG and MacTEX development funds.

◇ Khaled Hosny
`http://xetex.sourceforge.net`