The XeTeX project: typesetting for the rest of the world

Jonathan Kew
SIL International
Horsleys Green
High Wycombe HP14 3XL
England
jonathan_kew@sil.org

Abstract

This paper will introduce the XeTeX project, an extension of TeX that integrates its typesetting capabilities with the Unicode text encoding standard, supporting all the world's scripts, and with modern font technologies provided by today's operating systems and text layout services.

XeTeX offers the potential to be "TeX for the rest of the world" in several senses, as will be discussed and demonstrated:

- Much of the intimidating complexity of managing a TeX installation — in particular, the process of installing and using new fonts — is eliminated by XeTeX's integration with the host operating system's font management. This greatly reduces the "barrier to entry" into the TeX world for many non-technical users, and provides a richer and more flexible typographic environment.
- Because XeTeX is based on Unicode, the universal character encoding standard, and uses OpenType and AAT layout features in modern fonts to support complex non-Latin writing systems, it can work with Asian, Middle Eastern, and other traditionally "difficult" languages just as readily as with European languages.
- XeTeX was initially designed and implemented for Mac OS X, leveraging several key technologies available on that platform. However, this meant it was available only to a fairly small minority of potential users. However, with the introduction of XeTeX for Linux, the benefits of XeTeX become available to a new and wider community of users.

Introduction

XeTeX[1] is an extension of the TeX processor, designed to integrate TeX's "typesetting language" and document formatting capabilities with the Unicode/ISO 10646 universal character encoding for all the world's scripts, and with the font technologies available on today's computer systems. This includes fonts that support complex non-Latin writing systems and very large character sets, as well as the wide variety of Western typefaces now available.

XeTeX is in fact based on ε-TeX, and therefore includes a number of well-established extensions to TeX. These include additional registers (`\count`, `\dimen`, `\box`, etc.) beyond the 256 of each that TeX provides; various new conditional commands, tracing features, etc.; and of particular significance for multilingual work, the TeX--XeT extension for bidirectional layout.

The TeX extensions inherited from ε-TeX are not discussed further here, as they are already described in the ε-TeX documentation[2], except to note that for right-to-left scripts in XeTeX, it is necessary to set `\TeXXeTstate=1` and make proper use of the direction-changing commands `\beginR`, `\endR`, etc. Without these, there will still be some right-to-left behavior due to the inherent directionality defined by the Unicode standard for characters belonging to Hebrew, Arabic and similar scripts, but overall layout will not be correct.

Using XeTeX in conjunction with higher-level macro packages such as LaTeX or ConTeXt provides a powerful and flexible typesetting system that combines the strengths of

[1] The name XeTeX was inspired by the idea of a Mac OS X extension (hence the 'X' prefix) to ε-TeX; and as one of its intended uses is for bidirectional scripts such as Hebrew and Arabic, the name was designed to be reversible. The second letter should ideally be U+018E LATIN CAPITAL LETTER REVERSED E, but as few current fonts support this character, it is normal to use a rotated or reflected 'E' glyph. The name is pronounced as if it were written *zee-TeX*.

[2] E.g., *The ε-TeX Short Reference Manual*, `http://www.staff.uni-mainz.de/knappen/etex_ref.html`.

these well-developed markup systems and formatting tools with easy support for a huge range of industry-standard fonts and all the scripts and languages supported by the Unicode standard.

A rich world of fonts

Font installation In its early years, many users saw TeX as being inextricably linked with the Computer Modern typeface family created by Don Knuth specifically to work with TeX. In principle, other typefaces could be used, but few were available in a form that the TeX software could use, and few users knew how to install or access them.

As PostScript printers became widespread, TeX macro packages and supporting files (`.tfm`s, etc.) for fonts such as Times Roman and Helvetica were created and became part of typical TeX installations. The "New Font Selection Scheme" (NFSS) for LaTeX played a key role in allowing users easier access to alternative typefaces. A simple `\usepackage{times}` in the preamble of a LaTeX document could change the fonts throughout an article in a co-ordinated fashion.

However, for most users the choice of typefaces was still limited to those for which a preconfigured LaTeX package was available. Although various tools, scripts, and articles tried to simplify and explain the steps needed, most non-technical users were still overwhelmed by the apparent complexity and the technical knowledge required. (Do I want to use OT1 or T1 encoding, or perhaps Y&Y? How do I make `dvips` use a `.ttf` font? What exactly do I put in my `.fd` file—and where does that file need to go? Do I need to create a virtual font? How do I activate new `.map` file entries? Etc., etc.—with apologies to those for whom these issues are second nature.)

For an average user of a modern desktop computer and typical GUI software, using a new font in a document involves approximately two steps:

1. Drop the `.ttf` or `.otf` file into the computer's Fonts folder;
2. Select the font name from a menu in any application.

Any software—especially software that relates to typography—that requires a longer or more complex procedure will be perceived as "user-unfriendly" and "hard to use", and will face a barrier to wide acceptance.

XeTeX aims to bring this level of simplicity to the use of fonts with TeX. While selecting a font from a menu of installed fonts does not directly fit the TeX paradigm, the use of a new font is similarly straightforward:

1. Drop the `.ttf` or `.otf` file into the computer's Fonts folder;
2. Specify the font by name in the TeX document.

In Plain TeX terms, this second step might be:

```
\font\myfont="Charis SIL" at 9pt
\myfont Hello World
```

which results in Hello World in the typeset document.

LaTeX users do not normally declare fonts directly with TeX's `\font` command. Instead, they can say things like `\setromanfont{Charis SIL}`[3] in the preamble of the document. The present article, for example, includes the lines:

```
\usepackage{fontspec}
\setromanfont{Adobe Garamond Pro}
\setmonofont[Scale=MatchLowercase]
            {Andale Mono WT J}
```

These simple declarations are sufficient to use Adobe Garamond Pro (an OpenType font) as the primary typeface family throughout the article, with Andale Mono WT J (a monospaced TrueType font with an extended character set) for `typewriter` text, scaled to match the lowercase height of the Garamond. The fonts were installed by dropping them in the computer's Fonts folder; no additional TeX-specific steps such as file format conversions were required, no `.tfm`s, no `.fd`s, no `.map` files, etc.

Rich typographic features Modern OpenType and AAT fonts may provide a variety of sophisticated typographic features, far beyond the simple ligatures and kerning familiar to TeX users. For example, the cursive Zapfino font contains many alternate forms for use in specific contexts, as well as alternates that can be explicitly chosen by the user:

```
\font\zapfino = "Zapfino" at 7pt \zapfino
A sample of Zapfino using the default
settings built in to the font.
\font\zapfiii = "Zapfino:Stylistic
   variants=Third variant glyph set"
   at 7pt \zapfiii
A sample of Zapfino using the third of
several variant settings.
```

A sample of Zapfino using the default settings built in to the font.

A sample of Zapfino using the third of several variant settings.

Regular text faces may also include a number of interesting features, such as true Small Capitals, choice of lining (0123456789) or oldstyle (0123456789) numerals, automatic formation of arbitrary fractions (98/765) and others. The `\font` command accepts options to select whatever OpenType or AAT typographic features the font supports;

[3] This relies on the `fontspec` package by Will Robertson, which integrates XeTeX font support with the standard LaTeX font selection mechanisms.

or for LATEX users, the `fontspec` package provides a higher-level, unified interface to such features, independent of the particular font technology. The first sentence of this paragraph, for example, appears in the source document as:

```
Regular text faces may also include a
number of interesting features, such as
true{\addfontfeature{Letters=SmallCaps}
Small Capitals}, choice of lining
(0123456789) or oldstyle ({\addfontfeature
{Numbers=Lowercase}0123456789})
numerals, automatic formation of arbitrary
fractions ({\addfontfeature{Fractions=On}%
98/765}) and others.
```

Any language, any script

Unlike TEX, which treated 8-bit characters as the fundamental units of text, XƎTEX is based on the Unicode character set. By default, it reads input text as Unicode (supporting both UTF-8 and UTF-16), and expects Unicode-compliant fonts so that any valid Unicode character can be directly typeset, provided the font in use supports the relevant range of Unicode.

At a simple level, this means that with Unicode-compliant fonts, a wide range of accented and other "special" characters can be used with no special effort; they "just work":

```
\font\iwona="Iwona-Medium" at 9.5pt \iwona
Hej Slované, ještě naše slovanská řeč žije.
Óðinn átti tvá brœðr. Hét annarr Vé,
  en annarr Vílir.
\font\charis="Charis SIL" at 9pt \charis
Dünyayı verelim çocuklara hiç değilse bir
  günlüğüne.
Kur̃ bė́ga Šešùpė, kur̃ Nẽmunas tẽka, taĩ mū́sų
  tėvỹnė, gražì Lietuvà.
```

Hej Slované, ještě naše slovanská řeč žije.
Óðinn átti tvá brœðr. Hét annarr Vé, en annarr
Vílir.
Dünyayı verelim çocuklara hiç değilse bir
günlüğüne.
Kur̃ bė́ga Šešùpė, kur̃ Nẽmunas tẽka, taĩ mū́sų
tėvỹnė, gražì Lietuvà.

In addition to direct input of Unicode text, it is possible to use `\char` with Unicode character codes, so that `\char"0164\char"0119\char"015B\char"0165` will produce 'Ťęśť'. With an appropriate font selected, even characters such as Ugaritic 𐎄 or Linear B 𐂂 can be printed using their standard Unicode codepoints (those were `\char"10384` and `\char"10082`, using the Code2001 font).

Language-specific variants OpenType fonts may contain variant glyphs or behavior designed to support the typographic practices of specific languages. XƎTEX can access these features by adding a language code to the `\font` declaration; for example, Vietnamese uses different diacritic placement rules than the default "stacking" that is expected for arbitrary combinations of diacritics in generic Latin script:

```
\font\D="Doulos SIL" at 9pt
\font\V="Doulos SIL:language=VIT" at 9pt
\D cung cấp một con số duy nhất cho mỗi ký tự
\V cung cấp một con số duy nhất cho mỗi ký tự
```

cung cấp một con số duy nhất cho mỗi ký tự
cung cấp một con số duy nhất cho mỗi ký tự

Large character sets Because XƎTEX uses Unicode as its text encoding, large character sets such as those needed for Chinese and other East Asian languages present no real difficulties. Chinese characters are simply letters in the character set, just like English; all that is required is to select an appropriate font:

```
\font\myfont="STFangsong" at 10pt
% select a font that support Chinese
\myfont 基本上，计算机只是处理数字。它们指定一个数%
字，来储存字母或其他字符。在创造Unicode之前，...
```

This would be sufficient to print the Chinese characters. An additional complication for typesetting running text is that some of these languages are written without word spaces, so that TEX has no natural opportunity to break paragraphs into lines, or to justify lines to a precise width. XƎTEX solves this by offering a mechanism to find line-breaks according to the Unicode-based break rules, which can vary according to the settings of a specific locale (for example, Thai requires rules based on a dictionary to help find valid word boundaries). Further, glue can be introduced at each potential break position, so that the resulting lines of text have sufficient flexibility to be justified:

```
\XeTeXlinebreaklocale "zh"
% find line-break positions according
%   to "zh" (Chinese) locale's rules
\XeTeXlinebreakskip = 0pt plus 1pt
% add a little stretchability to
%   permit justification
```

Using these commands, XƎTEX typesets East Asian languages just as readily as English:

基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在创造Unicode之前，有数百种指定这些数字的编码系统。没有一个编码可以包含足够的字符：例如，单单欧州共同体就需要好几种不同的编码来包括所有的语言。即使是单一种语言，例如英语，也没有哪一个编码可以适用于所有的字母，标点符号，和常用的技术符号。

Complex-script languages Many non-Latin writing systems involve complex rendering rules, not simply printing one character after another in a linear fashion. Unicode encodes the fundamental characters that represent the text, but the display or printing system is responsible to map these to the proper glyphs to produce the right visual appearance. XeTeX relies on AAT or OpenType fonts with the correct tables to support such scripts, so that they automatically work in typeset documents exactly as they work in mainstream graphical applications.

For example, in Devanagari script, the short *i* vowel mark appears to the left of the preceding consonant, even though it is encoded after it; and consonant clusters are written using special "half-form" or "conjunct" characters, depending on the exact letters involved. With the appropriate fonts, this is all handled transparently during the typesetting process, with no complex macros or special preprocessing of the text:

```
\font\dev="Devanagari MT" at 9pt
  \dev हिन्दी  ⇒  हिन्दी
```

Similarly, Arabic uses contextual variants of the letters so that they connect in a cursive script:

```
\font\arb="Geeza Pro" at 9pt
  \arb العربي  ⇒  العربي
```

These examples use AAT fonts, which work with the Mac OS X Unicode text system to automatically render the text correctly. When using OpenType fonts, there is a minor difference: it is necessary to specify the script to be used, as OpenType relies on script-specific "shaping engines" to control certain aspects of the character behavior. A font may support several scripts with different behaviors, so XeTeX cannot always assume, merely from the font selected, which shaping engine should be used. Therefore, equivalent examples using OpenType fonts would look like:

```
\font\dev="Gargi_1.7:script=deva" at 9pt
  \dev हिन्दी  ⇒  हिन्दी
\font\arb="ae_AlMohanad:script=arab" at 9pt
  \arb العربي  ⇒  العربي
```

If no script is specified for an OpenType font, XeTeX will use its "generic" Latin engine, which applies common features such as ligatures and diacritic positioning, if available in the font, but does not provide the contextual shaping needed by complex Asian scripts. The results would be similar to the text as it appears in the `typewriter` text showing the input to the XeTeX processor; while the correct characters are shown, the text as a whole is not written properly.

Multi-directional text Not all languages and scripts are written from left to right across the page, which is TeX's natural way of typesetting. Some scripts run from right to left, and some are even written vertically.

For right-to-left text, XeTeX supports the TeX--XeT `\beginR` and `\endR` commands (and the `\beginL` and `\endL` commands, needed for left-to-right text embedded within a right-to-left environment), as implemented in ε-TeX. Even without these commands, individual words in scripts such as Arabic or Hebrew will appear correctly, because the Unicode characters have directional properties, but the TeX--XeT commands must be used for overall layout to work properly. For example, a typical idiom would be:

```
\everypar={\setbox0=\lastbox
    % save the paragraph indent
  \beginR % begin R-L typesetting
  \box0 } % restore indent at R side
```

This will cause all following paragraphs, until `\everypar` is reset, to default to right-to-left layout:

شروعات ۾ خدا زمين ۽ آسمان کي پيدا ڪيو. ان وقت
زمين بي‌ترتيب ۽ ويران هئي. اونهي سمنڊ جو مٿاڇرو اوندهه سان
ڍڪيل هو ۽ پاڻيءَ جي مٿان خدا جي روح ڦيرا پئي ڪيا. تڏهن
خدا حڪم ڏنو ته ”روشني ٿئي.“ سو روشني ٿي پيئي. خدا ڏٺو
ته روشني چڱي آهي. هن روشنيءَ کي اوندهه کان ڌار ڪيو.
پوءِ روشنيءَ جو نالو ”ڏينهن“ رکيائين ۽ اوندهه جو ”رات.“ سو
سانجهي ٿي ۽ صبح ٿيو. اهو پهريون ڏينهن هو.

An additional attribute that can be specified for AAT fonts in XeTeX is `vertical`. This causes the text rendering system to use vertical text-layout techniques, although it does not in itself re-orient the overall layout. Typically, glyphs will be rotated 90° counter-clockwise, likethis, and laid out according to their vertical rather than horizontal metrics.

If this capability is combined with macros that rotate the text block as a whole, which is readily achieved through graphic transformations in the output driver (see figure 1), it becomes possible to typeset languages such as Chinese using a traditional vertical layout. Figure 2 shows a sample text formatted in both horizontal and vertical styles. (The figure here is generated by code similar to that shown in figure 1, but the rotation to produce vertical text is applied just within a single `minipage` rather than to the entire page via the `\output` routine.) Note how certain glyphs such as the brackets do not undergo the same rotation as the rest of the text; the AAT `vertical` attribute automatically gives the correct behavior here.

XeTeX escapes from its nest

In the beginning The XeTeX program was begun as a project to integrate the rich support for international text and font support in Mac OS X with the TeX formatting engine. The approach of leveraging existing system libraries to handle Unicode, complex fonts and typographic features, graphics and PDF meant that a robust and highly functional system could be assembled with relatively little effort.

```
\newif\ifVertical \Verticaltrue % \Verticalfalse gives horizontal layout
\vsize=7in \hsize=4.5in \def\Vert{} % set up page size
\ifVertical % set parameters for vertical layout
  \hsize=7in \vsize=4.5in \def\Vert{:vertical} % attribute used in font defs
  % macro to rotate a box of Chinese text set with the "vertical" font attribute
  \def\ChineseBox#1{\setbox0=\vbox{\boxmaxdepth=0pt #1}\dimen0=\wd0 \dimen2=\ht0
    \vbox to \dimen0{\hbox to \dimen2{\hfil\special{x:gsave}\special{x:rotate -90}\rlap
        {\vbox to 0pt{\box0\vss}}\special{x:grestore}}\vss}}
  \def\ChineseOutput{\shipout \vbox{\ChineseBox{\makeheadline \pagebody \makefootline }}
      \advancepageno \ifnum \outputpenalty >-20000 \else \dosupereject \fi}
  \output={\ChineseOutput} \fi
\font\body="STKaiti\Vert" at 12pt \body
\font\bold="STHeiti\Vert" at 12pt  \font\title="STHeiti\Vert" at 18pt
\centerline{\title 三  国  演  义}
\bigskip
\centerline{\bold〔明〕罗贯中}
\XeTeXlinebreaklocale "zh"
\XeTeXlinebreakskip = 0pt plus 1pt minus 0.1pt
\medskip
\leftline{词曰：}
滚滚长江东逝水，浪花淘尽英雄。是非成败转头空：青山依旧在，几度夕阳红。\par
白发渔樵江渚上，惯看秋月春风。一壶浊酒喜相逢：古今多少事，都付笑谈中。\par
% ...etc...
```

Figure 1: Using a font attribute and graphic transformations to implement vertical typesetting

As a consequence of this starting point, however, XeTeX has been a single-platform system for the first two years of its existence, from the first publicly-released development version in April 2004. For Mac OS X users, it has offered an alternative to traditional TeX implementations, with some exciting new capabilities (in addition to some compatibility issues, naturally!). For the great majority of TeX users, however, font support and international typography have remained serious challenges, and a Mac OS X-only system had nothing to offer them except a tantalizing glimpse of other possibilities.

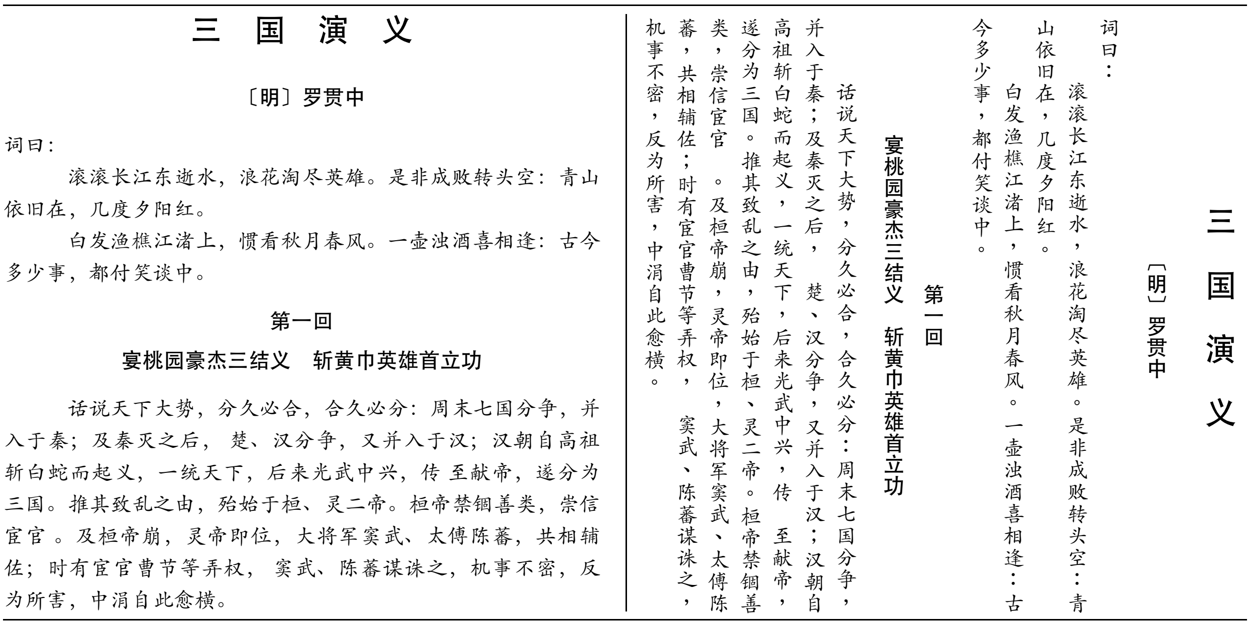
Branching out Following the initial development on the Mac OS X platform, XeTeX is now ready to "stretch its wings" and make its first moves into the wider TeX world on other architectures. At the time of writing (beginning of April, 2006), it is now possible to run XeTeX on Linux, including of course distributions running on standard x86-based PC systems.

In the Linux version of the system, the Mac OS X font and text APIs used in the original implementation are substituted with code using Fontconfig and FreeType for font access. Support for OpenType layout features and international text is provided using the ICU library (which is also used in the Mac OS X version, alongside the native ATSUI system). Graphics support, originally based on Apple's QuickTime, is provided through the ImageMagick library on Linux. These technologies have enabled creation of a Linux-based XeTeX formatting engine with the same capabilities as the Mac OS X version, except that there is no support for AAT font features—as AAT fonts are not normally used on non-Apple platforms.

The remaining part required for a complete system is an output driver that handles `.xdv` files, the extended `.dvi` format that XeTeX generates. On Mac OS X, this was implemented using the Quartz2D graphics system. As a replacement, an extended version of the DVIPDFMx driver has been created, thanks to generous assistance from Jin-Hwan Cho (one of the primary authors of that driver). This provides a portable PDF-generating back-end for the system.

To provide a graphical working environment, it is possible to configure the Kile TeX/LaTeX environment on Linux to run XeTeX or XeLaTeX as its typesetting process. This provides users with an editor that can work with Unicode TeX source documents, and can run the typesetting engine and view the resulting PDF at the touch of a keystroke, making use of TrueType and OpenType fonts just as readily as typical KDE or Gnome-based GUI applications.

Current status At present, the Linux implementation should still be considered a prototype, and will doubtless benefit from refinement over the coming months. Packaging and installation, in particular, are at early stages. But the system seems to run well, and has been successfully built on



三 国 演 义

〔明〕罗贯中

词曰：

滚滚长江东逝水，浪花淘尽英雄。是非成败转头空：青山依旧在，几度夕阳红。

白发渔樵江渚上，惯看秋月春风。一壶浊酒喜相逢：古今多少事，都付笑谈中。

第一回

宴桃园豪杰三结义　斩黄巾英雄首立功

话说天下大势，分久必合，合久必分：周末七国分争，并入于秦；及秦灭之后， 楚、汉分争，又并入于汉；汉朝自高祖斩白蛇而起义，一统天下，后来光武中兴，传 至献帝，遂分为三国。推其致乱之由，殆始于桓、灵二帝。桓帝禁锢善类，崇信宦官 。及桓帝崩，灵帝即位，大将军窦武、太傅陈蕃，共相辅佐；时有宦官曹节等弄权， 窦武、陈蕃谋诛之，机事不密，反为所害，中涓自此愈横。

三 国 演 义

〔明〕罗贯中

词曰：

滚滚长江东逝水，浪花淘尽英雄。是非成败转头空：青山依旧在，几度夕阳红。

白发渔樵江渚上，惯看秋月春风。一壶浊酒喜相逢：古今多少事，都付笑谈中。

第一回

宴桃园豪杰三结义　斩黄巾英雄首立功

话说天下大势，分久必合，合久必分：周末七国分争，并入于秦；及秦灭之后， 楚、汉分争，又并入于汉；汉朝自高祖斩白蛇而起义，一统天下，后来光武中兴，传 至献帝，遂分为三国。推其致乱之由，殆始于桓、灵二帝。桓帝禁锢善类，崇信宦官 。及桓帝崩，灵帝即位，大将军窦武、太傅陈蕃，共相辅佐；时有宦官曹节等弄权， 窦武、陈蕃谋诛之，机事不密，反为所害，中涓自此愈横。

Figure 2: Chinese text in horizontal and vertical formats

(at least) SuSE, Ubuntu, and Gentoo; users of other distributions are invited to share their experiences and contribute any necessary patches.

Looking ahead, besides refining the Linux version to ensure that it is usable on all distributions and architectures (64-bit systems will undoubtedly require some work, for example), and on other Unix-like operating systems, we also hope to adapt the code to provide a native Windows version of the tool. This will be based closely on the Linux version, except that it will need to locate installed fonts through Windows GDI instead of the Fontconfig library.

For the latest information, and downloads of both binary packages (where available) and source code (for more adventurous users), see the XƎTEX web site at `http://scripts.sil.org/xetex`. Feedback and suggestions are always welcome, with the aim of providing a powerful and flexible typesetting system that works smoothly with today's and tomorrow's text and font technologies, and with all the world's languages and scripts.