Unicode and multilingual typesetting with XeTeX

Jonathan Kew
SIL International
Horsleys Green
High Wycombe   HP14 3XL
England
jonathan_kew (at) sil dot org

This extended abstract demonstrates how extending TeX to natively handle the Unicode character set greatly simplifies the task of multilingual and multi-script typesetting. Because all characters of all the world's scripts are included in a single standard, it is not necessary to convert external encodings to a special internal representation, or to manage multiple input encodings for different languages, and any combination of scripts and languages can be freely mixed in a single document — even in a single line of text. The XeTeX extension of TeX makes it simple to use Unicode throughout, from input text to hyphenation tables and font access.

In addition to adopting Unicode as the standard character encoding, XeTeX has built-in support for modern font technologies (TrueType, OpenType, AAT), including glyph layout behavior defined in font tables. This means that complex scripts such as Indic and Arabic can be typeset with no special font setup and configuration. For example, using an off-the-shelf Arabic font, whether from a major vendor or a free font developer, involves no complex conversion processes or the creation of an "alphabet soup" of `.tfm`, `.vf`, `.ocp`, `.map`, `.enc`, `.fd`, etc. files; just drop the `.otf` or `.ttf` file into the computer's Fonts directory, and select the typeface in a TeX document.

Including Arabic in a LaTeX document can then be as simple as declaring the font to be used:

```
\usepackage{fontspec}
\newfontinstance{\arfont}[Script=Arabic]
  {Scheherazade}
% for in-line Arabic we need R-L control
\newenvironment{ar}
  {\beginR\arfont}{\endR}
```

To include الخط العربي in a document, we can just enter `\begin{ar} ... \end{ar}` in the source text, with Unicode Arabic text within the `ar` environment (not shown here because `cmtt` does not include Arabic characters).

For extended passages of Arabic, one additional factor needs to be taken into account: the overall paragraph direction should be made right-to-left, so that the paragraph indent and alignment of the last line behave as expected:

لكل إنسان حق التمتع بكافة الحقوق والحريات الواردة في هذا الإعلان،
دون أي تمييز، كالتمييز بسبب العنصر أو اللون أو الجنس أو اللغة أو الدين أو الرأي
السياسي أو أي رأي آخر، أو الأصل الوطني أو الإجتماعي أو الثروة أو الميلاد أو أي
وضع آخر، دون أية تفرقة بين الرجال والنساء.

وفضلاً عما تقدم فلن يكون هناك أي تمييز أساسه الوضع السياسي أو القانوني
أو الدولي لبلد أو البقعة التي ينتمي إليها الفرد سواء كان هذا البلد أو تلك البقعة
مستقلاً أو تحت الوصاية أو غير متمتع بالحكم الذاتي أو كانت سيادته خاضعة لأي
قيد من القيود.

Figure 1: Arabic text typeset by XeTeX using the minimal declarations shown in the text

```
% simple environment for R-L paragraphs
\newenvironment{ArabicPar}
  {\everypar={\setbox0\lastbox \beginR
   \box0 \arfont}}{}
```

This environment allows Arabic paragraphs to be properly laid out, as in figure 1.

Because XeTeX uses Unicode text and fonts, rather than a complex collection of macros to provide the script support, it is trivial to include other scripts such as Japanese, Devanagari, or many others in the same document. All we need is an appropriate Unicode font that covers the required character repertoire:

```
% Japanese, with proper line-breaking
\newfontinstance{\japfont}
  {Hiragino Kaku Gothic Pro}
\newenvironment{Japanese}
  {\XeTeXlinebreaklocale "jp"
   \XeTeXlinebreakskip0pt plus 1pt
   \japfont}{}
% Hindi
\newfontinstance{\devfont}{Devanagari MT}
\newenvironment{Hindi}
  {\devfont}{}
```

With these declarations, we can set Japanese and Hindi just as easily as Arabic. Figure 2 shows two examples using fonts included as standard with

すべて人は、人種、皮膚の色、性、言語、宗教、政治上その他の意見、国民的もしくは社会的出身、財産、門地その他の地位又はこれに類するいかなる自由による差別をも受けることなく、この宣言に掲げるすべての権利と自由とを享有することができる。

さらに、個人の属する国又は地域が独立国であると、信託統治地域であると、非自治地域であると、又は他のなんらかの主権制限の下にあるとを問わず、その国又は地域の政治上、管轄上又は国際上の地位に基ずくいかなる差別もしてはならない。

सभी को इस घोषणा में सन्निहित सभी अधिकारों और आज़ादियों को प्राप्त करने का हक़ है और इस मामले में जाति, वर्ण, लिंग, भाषा, धर्म, राजनीति या अन्य विचार-प्रणाली, किसी देश या समाज विषेश में जन्म, सम्पत्ति या किसी प्रकार की अन्य मर्यादा आदि के कारण भेदभाव का विचार न किया जायेगा।

इसके अतिरिक्त, चाहे कोीई देश या प्रदेश स्वतन्द हो, संरक्षित हो, या स्वशासन रहित हो या परिमित प्रभुसत्ता वाला हो, उस देश या प्रदेश की राजनैतिक, क्षेत्रीय या अन्तर्राष्ट्रीय स्थिति के आधार पर वहां के निवासियों के प्रति कोई फ़रक़ न रहा जाएगा।

Figure 2: Japanese and Hindi text set by XƎTEX

$$\left\{\begin{matrix}\alpha = f(z)\\ \beta = f(z^2)\\ \gamma = f(z^3)\end{matrix}\right\} \qquad \left\{\begin{matrix}x = \alpha^2 - \beta\\ y = 2\gamma\end{matrix}\right\}$$

$$\left\{\begin{matrix}\alpha = f(z)\\ \beta = f(z^2)\\ \gamma = f(z^3)\end{matrix}\right\} \qquad \left\{\begin{matrix}x = \alpha^2 - \beta\\ y = 2\gamma\end{matrix}\right\}$$

$$p_1(n) = \lim_{m\to\infty}\sum_{\nu=0}^{\infty}\bigl(1 - \cos^{2m}(\nu!^n\pi/n)\bigr)$$

$$p_1(n) = \lim_{m\to\infty}\sum_{\nu=0}^{\infty}\bigl(1 - \cos^{2m}(\nu!^n\pi/n)\bigr)$$

Figure 3: Math displays in Computer Modern (with custom encodings and multiple fonts) and Cambria Math (a single Unicode font, with no `.tfm`, etc.), typeset from the same source text

Mac OS X; similar results are obtained with OpenType fonts available on Windows, GNU/Linux, and other systems.

A more thorough implementation of script and language switching should of course also change hyphenation patterns, quote-mark styles, and other typographic niceties according to the language in use. These minimal examples show how easily multilingual fonts can be used; producing high-quality typography in varying scripts may require additional refinements.

Ongoing work on XƎTEX includes some experimental features to support the use of OpenType math fonts, which can contain a huge collection of math alphabets (italic, bold, blackboard, fraktur, script, etc.) and symbols, all encoded according to the Unicode standard. Forthcoming Microsoft products will include the Cambria Math font, and other projects such as the STIX fonts can be expected to support the same OpenType standard for math metrics. XƎTEX aims to be able to use such fonts directly, without needing to create custom-encoded subfonts, `.tfm` files, etc., and the current status of these features will be demonstrated. A couple of examples from *The TEXbook* are shown in figure 3, in both the original Computer Modern and Unicode-compliant Cambria Math fonts.