The XƎTEX reference guide

Will Robertson

June 24, 2007

1 Introduction

This document serves to summarise XƎTEX's additional features without being so much as a 'users' guide'. Note that much of the functionality addressed here is provided in abstracted form in various LATEX packages and ConTEXt modules.

The descriptions here correspond to a prerelease of version 0.997 of XƎTEX, and should be a fairly exhaustive list of its primitives. Descriptions are still a little aenemic, however. I don't have much time to maintain this document, so contributions are highly welcomed :)

Contents

Part I
XeTeX specifics

2 *The `\font` command*

The `\font` command has seen significant addition in XeTeX to facilitate rich font feature selection. Under TeX, fonts were selected like so: `\font\1="`‹*font name*›`"` with various options appended such as '`at 10pt`' or '`scaled 1.2`', with obvious meaning. This syntax has been extended in XeTeX by passing additions options through the ‹*font name*›.

This syntax looks something like

`\font\1="`‹*font name*›‹*font options*›`:`‹*font features*›`"` ‹*TeX font features*›

The ‹*font name*› is the actual name of the font (e.g., '`Charis SIL`') and the only mandatory part of the above syntax.

When using the `xdvipdfmx` driver (not yet the default in Mac OS X), it is possible to use fonts that aren't installed in the operating system by surrounding their name with square brackets. The current directory and the `texmf` trees are searched for files named in this way, or the path may be embedded in the font declaration. *E.g.*,

`\font\1="[fp9r8a]"`	*Selects 'FPL Neu' in the current directory*
`\font\1="[/myfonts/fp9r8a]"`	*Selects 'FPL Neu' somewhere else*
`\font\1="[lmroman10-regular]"`	*Selects a Latin Modern font in the* TDS

2.1 Font options

‹*Font options*› are only applicable when the font is selected through the operating system (*i.e.*, without square brackets), and may be any concatenation of the following:

`/B` Use the bold version of the selected font.
`/I` Use the italic version of the selected font.
`/BI` Use the bold italic version of the selected font.
`/IB` Same as `/BI`.
`/S=`x Use the version of the selected font corresponding to the optical size x pt.
`/AAT` Explicitly use the ATSUI renderer (Mac OS X only).
`/ICU` Explicitly use the ICU OpenType renderer (only useful on Mac OS X).

2.2 Font features

The ‹*font features*› is a comma or semi-colon separated list activating or deactivating various AAT or OpenType font features, which will vary by font. The XƎTEX documentation files `aat-info.tex` and `opentype-info.tex` provide per-font lists of supported features.

2.2.1 Arbitrary AAT or OpenType features

OpenType font features are chosen with standard tags[1]. They may be either comma- or semicolon-separated, and prepended with a + to turn them on and a - to turn them off.

Example:

```
\font\warnock="Warnock Pro/I/S=5:+smcp" at 12pt
\warnock This is the OpenType font Warnock Pro in italic
         with small caps at a small optical size.
```

THIS IS THE OPENTYPE FONT WARNOCK PRO IN ITALIC WITH SMALL CAPS AT A SMALL OPTICAL SIZE.

[1] `http://www.microsoft.com/typography/otspec/featuretags.htm`

Varying depending on the language and script in use (see section 2.2.3), a small number of OpenType features, if they exist, will be activated by default.

AAT font features are specified by strings within each font. Therefore, even equivalent features between different fonts can have different names.

> *Example:*
>
> ```
> \font\hoefler="Hoefler Text/B:Letter Case=Small Caps" at 12pt
> \hoefler This is the AAT font Hoefler Text in bold with small caps.
> ```
>
> **This is the AAT font Hoefler Text in bold with small caps.**

2.2.2 *Options for all fonts*

Some font features may be applied for any font. These are

`mapping=`*`<font map>`*

Uses the specified font mapping for this font. This uses the TECKit engine to transform unicode characters in the last-minute processing stage of the source. For example, `mapping=tex-text` will enable the classical mappings from ugly ascii ``` ``---'' ``` to proper typographical glyphs "—", and so on.

`color=`*`RRGGBB`*`[`*`TT`*`]`

Triple pair of hex values to specify the colour in RGB space, with an optional value for the transparency.

`letterspace=`x

Adds x/S space between letters in words, where S is the font size.

2.2.3 *OpenType script and language support*

OpenType font features (and font behaviour) can vary by script[2] ('alphabet') and by language[3]. These are selected with four and three letter tags, respectively.

`script=`*`<script tag>`*

Selects the font script.

`language=`*`<lang tag>`*

Selects the font language.

2.2.4 *Multiple Master and Variable Axes AAT font support*

`weight=`x

Selects the normalised font weight, x.

[2] `http://www.microsoft.com/typography/otspec/scripttags.htm`

[3] `http://www.microsoft.com/typography/otspec/languagetags.htm`

`width=`x

Selects the normalised font width, x.

`optical size=`x

Selects the optical size, x pt. Note the difference between the `/S` font option, which selects discrete fonts.

2.2.5 *Vertical typesetting*

For AAT fonts only?

`vertical`

Enables glyph rotation in the output so vertical typesetting can be performed.

3 *XƎTEX's* `\specials`

To be addressed. Hopefully not by me.

Part II
New commands

4 *Font primitives*

`\XeTeXuseglyphmetrics`

Counter to specify if the height and depth of characters are taken into account while typesetting (≥ 1). Otherwise (< 1), a single height and depth for the entire alphabet is used. Gives better output but is slower. Activated (≥ 1) by default.

> *Example:*
>
> ```
> \XeTeXuseglyphmetrics=0 \fbox{a}\fbox{A}\fbox{j}\fbox{J} vs.
> \XeTeXuseglyphmetrics=1 \fbox{a}\fbox{A}\fbox{j}\fbox{J}
> ```
>
> a A j J vs. a A j J

`\XeTeXglyph` ‹*Glyph slot*›

Inserts the glyph in ‹*slot*› of the current font. **Font specific**, so will give different output for different fonts.

\XeTeXglyphindex "*‹Glyph name›*" *‹space› or* \relax
Expands to the *‹glyph slot›* corresponding to the (possibly font specific) *‹glyph name›* in the currently selected font. Only works for TrueType fonts (or TrueType-based OpenType fonts) at present. Use fontforge or similar to discover glyph names.

\XeTeXcharglyph *‹Char code›*
Expands to the default glyph number of character *‹Char code›* in the current font, or 0 if the character is not available in the font.

> *Example:*
>
> ```
> \font\1="Charis SIL"\1
> The glyph slot in Charis SIL for the Yen symbol is:
>     \the\XeTeXglyphindex"yen" . % the font-specific glyph name
> Or: \the\XeTeXcharglyph"00A5.   % the unicode character slot
>
> This glyph may be typeset with the font-specific glyph slot:
> \XeTeXglyph1458,
> or the unicode character slot:
> \char"00A5.
> ```
>
> The glyph slot in Charis SIL for the Yen symbol is: 1458. Or: 1458.
> This glyph may be typeset with the font-specific glyph slot: ¥, or the unicode character slot: ¥.

\XeTeXfonttype *‹font›*
Expands to a number corresponding to which renderer is used for a *‹font›*:
0 for TEX (a legacy TFM-based font);
1 for ATSUI (usually an AAT font);
2 for ICU (an OpenType font);
3 for Graphite.

Example:

```
\newcommand\whattype[1]{%
  \texttt{\fontname#1} is rendered by
  \ifcase\XeTeXfonttype#1\TeX\or ATSUI\or ICU\fi.\par}
\font\1="cmr10"
\font\2="Hoefler Text"
\font\3="Charis SIL"
\font\4="Charis SIL/AAT"
\whattype\1\whattype\2\whattype\3\whattype\4
```

`cmr10` is rendered by TeX.

`"Hoefler Text"` is rendered by ATSUI.

`"Charis SIL"` is rendered by ICU.

`"Charis SIL/AAT"` is rendered by ATSUI.

4.1 OpenType fonts

`\XeTeXOTcountscripts` *‹Font›*
Expands to the number of scripts in a font.

`\XeTeXOTscripttag` *‹Font› ‹Integer, n›*
Expands to the n-th script tag of a font.

`\XeTeXOTcountlanguages` *‹Font› ‹Script tag›*
Expands to the number of languages in the script of a font.

`\XeTeXOTlanguagetag` *‹Font› ‹Script tag› ‹Integer, n›*
Expands to the n-th language tag in the script of a font.

`\XeTeXOTcountfeatures` *‹Font› ‹Script tag› ‹Language tag›*
Expands to the number of features in the language of a script of a font.

`\XeTeXOTfeaturetag` *‹Font› ‹Script tag› ‹Language tag› ‹Integer, n›*
Expands to the n-th feature tag in the language of a script of a font.

4.2 *AAT fonts*

4.2.1 *Features*

`\XeTeXcountfeatures` ‹*font*›
Expands to the number of features in the ‹*font*›.

`\XeTeXfeaturecode` ‹*font*› ‹*integer, n*›
Expands to the feature code for the n-th feature in the ‹*font*›.

`\XeTeXfeaturename` ‹*font*› ‹*feature code*›
Expands to the name corresponding to the ‹*feature code*› in the ‹*font*›.

`\XeTeXisexclusivefeature` ‹*font*› ‹*feature code*›
Expands to a number greater than zero if the feature of a font is exclusive (can only take a single selector).

4.2.2 *Feature selectors*

`\XeTeXcountselectors` ‹*font*› ‹*feature*›
Expands to the number of selectors in a ‹*feature*› of a ‹*font*›.

`\XeTeXselectorcode` ‹*font*› ‹*feature code*› ‹*integer, n*›
Expands to the selector code for the n-th selector in a ‹*feature*› of a ‹*font*›.

`\XeTeXselectorname` ‹*font*› ‹*feature code*› ‹*selector code*›
Expands to the name corresponding to the ‹*selector code*› of a feature of a ‹*font*›.

`\XeTeXisdefaultselector` ‹*font*› ‹*feature code*› ‹*selector code*›
Expands to a number greater than zero if the selector of a feature of a font is on by default.

4.2.3 *Variation axes*

`\XeTeXcountvariations` ‹*font*›
Expands to the number of variation axes in the ‹*font*›.

`\XeTeXvariation` ‹*font*› ‹*integer, n*›
Expands to the variation code for the n-th feature in the ‹*font*›.

`\XeTeXvariationname` ‹*font*› ‹*variation code*›
Expands to the name corresponding to the ‹*feature code*› in the ‹*font*›.

`\XeTeXvariationmin` ‹*font*› ‹*variation code*›
Expands to the minimum value of the variation corresponding to the ‹*variation code*› in the ‹*font*›.

`\XeTeXvariationmax` ‹*font*› ‹*variation code*›
Expands to the maximum value of the variation corresponding to the ‹*variation code*› in the ‹*font*›.

`\XeTeXvariationdefault` ‹*font*› ‹*variation code*›
Expands to the default value of the variation corresponding to the ‹*variation code*› in the ‹*font*›.

4.3 *Graphite fonts*

To do.

4.4 *Maths fonts*

The primitives described following are extensions of TeX's 8-bit primitives.

In the following commands, ‹*fam.*› is a number (0–255) representing font to use in maths. ‹*math type*› is the 0–7 number corresponding to the type of math symbol; see a TeX reference for details.

`\XeTeXmathcode` ‹*char slot*› [=] ‹*math type*› ‹*fam.*› ‹*glyph slot*›
Defines a maths glyph accessible via an input character. Note that the input takes *three* arguments unlike TeX's `\mathcode`.

`\XeTeXmathcodenum` ‹*char slot*› [=] ‹*math type/fam./glyph slot*›
Pure extension of `\mathcode` that uses a 'bit-packed' single number argument. Can also be used to extract the bit-packed mathcode number of the ‹*char slot*› if no assignment is given.

`\XeTeXmathchardef` ‹*control sequence*› [=] ‹*math type*› ‹*fam.*› ‹*glyph slot*›
Defines a maths glyph accessible via a control sequence.

`\XeTeXdelcode` ‹*char slot*› [=] ‹*fam.*› ‹*glyph slot*›
Defines a delimiter glyph accessible via an input character.

`\XeTeXdelcodenum` ‹*char slot*› [=] ‹*fam./glyph slot*›
Pure extension of `\delcode` that uses a 'bit-packed' single number argument. Can also be used to extract the bit-packed mathcode number of the ‹*char slot*› if no assignment is given.

`\XeTeXdelimiter` ‹*math type*› ‹*fam.*› ‹*glyph slot*›
Typesets the delimiter in the ‹*glyph slot*› in the family specified of either ‹*math type*› 4 (opening) or 5 (closing).

`\XeTeXradical` ‹*fam.*› ‹*glyph slot*›
Typesets the radical in the ‹*glyph slot*› in the family specified.

5 *Character classes*

The idea behind character classes is to define a boundary where tokens can be added to the input stream without explicit markup. It was originally intended to add glue around punctuation to effect correct Japanese typesetting. This feature can also be used to adjust space around punctuation for European traditions. The general nature of this feature, however, lends it to several other useful applications including automatic font switching when small amounts of another language (in another script) is present in the text.

`\XeTeXinterchartokenstate`
Counter. If positive, enables the character classes functionality.

`\XeTeXcharclass` ‹*char slot*› [=] ‹*class number*›
Assigns a class corresponding to ‹*class number*› (range 0–255) to a ‹*char slot*›. Most characters are class 0 by default. Class 1 is for CJK ideographs, classes 2 and 3 are CJK punctuation. The boundary of a text string is considered class 255. Special case class 256 is ignored; useful for diacritics so I'm told.

`\XeTeXinterchartoks` ‹*class num. 1*› ‹*class num. 2*› [=] `{`‹*token list*›`}`
Defines tokens to be inserted at the interface between ‹*class num. 1*› and ‹*class num. 2*› (in that order).

Example:

```
\XeTeXinterchartokenstate = 1
\XeTeXcharclass `\a 7
\XeTeXcharclass `\A 8
\XeTeXcharclass `\B 9

% between "a" and "A":
\XeTeXinterchartoks 7 8 = {[\itshape}
\XeTeXinterchartoks 8 7 = {\upshape]}

% between " " and "A":
\XeTeXinterchartoks 255 9 = {\bgroup\color{blue}}
\XeTeXinterchartoks 9 255 = {\egroup}

% between "B" and "B":
\XeTeXinterchartoks 9 9 = {.}

aAa A a B aBa BB
```

a[*A*]a A a B aBa B.B

In the above example the input text is typeset as

```
a[\itshape A\unshape]a A a \bgroup\color{blue}B\egroup aBa B.B
```

6 Encodings

`\XeTeXinputencoding` ‹*Charset name*›
Defines the input encoding of the following text.

`\XeTeXdefaultencoding` ‹*Charset name*›
Defines the input encoding of subsequent files to be read.

7 Line breaking

`\XeTeXdashbreakstate` ‹*Integer*›
Specify whether line breaks after en- and em-dashes are allowed. Off, 0, by default.

`\XeTeXlinebreaklocale` ‹*Locale ID*›
Defines how to break lines for multilingual text.

`\XeTeXlinebreakskip` ‹*Glue*›
Inter-character linebreak stretch

`\XeTeXlinebreakpenalty` ‹*Integer*›
Inter-character linebreak penalty

`\XeTeXupwardsmode` ‹*Integer*›
If greater than zero, successive lines of text (and rules, boxes, etc.) will be stacked upwards instead of downwards.

8 Graphics

This description is incomplete.

`\XeTeXpicfile` ‹*filename*› ‹*optional options*›
Insert an image.

`\XeTeXpdffile` ‹*filename*› ‹*optional options*›
Insert (pages of) a PDF.

8.1 Parity with pdfTEX

`\pdfpageheight` ‹*dimension*›
The height of the PDF page.

`\pdfpagewidth` ‹*dimension*›
The width of the PDF page.

`\pdfsavepos`
Saves the current location of the page in the typesetting stream.

`\pdflastxpos`
Retrieves the horizontal position saved by the above.

`\pdflastypos`
Retrieves the vertical position saved by the above.

9 *Misc.*

`\XeTeXversion`
Expands to a number corresponding to the XƎTEX version.

`\XeTeXrevision`
Expands to a string corresponding to the XƎTEX revision number.

> *Example:*
>
> ```
> The \XeTeX\ version used to typeset this document is:
> \the\XeTeXversion\XeTeXrevision
> ```
>
> The XƎTEX version used to typeset this document is: 0.997-dev