The xltxtra package

Will Robertson

2006/06/13 v0.2

Contents

1 Introduction

This document describes the xltxtra package. It implements in general improved functionality for broken or sub-standard LaTeX methods when using the XeTeX format.

Documentation is slim, and a bit ugly at this stage. I intend to improve it at some stage...Anyway, look through the sections to see what's contained within.

File I

The xltxtra package

This is the package.

```
\ProvidesPackage{xltxtra}
  [2006/06/13 v0.2 Improvements for the XeTeX/LaTeX format]
```

Required packages

```
\RequirePackage{ifxetex}
\RequireXeTeX
\RequirePackage{graphicx}
\RequirePackage{fontspec}
\RequirePackage{xunicode}
```

2 Logos

\XeTeX
\XeLaTeX

The TEX-related logos people insist upon using need to be tuned on a per-font basis. This package will eventually allow this, but for now, it's baby steps. The XƎTEX and XƎLATEX logos are provided.

The various TEX-like logos that extend outside the regular vertical alphabetic bounds of running text have the unfortunate side-effects in XƎTEX of often overrunning the `\baselineskip`. Putting the logos in zero-height boxes prevents this problem. Actually, this problem doesn't happen anymore.

Here're some examples. The default:

TEX XƎTEX LATEX XƎLATEX

```
\TeX\ \XeTeX\ \LaTeX\ \XeLaTeX
```

Notice it's a bit tight when not using Computer Modern, as here:

TEX XƎTEX LATEX XƎLATEX

```
\usefont{OT1}{cmr}{m}{n}
\TeX\ \XeTeX\ \LaTeX\ \XeLaTeX
```

This package provides a *non-stable* method of specifying the spacings in these logos. In the future, it will hopefully adjust somewhat automatically. To do:

- adapt \LaTeX to use small caps if available...
- ...otherwise, need a scaling factor, and maybe a vertical nudge factor
- add other logos
- per-font parameters, with some defaults for common fonts
- add 'low contrast' small caps versions, et al.
- probably break out the whole thing into its own package, if it works

\TeX@logo@spacing

#1: Kern between T & eX
#2: Kern between Te & X
#3: Lowering amount for E in TeX
#4: Kerning between L & aTeX
#5: Kerning between La & TeX
#6: Kerning between Xe & LaTeX

This macro defines new \TeX and \XeTeX logos. Parameters must be tuned on a per-font basis:

TEX XƎTEX LATEX XƎLATEX

```
\TeX@logo@spacing{-0.12em}{-0.12em}%
  {0.5ex}{-0.3em}{-0.12em}{-0.1em}
\TeX\ \XeTeX\ \LaTeX\ \XeLaTeX
```

Warning! This macro will **definitely** change in the future. If you care abouts backwards compatibility in your documents, copy+paste the definitions below rather than using \TeX@logo@spacing.

```
\newlength\xxt@kern@Te
\newlength\xxt@kern@eX
\newlength\xxt@lower@e
\newlength\xxt@kern@La
\newlength\xxt@kern@aT
\newlength\xxt@kern@eL
\newcommand*\TeX@logo@spacing[6]{%
  \setlength\xxt@kern@Te{#1}%
  \setlength\xxt@kern@eX{#2}%
  \setlength\xxt@lower@e{#3}%
  \setlength\xxt@kern@La{#4}%
  \setlength\xxt@kern@aT{#5}%
  \setlength\xxt@kern@eL{#6}%
}
\DeclareRobustCommand\TeX{%
  \leavevmode
  \smash{%
```

```
    T\kern\xxt@kern@Te
    \lower\xxt@lower@e\hbox{E}\kern\xxt@kern@eX X}%
  \spacefactor1000\relax}
\DeclareRobustCommand{\LaTeX}{%
  \leavevmode
  \smash{%
  L\kern\xxt@kern@La
  {\sbox\z@ T%
    \vbox to\ht\z@{\hbox{\check@mathfonts
      \fontsize\sf@size\z@
      \math@fontsfalse\selectfont
      A}%
    \vss}%
  }%
  \kern\xxt@kern@aT
  \TeX}}
\DeclareRobustCommand\XeTeX{%
  \leavevmode
  \smash{%
   X\lower\xxt@lower@e
   \hbox{\kern\xxt@kern@eX
     \ifnum\XeTeXfonttype\font>0
       \ifnum\XeTeXcharglyph"018E>0
         \char"018E\relax
       \else
         \ifdim\fontdimen1\font=0pt
           \reflectbox{E}%
         \else
           \XeTeXuseglyphmetrics=1%
           \setbox0=\hbox{E}\dimen0=\ht0\advance\dimen0by\dp0%
           \raise\dimen0\hbox{\rotatebox{180}{\box0}}%
         \fi
       \fi
     \else
       \setbox0=\hbox{E}\dimen0=\ht0\advance\dimen0by\dp0%
       \raise\dimen0\hbox{\rotatebox{180}{\box0}}%
     \fi
   }\kern\xxt@kern@Te\TeX}}%
\DeclareRobustCommand\XeLaTeX{%
   \leavevmode
   \smash{%
```

```
    X\lower\xxt@lower@e
    \hbox{\kern\xxt@kern@eX
      \ifnum\XeTeXfonttype\font>0\relax
        \ifnum\XeTeXcharglyph"018E>0\relax
          \char"018E\relax
        \else
          \ifdim\fontdimen1\font=0pt\relax
            \reflectbox{E}%
          \else
            \XeTeXuseglyphmetrics=1\relax
        \setbox0=\hbox{E}\dimen0=\ht0\advance\dimen0by\dp0\relax
            \raise\dimen0\hbox{\rotatebox{180}{\box0}}%
          \fi
        \fi
      \else
       \setbox0=\hbox{E}\dimen0=\ht0\advance\dimen0by\dp0\relax
        \raise\dimen0\hbox{\rotatebox{180}{\box0}}%
      \fi}\kern\xxt@kern@eL\LaTeX}}
\TeX@logo@spacing{-0.15em}{-0.15em}{0.5ex}{-0.36em}{-0.15em}{-0.1em}
```

3 ε-TEX functionality

Because it's just sensible, we load the package that actually allows LATEX to access the extra registers, etc., provided by ε-TEX.

```
\RequirePackage{etex}
```

4 Unicode footnote symbols

LATEX defines footnote symbols with LICRs that don't resolve well with the xunicode package; better results can be achieved by using specific unicode characters.

This problem is solved by the fixltx2e package.

```
\RequirePackage{fixltx2e}[2006/03/24]
```

5 \eminnershape

\em \emph

fixltx2e's method for checking for "inner" emphasis is a little fragile in XETEX, because font slant information might be missing from the font.

Therefore, we use LaTeX's NFSS information, which is more likely to be correct.

Nested *emphasis is* NOW *fixed.*

```
\renewcommand\eminnershape{\scshape}
\fontspec{Didot}
Nested {\em emphasis is \emph{now} fixed.}
```

```
\DeclareRobustCommand\em
  {\@nomath\em
   \edef\@tempa{\f@shape}%
   \edef\@tempb{\itdefault}%
   \ifx\@tempa\@tempb
     \eminnershape
   \else
     \emshape
   \fi}
\DeclareTextFontCommand{\emph}{\em}
\let\emshape\itshape
\let\eminnershape\upshape
```

6 Programmming bits and pieces

Thanks to a long-ago c.t.t. post by Robin Fairbairns for the code how to `\let` a robust macro.

```
\newcommand*\robust@let@nc[2]{%
  \expandafter\let\expandafter#1\csname #2 \endcsname}
```

7 \-

LaTeX defines the macro `\-` to insert discretionary hyphenation points. However, it is hard-coded to use the hyphen `-` character. Since fontspec makes it easy to change the hyphenation character on a per font basis, it would be nice if `\-` adjusted automatically.

`\-` This macro is courtesy of Frank Mittelbach and the LaTeX 2_ε source code.

```
%\CheckCommand\-{\discretionary{-}{}{}}
\DeclareRobustCommand{\-}{%
  \discretionary{%
```

```
    \char\ifnum\hyphenchar\font<\z@
      \xlx@defaulthyphenchar
    \else
      \hyphenchar\font
    \fi}{}{}}
\def\xlx@defaulthyphenchar{`\-}
```

8 `\textsuperscript` and `\textsubscript`

The new macros now allow real text inferiors and superiors:

`\textsuperscript`	abcdefghijklmnopqrstuvwxyz1234567890
`\textsubscript`	abcdefghijklmnopqrstuvwxyz1234567890

As opposed to fake ones:

`\textsuperscript*`	abcdefghijklmnopqrstuvwxyz1234567890
`\textsubscript*`	abcdefghijklmnopqrstuvwxyz1234567890

Or:

`\faketextsuperscript`	abcdefghijklmnopqrstuvwxyz1234567890
`\faketextsubscript`	abcdefghijklmnopqrstuvwxyz1234567890

But beware fonts lacking the full repertoire: (this is Adobe Jenson Pro)

`\textsuperscript`	$^{ab}c^{de}f g^{hi}j k^{lmno}p q^{rst}uvwxyz^{1234567890}$
`\textsubscript`	abcdefghijklmnopqrstu$_{vwxyz_{1234567890}}$

For OpenType fonts, the subscript feature (`subs`) is used, but if that doesn't exist then the scientific inferior feature (`sinf`) is used on the assumption that something's better than nothing. This assumption may prove to be a poor one, and the functionality of the package may change in the future.

`\faketextsubscript` `\faketextsuperscript`

The old ('fake') methods:

```
\robust@let@nc\faketextsubscript{textsubscript}
\robust@let@nc\faketextsuperscript{textsuperscript}
```

\textsubscript
\textsubscript*

Text subscripts:

```
\DeclareRobustCommand*\textsubscript{%
  \@ifstar{\faketextsubscript}{\@@textsubscript}}
\newcommand\@@textsubscript[1]{%
  \begingroup
    \c@zf@script 1818326126\relax
   \font\zf@basefont="\csname zf@family@fontdef\f@family\endcsname" at \f@size pt
    \zf@set@font@type
    \ifzf@atsui
      \zf@make@aat@feature@string{10}{2}%
      \unless\ifx\zf@thisfontfeature\@empty
        {\addfontfeature{VerticalPosition=Inferior}#1}%
      \else
        \faketextsubscript{#1}%
      \fi
    \fi
    \ifzf@icu
      \zf@check@ot@feat{+subs}%
      \if@tempswa
        {\addfontfeature{VerticalPosition=Inferior}#1}%
      \else
        \zf@check@ot@feat{+sinf}%
        \if@tempswa
       {\addfontfeature{VerticalPosition=ScientificInferior}#1}%
        \else
          \faketextsubscript{#1}%
        \fi
      \fi
    \fi
  \endgroup}
```

\textsuperscript
\textsuperscript*

Text superscripts:

```
\DeclareRobustCommand*\textsuperscript{%
  \@ifstar{\faketextsuperscript}{\@@textsuperscript}}
\newcommand\@@textsuperscript[1]{%
  \begingroup
    \c@zf@script 1818326126\relax
   \font\zf@basefont="\csname zf@family@fontdef\f@family\endcsname" at \f@size pt
    \zf@set@font@type
    \ifzf@atsui
      \zf@make@aat@feature@string{10}{1}%
```

```
      \unless\ifx\zf@thisfontfeature\@empty
        {\addfontfeature{VerticalPosition=Superior}#1}%
      \else
        \faketextsuperscript{#1}%
      \fi
    \fi
    \ifzf@icu
      \zf@check@ot@feat{+sups}%
      \if@tempswa
        {\addfontfeature{VerticalPosition=Superior}#1}%
      \else
        \faketextsuperscript{#1}%
      \fi
    \fi
  \endgroup}
```

9 \vfrac

A command for setting vulgar fractions based on AAT or OpenType font features. Not really recommended for many purposes, depending on your text, but it's a good example.

AAT: $^{123}/_{456}$
ICU: $^{123}/_{456}$

```
\fontspec{Hoefler Text}
AAT: \vfrac{123}{456}\par
\fontspec{Warnock Pro}
ICU: \vfrac{123}{456}
```

`\vfrac` #1: Numerator
#2: Denominator

No error checking is done to ensure that the font actually has the neceessary features. Requires the xunicode package for `\textfraction-solidus`.

```
\newcommand*\vfrac[2]{%
  \begingroup
    \c@zf@script 1818326126\relax
   \font\zf@basefont="\csname zf@family@fontdef\f@family\endcsname" at \f@size pt
    \zf@set@font@type
    \ifzf@atsui
      {\addfontfeature{VerticalPosition=Superior}#1}%
```

```
      \textfractionsolidus
    {\addfontfeature{VerticalPosition=Inferior}#2}%
  \fi
  \ifzf@icu
    {\addfontfeature{VerticalPosition=Numerator}#1}%
     \textfractionsolidus
    {\addfontfeature{VerticalPosition=Denominator}#2}%
  \fi
\endgroup}
```

10 Named glyphs

Along the way somewhere, XeTeX added support for selecting glyphs from a TrueType-based OpenType font based on their internal glyph name. Jonathan Kew posted the following definition as a nice interface to it.

¥ [smile]

```
\fontspec{Charis SIL}
\namedglyph{yen}
\namedglyph{smile}
```

`\namedglyph` #1: Name of the font glyph to be typeset

```
\newcommand\namedglyph[1]{%
  \@tempcnta=\XeTeXglyphindex "#1"\relax
  \ifnum\@tempcnta>0
    \XeTeXglyph\@tempcnta
  \else
    \xxt@namedglyph@fallback{#1}%
  \fi}
```

`t@namedglyph@fallback` Redefine this macro to change how glyph names that aren't found get typeset.

```
\newcommand\xxt@namedglyph@fallback[1]{[#1]}
```

Change History

v0.1

v0.2

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.