The `xltxtra` package

Will Robertson

2006/05/30 v0.3

1 Introduction

This document describes the `xltxtra` package. It implements some odds-and-ends features and improved functionality for broken or sub-standard LaTeX methods when using the XeTeX format.

1.1 Usage

Easy: `\usepackage{xltxtra}`. This package automatically loads the following packages: `fixltx2e`, `etex`, `xunicode`, `fontspec`.

There are some package options to disable various functionality that could clash with other things:

`no-sscript` Swaps the definitions of `\textsubscript` and `\textsuperscript` with their respective starred versions, as described in section 2.1.

`no-emph` Disables the redefinition of `\emph` and `\em` described in section 2.2.

`no-logos` Disables the redefinition of `\TeX`, etc. described in section 2.4, but *does* still define the `\XeTeX` and `\XeLaTeX` logo commands.

`no-hyphen` Disables the redefinition of `\-` (probably harmless anyway) described in section 2.5.

2 Features

2.1 `\textsuperscript` and `\textsubscript`

These two macros have been redefined to take advantage, if possible, of actual superior or inferior glyphs in the main document font. This is very important for high-quality typesetting — compare this first example to the third; yes, they are the same font.

`\textsuperscript`	abcdefghijklmnopqrstuvwxyz1234567890
`\textsubscript`	abcdefghijklmnopqrstuvwxyz1234567890

But will fall back on 'faked' ones if they don't exist: (this is Didot)

`\textsuperscript`	abcdefghijklmnopqrstuvwxyz1234567890
`\textsubscript`	abcdefghijklmnopqrstuvwxyz1234567890

The original definitions are available in starred verions of the commands:

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

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

`\textsuperscript`	abcdefghijklmnopqrstuvwxyz1234567890
`\textsubscript`	abcdefghijklmnopqrstuvwxyz1234567890

The `[no-sscript]` package option will swap the definitions of the starred and non-starred versions of the commands described above if the new definitions are undesirable.

The macros `\realsubscript`, `\realsuperscript`, `\fakesubscript`, and `\fakesuperscript` may be used to access the 'new' and 'old' functionalities regardless of the `[no-sscript]` package option.

2.2 Inner emphasis

`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.}
```

The `[no-emph]` package option will disable this redefinition.

2.3 Unicode footnote symbols

By default LaTeX defines symbolic footnote characters in terms of commands that don't resolve well; better results can be achieved by using specific unicode characters or proper LICRs with the `xunicode` package.

This problem has been solved by loading the `fixltx2e` and `xunicode` packages in `xltxtra`.

2.4 Logos

This part of the package essentially exists to define the `\XeTeX`and`\XeLaTeX` logos. Here're some examples. The default:

TeX XeTeX LaTeX XeLaTeX

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

Notice it's a bit tight compared to not using Computer Modern, for which the logos were designed:

TeX XeTeX LaTeX XeLaTeX

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

Look in the implementation corresponding to this section to see how to customise the spacings in these logos, but be warned it's fairly crude and may change in the future.

The `[no-logos]` package option will not redefine `\TeX` or `\LaTeX` but will still define `\XeTeX` and `\XeLaTeX`. (The only advantage for not doing

this is more consistency when customising the spacing, which isn't really recommended anyway...)

2.5 Discrectionary hyphenation: `\-`

LaTeX defines the macro `\-` to insert discretionary hyphenation points. However, it is hard-coded in LaTeX 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 — and now it does.

2.6 Vulgar fractions

The `\vfrac` 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 of how to program such things using `fontspec`.

AAT: 123⁄456
ICU: 123⁄456

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

(This can be achieved in regular LaTeX with the `nicefrac` package, but don't believe its name: these fractions aren't nice!)

Only use it when you know it will work; no warnings are given if the font doesn't support it.

2.7 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}
```

2.8 The `\showhyphens` command

The default definition doesn't work in XeTeX. A new version, written by Jonathan Kew, is included in the package that *does* work. Minor differences with the original: the showing of hyphens in the console output will be marked with explanatory text. Also, multiple words, separated by commas, will end up in separate instances of 'showing hyphens'.

File I
The xltxtra package

This is the package implementation.

```
1 \ProvidesPackage{xltxtra}
2   [2006/05/30 v0.3 Improvements for the "XeLaTeX" format]
```

Change History

v0.1

v0.2

Option processing

```
\newif\if@xxt@nosscript@
\newif\if@xxt@nologos@
\newif\if@xxt@nohyphen@
\newif\if@xxt@noemph@
\DeclareOption{no-sscript}{\@xxt@nosscript@true}
\DeclareOption{no-logos}{\@xxt@nologos@true}
\DeclareOption{no-hyphen}{\@xxt@nohyphen@true}
\DeclareOption{no-emph}{\@xxt@noemph@true}
\ProcessOptions*
```

Required packages

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

3 Programmming bits and pieces

4 Logos

`\XeTeX` `\XeLaTeX`

The TEX-related logos people insist upon using need to be tuned on a per-font basis. This package will (might!) 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.

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}%
}
\unless\if@xxt@nologos@
\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}}
\fi
\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}
```

5 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}
```

5.1 Unicode footnote symbols

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

5.2 Emph

```
\unless\if@xxt@noemph@
```

\em
\emph

```
\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

\fi
```

5.3 \-

```
\unless\if@xxt@nohyphen@
```

\- 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{`\-}

\fi
```

5.4 Subscript and superscript

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 matches current trends in OpenType font design.

Footnotes are patched to use this better `\textsuperscript`.

`\fakesubscript` `\fakesuperscript` The old ('fake') methods:

```
\DeclareRobustCommand*\fakesubscript[1]{%
  \@textsubscript{\selectfont#1}}
\DeclareRobustCommand*\fakesuperscript[1]{%
  \@textsuperscript{\selectfont#1}}
```

`\textsubscript` `\textsubscript*` `\textsuperscript` `\textsuperscript*` These commands are either defined to create fake or real sub-/super-scripts if they are starred or not, respectively. This swaps if the `[no-sscript]` package option is in effect. Text subscripts:

```
\if@xxt@nosscript@
  \DeclareRobustCommand*\textsubscript{%
    \@ifstar{\realsubscript}{\fakesubscript}}
  \DeclareRobustCommand*\textsuperscript{%
    \@ifstar{\realsuperscript}{\fakesuperscript}}
\else
  \DeclareRobustCommand*\textsubscript{%
    \@ifstar{\fakesubscript}{\realsubscript}}
```

```
    \DeclareRobustCommand*\textsuperscript{%
      \@ifstar{\fakesuperscript}{\realsuperscript}}
\fi
```

\realsubscript

```
\DeclareRobustCommand*\realsubscript[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\@tempa\@empty
        {\addfontfeature{VerticalPosition=Inferior}#1}%
      \else
        \fakesubscript{#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
          \fakesubscript{#1}%
        \fi
      \fi
    \fi
  \endgroup}
```

\realsuperscript

Text superscripts:

```
\DeclareRobustCommand*\realsuperscript[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\@tempa\@empty
        {\addfontfeature{VerticalPosition=Superior}#1}%
      \else
        \fakesuperscript{#1}%
      \fi
    \fi
    \ifzf@icu
      \zf@check@ot@feat{+sups}%
      \if@tempswa
        {\addfontfeature{VerticalPosition=Superior}#1}%
      \else
        \fakesuperscript{#1}%
      \fi
    \fi
  \endgroup}
```

Patching footnotes:

\@makefnmark

```
\def\@makefnmark{\mbox{\normalfont\textsuperscript{\@thefnmark}}}
```

\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}
```

`\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}
```

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

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

`\showhyphens` This macro is entirely due to Jonathan Kew. I wish I knew how to write these sorts of things.

```
\newbox\xxt@tempbox
\def\showhyphens#1{%
  \typeout{^^J**********************
           \string\showhyphens:
           **********************}%
  \@for\@ii:=#1\do{\xxt@showhyphens{\@ii}}%
  \typeout{^^J**********************%
           ***************%
           **********************^^J}}
\def\xxt@showhyphens#1{%
   \setbox\@tempboxa=\vbox{%
     \hsize1sp \hbadness10000 \hfuzz\maxdimen
     \everypar={} \leftskip\z@ \rightskip\leftskip
     \pretolerance\m@ne \noindent \hskip\z@ #1\par
```

```
      \global\setbox\xxt@tempbox=\hbox{}\xxt@sh@cat}%
   \setbox\@tempboxa=\hbox to \maxdimen{\unhbox\xxt@tempbox}}
\def\xxt@sh@cat{\unskip\unpenalty
   \setbox\@tempboxa=\lastbox
   \unless\ifvoid\@tempboxa
     \global\setbox\xxt@tempbox=\hbox{%
       \unhbox\@tempboxa
       \unskip\unskip
       \unhbox\xxt@tempbox}%
     \expandafter\xxt@sh@cat
   \fi}
```

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.

G

H

I

K

L

M

N

P

R