The `xltxtra` package

Will Robertson

2010/06/03 v0.5d

Contents

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`, `metalogo`, `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-logos Disables the redefinition of `\TeX`, etc. described in section §2.2, but *does* still define the `\XeTeX` and `\XeLaTeX` logo commands.

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 Logos

This part of the package essentially exists to define the `\XeTeX` and `\XeLaTeX` logos, which need to be tuned according to the font that is used. Originally I had some hard-coded definitions in here, but Andrew Moschou's `metalogo` package now provides a much more flexible and useful interface to a variety of TEX-related logos.

Here are some examples. The default:

TEX XETEX LATEX XELATEX

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

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

TEX XETEX LATEX XELATEX

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

These logos, ideally, should be hand-tuned for each font that they're used in. Please refer to the `metalogo` documentation for more information.

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

2.3 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{Skia}
AAT: \vfrac{123}{456}\\
\fontspec{Warnock Pro}
ICU: \vfrac{123}{456}
```

(This can also be achieved in regular LATEX with either the `nicefrac` or `xfrac` package.)

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

2.4 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}`

2.5 The `\showhyphens` command

The default definition doesn't work in XeTeX. A new version, written by Jonathan Kew, is included in this 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.

```
\ProvidesPackage{xltxtra}
  [2010/06/03 v0.5d Improvements for the "XeLaTeX" format]
```

Not for LuaTEX

```
\RequirePackage{ifluatex}
\ifluatex
  \PackageWarningNoLine {xltxtra} {^^J
    XLTXTRA IS TO BE USED ONLY UNDER XETEX.
    LOAD FONTSPEC DIRECTLY, INSTEAD.^^J
    ABORTING LOADING%
  }
  \RequirePackage{fontspec}[2010/05/14 v2.0]
  \expandafter \endinput
\fi
```

Required packages

```
\RequirePackage{ifxetex}
\RequireXeTeX
\RequirePackage{fontspec}[2010/05/14 v2.0]
\RequirePackage{xunicode}
```

Option processing

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

3 Logos

\XeTeX
\XeLaTeX

The TEX-related logos people insist upon using need to be tuned on a per-font basis. This package calls upon Andrew Moschou's package `metalogo` for this purpose. To tune the logos to each font, use the commands `\setlogokern`, `\setlogodrop`, etc. Refer to `mathspec`'s documentation for further details.

TEX XƎTEX LATEX XƎLATEX
LATEX 2ε

```
\setlogokern{Xe}{-0.061em}
\setlogokern{eL}{-0.057em}
\setlogokern{La}{-0.265em}
\setlogokern{aT}{-0.0585em}
\setlogokern{Te}{-0.0575em}
\setlogokern{eX}{-0.072em}
\setlogokern{eT}{-0.056em}
\setlogokern{X2}{0.1667em}
\setlogodrop{0.153em}
\setLaTeXa{\scshape a}
\setLaTeXee{\mbox{\fontspec{Times}\itshape ε}}
\TeX\ \XeTeX\ \LaTeX\ \XeLaTeX\ \LaTeXe
```

```
\RequirePackage{metalogo}
```

The `[no-logos]` package option might be in effect, in which case `\TeX`, `\LaTeX` and `\LaTeXe` should keep their original definitions (which were saved by `metalogo`).

```
\if@xxt@nologos@
  \let\TeX\original@TeX
  \let\LaTeX\original@LaTeX
  \let\LaTeXe\original@LaTeXe
\fi
```

`\TeX@logo@spacing` This macro is now deprecated. It is recommended to use the commands from `metalogo`.

```
\newcommand*\TeX@logo@spacing[6]{%
  \PackageWarning{xltxtra}{%
    Use of \protect\TeX@logo@spacing\space is deprecated,\MessageBreak
    recommend to use commands from package `metalogo' instead}
  \setlogokern{Te}{#1}%
  \setlogokern{eT}{#1}%
  \setlogokern{eX}{#2}%
  \setlogokern{Xe}{#2}%
  \setlogodrop{#3}%
  \setlogokern{La}{#4}%
  \setlogokern{aT}{#5}%
  \setlogokern{eL}{#6}}
```

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

```
\ExplSyntaxOn
\DeclareRobustCommand*\realsubscript[1]{
  \fontspec_if_fontspec_font:TF
  {
```

OpenType fonts:

```
    \fontspec_if_opentype:TF
    {
      \fontspec_if_feature:nTF {+subs}
      {
        {\addfontfeature{VerticalPosition=Inferior}#1}
      }
      {
        \fontspec_if_feature:nTF {+sinf}
        {
          {\addfontfeature{VerticalPosition=ScientificInferior}#1}
        }
        {
          \fakesubscript{#1}
        }
      }
    }
```

ATSUI fonts:

```
    {
```

```
      \fontspec_if_aat_feature:nnTF {10} {2}
      {
        {\addfontfeature{VerticalPosition=Inferior}#1}
      }
      {
        \fakesubscript{#1}
      }
    }
  }
```

Non-fontspec fonts:

```
  {
    \fakesubscript{#1}
  }
}
```

`\realsuperscript` Text superscripts:

```
\DeclareRobustCommand*\realsuperscript[1]{
  \fontspec_if_fontspec_font:TF
  {
```

OpenType fonts:

```
    \fontspec_if_opentype:TF
    {
      \fontspec_if_feature:nTF {+sups}
      {
        {\addfontfeature{VerticalPosition=Superior}#1}
      }
      {
        \fakesuperscript{#1}
      }
    }
```

ATSUI fonts:

```
    {
      \fontspec_if_aat_feature:nnTF {10} {1}
      {
        {\addfontfeature{VerticalPosition=Superior}#1}
      }
      {
        \fakesuperscript{#1}
      }
    }
  }
```

Non-fontspec fonts:

```
  {
    \fakesuperscript{#1}
  }
```

```
}
```

Patching footnotes:

\@makefnmark

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

5 Assorted commands

\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 `\textfractionsolidus`.

```
\newcommand*\vfrac[2]{
  \fontspec_if_fontspec_font:TF
  {
    \fontspec_if_opentype:TF
    {
      {\addfontfeature{VerticalPosition=Numerator}#1}
       \textfractionsolidus
      {\addfontfeature{VerticalPosition=Denominator}#2}
    }
    {
      {\addfontfeature{VerticalPosition=Superior}#1}
       \textfractionsolidus
      {\addfontfeature{VerticalPosition=Inferior}#2}
    }
  }
  {
    \PackageError {xltxtra}
      { \string\vfrac\space~can~only~be~used~with~fontspec~fonts }
      { Nothing~more~to~tell. }
  }
}
\ExplSyntaxOff
```

\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}
```