The `xltxtra` package

Will Robertson

2018/12/31 v0.7

Contents

1 Introduction

This document describes the `xltxtra` package. It implements some odds-and-ends features when using the XeLaTeX format.

1.1 Usage

Easy: `\usepackage{xltxtra}`. This package automatically loads the following packages: `fontspec`, `realscripts`, `metalogo`.

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`

This functionality is achieved through loading the `realscripts` package.

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

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`.

123⁄456

```
\fontspec{LibreCaslonText-Regular.otf}
\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}
\namedglyph{yen}
\namedglyph{smile}
```

2.5 The `\showhyphens` command

The default definition doesn't work in XeTeX. A new version, written by Enrico Gregorio, is included in this package that *does* work; note that the syntax now matches plain TeX's original rather than the comma-list approach taken by an earlier version of this package.

File I
The xltxtra package

This is the package implementation.

```
\ProvidesPackage{xltxtra}
  [2018/12/31 v0.7 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}
  \expandafter \endinput
\fi
```

Required packages

```
\RequirePackage{ifxetex}
\RequireXeTeX
\RequirePackage{fontspec}
\RequirePackage{realscripts}
```

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 }}
\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
```

4 Subscript and superscript

`\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}}
\fi
```

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`.

```
\ExplSyntaxOn
```

```
\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` Courtesy egreg.

```
\ExplSyntaxOn
\seq_new:N \l__xetex_showhyphens_seq
\box_new:N \l__xetex_show_hyphens_wrapping_box
\box_new:N \l__xetex_show_hyphens_temp_box
\box_new:N \l__xetex_show_hyphens_final_box
\box_new:N \g__xetex_show_hyphens_word_box

\cs_new_protected:Npn \xetex_show_hyphens:n #1
 {
  \box_clear:N \l__xetex_show_hyphens_final_box
  % split the input into items
```

```
  \seq_set_split:Nnn \l__xetex_showhyphens_seq { ~ } { #1 }
  % hyphenate all items
  \seq_map_function:NN \l__xetex_showhyphens_seq \xetex_hyphenate_word:n
  % set a box to the maximum dimension to force a Underfull \hbox warning
  \hbox_set_to_wd:Nnn \l__xetex_show_hyphens_final_box { \c_max_dim }
   {
    \hbox_unpack_clear:N \l__xetex_show_hyphens_final_box
   }
 }

\cs_new_protected:Npn \xetex_hyphenate_word:n #1
 {
  \vbox_set:Nn \l__xetex_show_hyphens_wrapping_box
   {% build a paragraph with the word with a very narrow line width
    \dim_set:Nn \hsize { 1sp }
    % disregard spurious messages
    \hbadness = 10000\relax
    \dim_set:Nn \hfuzz { \c_max_dim }
    % clear possible values of \everypar and other parameters
    \everypar={}
    \skip_set:Nn \leftskip { 0pt }
    \skip_set_eq:NN \rightskip \leftskip
    % skip the first step
    \pretolerance = -1\relax
    % avoid the indentation and add a skip to allow hyphenation
    \noindent
    \skip_horizontal:n { 0pt }
    #1
    \par
    \hbox_gset:Nn \g__xetex_show_hyphens_word_box {}
    % start a recursion to dismantle the paragraph just built
    \xetex_show_hyphens_split:
    % the result is put into \g__xetex_show_hyphens_word_box
   }
   % add the box to the final container
   \hbox_set:Nn \l__xetex_show_hyphens_final_box
    {
     \hbox_unpack_clear:N \l__xetex_show_hyphens_final_box
     \hbox_unpack_clear:N \g__xetex_show_hyphens_word_box
    }
 }

\cs_new_protected:Npn \xetex_show_hyphens_split:
 {
  \unskip % remove the interline glue
  \unpenalty % remove possible penalties
  % get the last line
```

```
  \box_set_to_last:N \l__xetex_show_hyphens_temp_box
  \box_if_empty:NF \l__xetex_show_hyphens_temp_box
   {% if there is a last line unpack it into a container
    \hbox_gset:Nn \g__xetex_show_hyphens_word_box
     {% the order is last to first
      \hbox_unpack_clear:N \l__xetex_show_hyphens_temp_box
      \unskip\unskip % remove spaces
      \hbox_unpack_clear:N \g__xetex_show_hyphens_word_box
     }
    % restart the recursion
    \xetex_show_hyphens_split:
   }
 }

\cs_set_eq:NN \showhyphens \xetex_show_hyphens:n
\ExplSyntaxOff
```