

The xltextra package

Will Robertson

2016/01/21 v0.6

Contents

1	Introduction	1
1.1	Usage	1
2	Features	2
2.1	<code>\textsuperscript</code> and <code>\textsubscript</code>	2
2.2	Logos	2
2.3	Vulgar fractions	3
2.4	Named glyphs	3
2.5	The <code>\showhyphens</code> command	4
I	The xltextra package	5
3	Logos	5
4	Subscript and superscript	6
5	Assorted commands	7

1 Introduction

This document describes the xltextra package. It implements some odds-and-ends features when using the X_YL^AT_EX format.

1.1 Usage

Easy: `\usepackage{xltextra}`. 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 versions 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 `\fake-superscript` 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 T_EX-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: $\frac{123}{456}$

ICU: $\frac{123}{456}$

`\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 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 `xltextra` package

This is the package implementation.

```
1 \ProvidesPackage{xltextra}
2 [2016/01/21 v0.6 Improvements for the "XeLaTeX" format]
```

Not for LuaTeX

```
3 \RequirePackage{ifluatex}
4 \ifluatex
5   \PackageWarningNoLine {xltextra} {^^}
6     XLTXTRA IS TO BE USED ONLY UNDER XETEX.
7     LOAD FONTSPEC DIRECTLY, INSTEAD.^^}
8   ABORTING LOADING%
9 }
10 \RequirePackage{fontspec}
11 \expandafter \endinput
12 \fi
```

Required packages

```
13 \RequirePackage{ifxetex}
14 \RequireXeTeX
15 \RequirePackage{fontspec}
16 \RequirePackage{realscripts}
```

Option processing

```
17 \newif\if@xxt@nosscript@
18 \newif\if@xxt@nologos@
19 \DeclareOption{no-sscript}{\@xxt@nosscript@true}
20 \DeclareOption{no-logos}{\@xxt@nologos@true}
21 \ProcessOptions*
```

3 Logos

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

```

\setlogokern{Xe}{-0.061em}
\setlogokern{eL}{-0.057em}
\setlogokern{La}{-0.265em}
\setlogokern{aT}{-0.0585em}
\setlogokern{Te}{-0.0575em}
TeX XeTeX LaTeX XeLaTeX \setlogokern{eX}{-0.072em}
LaTeX 2 $\epsilon$  \setlogokern{eT}{-0.056em}
\setlogokern{X2}{0.1667em}
\setlogodrop{0.153em}
\setLaTeXa{\scshape a}
\setLaTeXee{\mbox{\fontspec{Times}\itshape  $\epsilon$ }}
TeX\ XeTeX\ LaTeX\ XeLaTeX\ LaTeXe

```

```
22 \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`).

```

23 \if@xxt@nologos@
24 \let\TeX\original@TeX
25 \let\LaTeX\original@LaTeX
26 \let\LaTeXe\original@LaTeXe
27 \fi

```

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

```

28 \newcommand*\TeX@logo@spacing[6]{%
29 \PackageWarning{xltextra}{%
30 Use of \protect\TeX@logo@spacing\space is deprecated,\MessageBreak
31 recommend to use commands from package `metalogo' instead}
32 \setlogokern{Te}{#1}%
33 \setlogokern{eT}{#1}%
34 \setlogokern{eX}{#2}%
35 \setlogokern{Xe}{#2}%
36 \setlogodrop{#3}%
37 \setlogokern{La}{#4}%
38 \setlogokern{aT}{#5}%
39 \setlogokern{eL}{#6}}

```

4 Subscript and superscript

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

```

40 \if@xxt@noscript@
41 \DeclareRobustCommand*\textsubscript{%
42   \@ifstar{\realsubscript}{\fakesubscript}}
43 \DeclareRobustCommand*\textsuperscript{%
44   \@ifstar{\realsuperscript}{\fakesuperscript}}
45 \fi

```

5 Assorted commands

`\vfrac` #1: Numerator

#2: Denominator

No error checking is done to ensure that the font actually has the necessary features. Requires the xunicode package for `\textfractionsolidus`.

```

46 \ExplSyntaxOn
47 \newcommand*\vfrac[2]{
48   \fontspec_if_fontspec_font:TF
49   {
50     \fontspec_if_opentype:TF
51     {
52       {\addfontfeature{VerticalPosition=Numerator}#1}
53       \textfractionsolidus
54       {\addfontfeature{VerticalPosition=Denominator}#2}
55     }
56     {
57       {\addfontfeature{VerticalPosition=Superior}#1}
58       \textfractionsolidus
59       {\addfontfeature{VerticalPosition=Inferior}#2}
60     }
61   }
62   {
63     \PackageError {xltextra}
64       { \string\vfrac\space~can~only~be~used~with~fontspec~fonts }
65     { Nothing~more~to~tell. }
66   }
67 }
68 \ExplSyntaxOff

```

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

```

69 \newcommand\namedglyph[1]{%
70   \@tempcnta=\XeTeXglyphindex "#1"\relax
71   \ifnum\@tempcnta>0
72     \XeTeXglyph\@tempcnta
73   \else
74     \xxt@namedglyph@fallback{#1}%
75   \fi}

```

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

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

`\showhyphens` Courtesy egreg.

```
77 \ExplSyntaxOn
78 \seq_new:N \l__xetex_showhyphens_seq
79 \box_new:N \l__xetex_show_hyphens_wrapping_box
80 \box_new:N \l__xetex_show_hyphens_temp_box
81 \box_new:N \l__xetex_show_hyphens_final_box
82 \box_new:N \g__xetex_show_hyphens_word_box
83
84 \cs_new_protected:Npn \xetex_show_hyphens:n #1
85 {
86   \box_clear:N \l__xetex_show_hyphens_final_box
87   % split the input into items
88   \seq_set_split:Nnn \l__xetex_showhyphens_seq { ~ } { #1 }
89   % hyphenate all items
90   \seq_map_function:NN \l__xetex_showhyphens_seq \xetex_hyphenate_word:n
91   % set a box to the maximum dimension to force a Underfull \hbox warning
92   \hbox_set_to_wd:Nnn \l__xetex_show_hyphens_final_box { \c_max_dim }
93   {
94     \hbox_unpack_clear:N \l__xetex_show_hyphens_final_box
95   }
96 }
97
98 \cs_new_protected:Npn \xetex_hyphenate_word:n #1
99 {
100  \vbox_set:Nn \l__xetex_show_hyphens_wrapping_box
101  {% build a paragraph with the word with a very narrow line width
102   \dim_set:Nn \hsize { 1sp }
103   % disregard spurious messages
104   \hbadness = \c_ten_thousand
105   \dim_set:Nn \hfuzz { \c_max_dim }
106   % clear possible values of \everypar and other parameters
107   \everypar={ }
108   \skip_set:Nn \leftskip { 0pt }
109   \skip_set_eq:NN \rightskip \leftskip
110   % skip the first step
111   \pretolerance = \c_minus_one
112   % avoid the indentation and add a skip to allow hyphenation
113   \noindent
114   \skip_horizontal:n { 0pt }
115   #1
116   \par
117   \hbox_gset:Nn \g__xetex_show_hyphens_word_box { }
118   % start a recursion to dismantle the paragraph just built
119   \xetex_show_hyphens_split:
```

```

120 % the result is put into \g__xetex_show_hyphens_word_box
121 }
122 % add the box to the final container
123 \hbox_set:Nn \l__xetex_show_hyphens_final_box
124 {
125   \hbox_unpack_clear:N \l__xetex_show_hyphens_final_box
126   \hbox_unpack_clear:N \g__xetex_show_hyphens_word_box
127 }
128 }
129
130 \cs_new_protected:Npn \xetex_show_hyphens_split:
131 {
132   \unskip % remove the interline glue
133   \unpenalty % remove possible penalties
134   % get the last line
135   \box_set_to_last:N \l__xetex_show_hyphens_temp_box
136   \box_if_empty:NF \l__xetex_show_hyphens_temp_box
137   {% if there is a last line unpack it into a container
138     \hbox_gset:Nn \g__xetex_show_hyphens_word_box
139     {% the order is last to first
140       \hbox_unpack_clear:N \l__xetex_show_hyphens_temp_box
141       \unskip\unskip % remove spaces
142       \hbox_unpack_clear:N \g__xetex_show_hyphens_word_box
143     }
144     % restart the recursion
145     \xetex_show_hyphens_split:
146   }
147 }
148
149 \cs_set_eq:NN \showhyphens \xetex_show_hyphens:n
150 \ExplSyntaxOff

```