The `xltxtra` package

Will Robertson

2009/12/26 v0.5b

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`, `etex`, `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-emph` Disables the redefinition of `\emph` and `\em` described in section §2.2.

`no-logos` Disables the redefinition of `\TeX`, etc. described in section §2.5, but *does* still define the `\XeTeX` and `\XeLaTeX` logo commands.

`no-hyphen` Disables the redefinition of `\-` (probably harmless anyway) described in section §2.6.

`no-verb` Disables the redefinition of `\verb*` and `\begin{verbatim*}`, and the patching of various verbatim packages, as described in section §2.4.

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 Verbatim

Many verbatim mechanisms assume the existence of a 'visible space' character that exists in the ASCII space slot of the typewriter font. This character is known in unicode as U+2434: BOX OPEN, which looks like this: '␣'.

When a unicode typewriter font is used, LaTeX no longer prints visible spaces for the `verbatim*` environment and `\verb*` command. `xltxtra` fixes this problem by using the correct unicode glyph, and patches the following packages to do the same: `listings`, `fancyvrb`, `moreverb`, and `verbatim`.

In the case that the typewriter font does not contain '␣', the Latin Modern Mono font is used as a fallback.

2.5 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`.

2.6 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.

2.7 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.

2.8 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}
```

2.9 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}
  [2009/12/26 v0.5b Improvements for the "XeLaTeX" format]
```

Option processing

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

Required packages

```
\RequirePackage{ifxetex}
\RequireXeTeX
\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 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}}
```

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

Redefinition of `{\em ...}` and `\emph{...}` to use NFSS info to detect when the inner shape should be used.

```
\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.

```
\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
    \ifcsname zf@family@fontdef\f@family\endcsname
      \c@zf@script 1818326126\relax
    \font\zf@basefont="\csname zf@family@fontdef\f@family\endcsname" at \f@size pt
      \zf@set@font@type
      \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
      \else\ifzf@atsui
        \zf@make@aat@feature@string{10}{2}%
        \unless\ifx\@tempa\@empty
          {\addfontfeature{VerticalPosition=Inferior}#1}%
        \else
          \fakesubscript{#1}%
```

```
        \fi
      \fi\fi
    \else
      \fakesubscript{#1}%
    \fi
  \endgroup}
```

\realsuperscript Text superscripts:

```
\DeclareRobustCommand*\realsuperscript[1]{%
  \begingroup
    \ifcsname zf@family@fontdef\f@family\endcsname
      \c@zf@script 1818326126\relax
    \font\zf@basefont="\csname zf@family@fontdef\f@family\endcsname" at \f@size pt
      \zf@set@font@type
      \ifzf@icu
        \zf@check@ot@feat{+sups}%
        \if@tempswa
          {\addfontfeature{VerticalPosition=Superior}#1}%
        \else
          \fakesuperscript{#1}%
        \fi
      \else\ifzf@atsui
        \zf@make@aat@feature@string{10}{1}%
        \unless\ifx\@tempa\@empty
          {\addfontfeature{VerticalPosition=Superior}#1}%
        \else
         \fakesuperscript{#1}%
        \fi
      \fi\fi
    \else
      \fakesuperscript{#1}%
    \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 `\textfractionsolidus`.

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

5.5 Verbatims

Many thanks to Apostolos Syropoulos for discovering this problem and writing the redefinion of LaTeX's `verbatim` environment and `\verb*` command.

```
\unless\if@xxt@noverb@
```

`\xxt@visiblespace` Print U+2434: OPEN BOX, which is used to visibly display a space character.

```
\def\xxt@visiblespace{%
  \iffontchar\font"2423
    \expandafter\textvisiblespace
  \else
    \expandafter\xxt@visiblespace@fallback
  \fi}
```

`\xxt@visiblespace@fallback` If the current font doesn't have u2434, use Latin Modern Mono instead.

```
\def\xxt@visiblespace@fallback{{%
  \usefont{EU1}{lmtt}{\f@series}{\f@shape}%
  \textvisiblespace}}
```

`\xxt@vprintspaces` Helper macro to turn spaces active and print visible space instead.

```
\begingroup
  \catcode`\ =\active%
  \gdef\xxt@vprintspaces{\catcode`\ \active\let \xxt@visiblespace}%
\endgroup
```

`\verb` `\verb*` Redefine `\verb` to use `\xxt@vprintspaces`.

```
\def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
  \bgroup
    \verb@eol@error \let\do\@makeother \dospecials
    \verbatim@font\@noligs
    \@ifstar\@@sverb\@verb}
\def\@@sverb{\xxt@vprintspaces\@sverb}
```

It's better to put small things into `\AtBeginDocument`, so here we go:

```
\AtBeginDocument{%
  \xxt@patch@verbatim
  \xxt@patch@moreverb
  \xxt@patch@fancyvrb
  \xxt@patch@listings}
```

`verbatim*` With the `verbatim` package.

```
\def\xxt@patch@verbatim{%
  \@ifpackageloaded{verbatim}{%
   \@namedef{verbatim*}{\begingroup\@verbatim\xxt@vprintspaces\verbatim@start}%
  }{%
```

This is for vanilla LaTeX.

```
    \@namedef{verbatim*}{\@verbatim\xxt@vprintspaces\@sxverbatim}%
  }}
```

`listingcont*` This is for `moreverb`. The main `listing*` environment inherits this definition.

```
\def\xxt@patch@moreverb{%
  \@ifpackageloaded{moreverb}{%
    \@namedef{listingcont*}{%
      \def\verbatim@processline{%
        \thelisting@line \global\advance\listing@line1
        \the\verbatim@line\par}%
      \@verbatim\xxt@vprintspaces\verbatim@start}%
  }{}}
```

`listings` and `fancvrb` make things nice and easy:

```
\def\xxt@patch@fancyvrb{%
  \@ifpackageloaded{fancyvrb}{%
    \let\FancyVerbSpace\xxt@visiblespace
  }{}}
\def\xxt@patch@listings{%
  \@ifpackageloaded{listings}{%
    \let\lst@visiblespace\xxt@visiblespace
  }{}}
```

Finish verbatim features:

```
\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.

U

V

X

Z