

***stackmarks.sty* – faking stacked diacritical marks with challenged fonts in X_YL^AT_EX**

<https://github.com/bpj/stackmarks-xelatex>

Benct Philip Jonsson bpjonsson@gmail.com

Contents

- The problem
 - Todo (maybe)
- Dependencies
- Installation
- Basic usage
- User macros for stacking diacritics
 - `\stackMarks`
 - bold, italic and bold italic
- Adjusting the ‘raise’ and kerning of marks
 - `\stackMarksMarkBase`
 - User macros for setting default arguments

The problem

Sometimes you have to use fonts with X_YL^AT_EX which lack mark-to- mark anchors, and then have to fake stacked diacritics by moving around T_EX boxes. This hack makes that somewhat easier.

Note that currently it is not possible to nest `\stackMarks*` commands inside each other: since the L^AT_EX lengths used internally will be set to values which are appropriate only for one of them. This may be adressed in the future by adding support for optional extra mark arguments.

Todo (maybe)

stackmarks.sty is currently limited to one diacritic above another, the lower mark being non-faked and marks above the letter.

I plan to possibly add macros for, so please [let me know](#) if you need it!

- centering a single mark above the letter,
- support for at least three stacked marks (including the non-faked one),
- support for marks below the letter,

- stacking things which don't have Unicode general category *Mn*,
- a command/argument to set the desired height of boxes generated with the `\stackMarks*` commands,
- make it possible to set `\stackMarksMarkBase` individually for B/I/BI faces.

(The above is roughly an order of importance, at least as far as I'm concerned.)

Dependencies

- [The xparse package](#)
- [The calc package](#)

Installation

I regard this as a hack, so it never will be a real, properly packaged package on CTAN. Instead do this (on Linux, or its equivalent on your system):

```
cd ~
mkdir -p texmf/tex/latex
cd -p texmf/tex/latex
git clone https://github.com/bpj/stackmarks-xelatex.git
```

this should create a directory `~/texmf/tex/latex/stackmarks-xelatex` which contains the necessary files and documentation.

to update do:

```
cd ~/texmf/tex/latex/stackmarks-xelatex
git pull
```

Basic usage

```
...
\usepackage{stackmarks}
\setmainfont{FreeSerif}

\renewcommand{\stackMarksMarkBase}{r}
\renewcommand{\stackMarksRaiseFixEx}{-0.01}
\renewcommand{\stackMarksBRaiseFixEx}{-0.01}
\renewcommand{\stackMarksIRaiseFixEx}{-0.01}
\renewcommand{\stackMarksBIRaiseFixEx}{-0.01}
\renewcommand{\stackMarksKernFixEm}{0.0}
```

```

\renewcommand{\stackMarksIKernFixEm}{0.15}
\renewcommand{\stackMarksBKernFixEm}{0.09}
\renewcommand{\stackMarksBIKernFixEm}{0.0}

...

\begin{document}

χώρα is transliterated \textit{ch\stackMarks{ō}{\char"0301}ra}

\end{document}

```

User macros for stacking diacritics

All of these use [xparse](#) to handle their arguments, and yes all arguments in `{...}` are required, and all those in `[...]` are optional. Those in `<...>` should actually be entered as `<r>` etc., with angle brackets (well, less/greater signs) *instead* of curly or angle brackets. They are optional too and mean that you should enter characters the measures of which will be used to determine the placement of marks relative their base.

`\stackMarks`

`\stackMarksB`

`\stackMarksI`

`\stackMarksBI`

A basic call, relying on defaults takes only two arguments, viz. the base letter with the lower mark, and the upper mark to be placed above it:

```
\stackMarks{BASE}{MARK}
```

as in

```
\stackMarks{Ä}{\char"0301}
```

Normally the upper mark is placed in a zero-width, zero-height box which is raised a length equal to the difference in height between the box containing the *base*, and an empty box with the same proportions as the box containing the `\stackMarksMarkBase` – a letter fitting the actual kerning and height of the overstriking combining mark glyphs in the font, and kerned so that the center of the *mark* glyph stands over the center of the *base* box.

(The `\stackMarks*` commands always use relative units – *em* and *ex* – in the hope that the calculated mark placement will work equally well at different sizes.)

Because the box containing the upper mark does not have any width or height the spacing between lines will not be affected. This is intended as a feature – you may need to increase the spacing between lines to accomodate stacked marks, but this is better done globally so that the entire document uses the same spacing. If this is not desired, e.g. because you use only a few stacked marks, you can use something like

```
\raisebox{0p}[1.5em]{\stackmarks{Ã}{\char"0301}}
```

Depending on the font this necessarily simple placement algorithm does not always lead to results which look good. For this reason there are a number of optional extra parameters which can be used to fine-tune the placement of the upper mark; you can, so to say, ‘lie’ to the commands about the width and the height of the *bases* bounding box, as well as specifying a custom letter to calculate the proportions of the `\stackMarksMarkBase` box. Each of these optional extra commands has a default, which is equal to the return value of a command which shall be redefined by the user to reflect the actual needs of the *mainfont* of the document. These default- defining commands are described further below. Because the needed default values will usually differ between normal, bold, italic and bold italic faces of a font there is a variant form of `\stackMarks`, with different defaults, for each of Bold, Italic and BoldItalic, called `\stackMarksB`, `\stackMarksI`, and `\stackMarksBI` respectively, which each should be used in the appropriate context.

The full set of arguments is between two and five:

```
\stackMarks[KERN FIX]{BASE}[RAISE FIX]{MARK}[FAKE BASE]
```

The five possible arguments are, in order:

[*kern fix*] A (decimal) number specifying, as a fraction of an *em* like 0.15 or -0.09, how much extra kerning to add or, more often, subtract, to the kern centering the upper *mark* over the *base*.

When calculating the kern the length 1em will be multiplied by this value and added to a length which is equal to half the width of the *base* bounding box.

This argument defaults to the value – which should be a decimal number – returned by the commands `\stackMarksKernFixEm`, `\stackMarksBKernFixEm` and so on, which you should redefine to values appropriate for your main font, and for which see below.

{*base*} This mandatory argument should be the letter with the lower diacritical mark, upon which the stacked mark should be placed. In fact it can be any string, so it can be used with varying success to center a diacritical mark above a wide or narrow letter like *m* or *l* in cases where the font doesn’t do this right by itself, or to place a (single) diacritic above two tied letters like *au̇*. Remember that in such cases you may need to explicitly use `ı` U+0131 LATIN SMALL LETTER DOTLESS I and `j` U+0237 LATIN SMALL LETTER DOTLESS J.

[raise fix] A (decimal) number specifying, as a fraction of an *ex* like 0.005 or -0.01, how much extra ‘raise’ (as in `\raisebox`) to add or, more often, subtract, to the kern centering the upper *mark* over the *base*.

When calculating the ‘raise’ the length *lex* will be multiplied by this value and added to a length which is equal to the difference in height between the *base* and the *mark base* (see below!).

This argument defaults to the value – which should be a decimal number – returned by the commands `\stackMarksRaiseFixEx`, `\stackMarksRaiseFixEx` and so on, which you should redefine to values appropriate for your main font, and for which see below.

This is the vertical counterpart of *kern fix*, which see.

Note that while the *kern fix* is given in terms of an *em* the *raise fix* is given in terms of an *ex*! This is so as to not get decimal values with a lot of leading zeros, and so that you can estimate the needed value by visually comparing the height of a *base* letter with the ‘raise’ of the *mark*.

{mark} This second mandatory argument is the (upper) *mark* to place above the *base*. While you can give this mark in any notation \LaTeX understands, including an actual combining mark character it is often practical to use the `\char“xxxx` notation like `\char"0301` for COMBINING ACUTE ACCENT or an ad hoc defined command like `\newcommand{\acuteAccentMark}{\char"0301}`.

[fake base] in some fonts a combining diacritical mark in the *base* will not contribute to the bounding box of the *base*. In these cases you may give as this extra optional argument a comparable precomposed letter+diacritic character, and the height of the bounding box of that letter will be used to compute the ‘raise’ of the upper *mark* instead of the height of the actual *base*.

Adjusting the ‘raise’ and kerning of marks

These macros should be redefined to set the defaults of the optional arguments (except for the *fake base* for which it makes no sense to have a default) to appropriate values for the *mainfont* of your document, or rather the main font which you need to fix with this package!

These are defined with the standard \LaTeX `\newcommand` and thus should be redefined with `\renewcommand`, and *not* the corresponding [xparse](#) commands!.

`\stackMarksMarkBase`

Should return a letter/glyph which in the absence of anchors has the right proportions for an overstriking (combining) diacritical mark. Its proportions will be used to determine the placement of the stacked mark relative to the actual *base*. In most Latin- script fonts this letter will be a medium-width lowercase letter; in practice I’ve found this to often be *r*, which is usually a bit narrower than *a* or *n* and a bit wider than *i* and *l*, and consequently *r* is the default value. I’m still to encounter a case where the needed `\stackMarksMarkBase` is different for the different faces of a font, so there is currently only one command used for all. I’m also still to encounter a case where the needed value is not the same for all marks, for which reason this is not a parameter.

Redefinable macros for setting default arguments

Cf. the *kern fix* and *raise fix* arguments to `\stackMarks` above!

As mentioned above there are two extra arguments to `\stackMarks` and friends, referred to above as *kern fix* and *raise fix* which may be used to fine-tune the placement of the *mark* above the *base*.

This may, depending on the font, be especially necessary with italics, where the geometrical center of the bounding box of the base often does not coincide with the visual center of the lower diacritical mark, included in the base.

Because the needed default values will usually differ between normal, bold, italic and bold italic faces of a font there is a variant form of `\stackMarks`, with different defaults, for each of Bold, Italic and BoldItalic, called `\stackMarksB`, `\stackMarksI`, and `\stackMarksBI` respectively, which each should be used in the appropriate context, and their defaults can be changed individually by redefining the following commands:

- `\stackMarksKernFixEm`,
- `\stackMarksIKernFixEm`,
- `\stackMarksBKernFixEm`,
- `\stackMarksBIKernFixEm`,

for the *kern fix* of each face, and

- `\stackMarksRaiseFixEx`,
- `\stackMarksBRaiseFixEx`,
- `\stackMarksIRaiseFixEx`,
- `\stackMarksBIRaiseFixEx`,

for the *raise fix* of each face.

Note that the values returned by these commands should **not** be L^AT_EX lengths but *decimal numbers*, usually smaller than 1 since they are given in terms of *em* and *ex* respectively! (The `\stackMarks*` commands always use relative units – *em* and *ex* – in the hope that the calculated mark placement will work equally well at different sizes.)

When loading the package the value returned by each of these commands is 0.0 which may or may not be appropriate for your font.

You should write a test document where you place different upper marks over several letters (at least a, b, i, l, m, n, r, t and their respective capitals) of different height and width and with different lower diacritical marks, in the four styles, and zooming the produced PDF page to simulate differences in size, to determine the most suitable default values, then redefine the `\stackMarksKernFixEm`, `\stackMarksRaiseFixEx` and related commands to these values in your main document. There is no substitute to trial and error and trusting your eyes. Once you have found your ideal values for a particular font family you can save those values in a *.sty* file (or a *.tex* file which you `\include`) for later reuse. Remember that if you can't find a single value which looks good on all of the above- mentioned

letters you should set a default which looks good on most of them and define wrapper commands with suitable values as arguments for the rest.

Such a file for setting custom defaults may look something like this:

```
% stackmarksfreeserif.sty
% custom defaults for stackmarks appropriate for FreeSerif

\ProvidesPackage{stackmarksfreeserif}
\NeedsTeXFormat{LaTeX2e}
\usepackage{stackmarks}

\renewcommand{\stackMarksMarkBase}{r}
\renewcommand{\stackMarksRaiseFixEx}{-0.01}
\renewcommand{\stackMarksBRaiseFixEx}{-0.01}
\renewcommand{\stackMarksIRaiseFixEx}{-0.01}
\renewcommand{\stackMarksBIRaiseFixEx}{-0.01}
\renewcommand{\stackMarksKernFixEm}{0.0}
\renewcommand{\stackMarksIKernFixEm}{0.15}
\renewcommand{\stackMarksBKernFixEm}{0.09}
\renewcommand{\stackMarksBIKernFixEm}{0.0}
```