Junicode VF

Peter S. Baker

December 19, 2023

1 Introduction

This package supports Junicode VF, the variable version of Junicode (2.204 or higher) for LuaLaTeX. Junicode VF is not yet in CTAN, so you must install the font in your system in order to use it. Place the files `junicodevf.sty` and `junicodevf.lua` somewhere your TeX installation can find them (perhaps in the directory with the document you're currently working on). This package loads `fontspec`, so it is not necessary to load that package separately, even if you are using other fonts alongside Junicode VF.

2 Loading Junicode VF

Load the package in the usual way, with `\usepackage{junicodevf}`, to set Junicode VF as the main font. By default, the main font is not a set of static outlines whose proportions remain the same though they can be scaled, but rather a set of *variable* outlines that become relatively narrower and lighter as the text size decreases. You can see the difference if we scale a line of footnote text and a line of header text to the same `\large` size:

Here is some sample text for footnotes (usually about 8pt).
Here is some sample text for headers (18pt or larger).

The letter-shapes are markedly different, but on the page they look pretty much the same, because the purpose of these changes in shape, in addition to promoting legibility, is to allow blocks of text in different sizes (headers, main text, block quotations, footnotes) to coexist on a page without any of them looking too dark or too light.[1] Evenness of texture makes text in different point sizes *look* the same.

Junicode VF's package options give you a number of ways to fine-tune the look of your text:

light The weight of the type for the main text is Light. As with the default weight, and all weights selectable by options, "Light" is a range of weights that varies with the size of the type.

medium The weight of the type for the main text is Medium—that is, darker than Regular but lighter than Semibold.

semibold The weight of bold type is somewhat lighter than the usual bold. This may be a good choice if you have selected the **light** option.

weightadjustment Adjusts the weight of the type by adding this number. For example, if you choose **medium** for your document (weight averaging about 500) and **bold** (weight around 700), and also include the option `weightadjustment=-25`, then the weights of medium and bold text will be lightened by 25 (475, 675).

condensed The width of the type is narrow—about 85% of the width of the Regular style. As with the default width, and all widths selectable by options, ”Condensed” is a range of widths that varies with the size of the type.

semicondensed The width of the type is wider than condensed but narrower than the default.

expanded The width of the type is wide—about 125% of the width of the Regular style.

[1]For example, on a typical LaTeX page a footnote like this, looked at as a block of gray, is usually a little lighter than the main text. But on this page, the "color" of the footnote matches that of the main text. The variation in glyph shape responsible for this effect approximates the way letters in metal type were typically wider and heavier at small sizes.

semiexpanded The width of the type is wider than Regular but narrower than Expanded.

widthadjustment Adjusts the width of the type by adding this number. For example, if you choose **semicondensed** for your document (width averaging 87.5), and you also include the option `widthadjustment=5`, then the average width will be 92.5, between **semicondensed** and **regular**.

proportional Numbers in the document will be proportionally spaced. This is the default.

tabular Numbers will be tabular (monospaced).

oldstyle Numbers will be old-style, harmonizing with lowercase letters. This is the default.

lining Numbers will be lining, harmonizing with uppercase letters.

3 Customizing the Main Font

If the options listed in the previous section don't give you the effect you're looking for, this package's more advanced options allow you to choose from a virtually infinite number of styles. Do this by passing OpenType features for your document's main text or for one or more of the four main styles (Regular, Italic, Bold, Bold Italic), and also by supplying custom values for the font's four axes.

For example, the style used for the body text of the *Junicode Manual* is wider than the default, giving the text a lighter and more open look. You can get that look (or any other) by passing a **SizeFeatures** option for each of the four standard styles:

```
\usepackage[
    MainFeatures={StylisticSet=9},
    MainRegularSizeFeatures={
        SizeFeatures={
            {Size={-8.6},        RawFeature={axis={wght=550,wdth=120}}},
            {Size={8.6-10.99},   RawFeature={axis={wght=475,wdth=115}}},
            {Size={10.99-21.59}, RawFeature={axis={wght=400,wdth=112.5}}},
            {Size={21.59-},      RawFeature={axis={wght=351,wdth=100}}}
        }
    },
    MainItalicSizeFeatures={
        SizeFeatures={
            {Size={-8.6},        RawFeature={axis={wght=550,wdth=118}}},
            {Size={8.6-10.99},   RawFeature={axis={wght=475,wdth=114}}},
            {Size={10.99-21.59}, RawFeature={axis={wght=450,wdth=111}}},
            {Size={21.59-},      RawFeature={axis={wght=372,wdth=98}}}
        }
    },
    MainBoldSizeFeatures={
        SizeFeatures={
            {Size={-8.6},        RawFeature={axis={wght=700,wdth=120}}},
            {Size={8.6-10.99},   RawFeature={axis={wght=700,wdth=115}}},
            {Size={10.99-21.59}, RawFeature={axis={wght=650,wdth=112.5}}},
            {Size={21.59-},      RawFeature={axis={wght=600,wdth=100}}}
        }
    },
    MainBoldItalicSizeFeatures={
        SizeFeatures={
            {Size={-8.6},        RawFeature={axis={wght=700,wdth=118}}},
            {Size={8.6-10.99},   RawFeature={axis={wght=700,wdth=114}}},
            {Size={10.99-21.59}, RawFeature={axis={wght=650,wdth=111}}},
            {Size={21.59-},      RawFeature={axis={wght=600,wdth=98}}}
        }
    }
]{junicodevf}
```

This is less intimidating than it looks. With **MainRegularSizeFeatures** and the other options for font styles, we pass **SizeFeatures** to fontspec's `\setmainfont` command—precisely the same options we would pass directly to fontspec. For each size-range, **RawFeature** defines values for the font's **wght** (Weight) and **wdth** (Width) axes.[2] Possible values for **wght** are 300–700 (400 is the default), and possible values for **wdth** are 75–125 (100 is the default).

If you like, you can simplify these options by defining a new command:

```
\newcommand{\SizeRecord}[3]{
```

[2]There is also a third axis, Enlarged (ENLA), but this is highly specialized and won't be useful in most documents. It is discussed separately below.

```
    {Size={#1},RawFeature={axis={wght=#2,wdth=#3}}}
}
\usepackage[
    MainFeatures={StylisticSet=10},
    MainRegularSizeFeatures={
        SizeFeatures={
            \SizeRecord{-8.59}{550}{120},
            \SizeRecord{8.59-10.99}{475}{115},
            \SizeRecord{10.99-21.59}{400}{112.5},
            \SizeRecord{21.59-}{475}{115}
        }
    },
    . . .
]{junicodevf}
```

In addition to options like **MainRegularSizeFeatures**, you can pass options for features other than **SizeFeatures**. **MainFeatures** is for enabling features in all of the four main styles—in the example above, Stylistic Set 10 (“Entities”). You can enable features in the individual styles with **MainRegularFeatures**, **MainItalicFeatures**, **MainBoldFeatures**, and **MainBoldItalicFeatures**—named like the other options, but without **Size**. For example, if you want Discretionary Ligatures to be on only for the Italic style, simply add a **MainItalicFeatures** option:

```
\usepackage[
    MainFeatures={StylisticSet=10},
    MainItalicFeatures={Ligatures=Discretionary},
    . . .
]{junicodevf}
```

4 Selecting Alternate Styles

In addition to the document’s main font, you can choose from fifty predefined styles. These match the thirty-eight styles supplied by the static version of Junicode, plus twelve more. The commands for shifting to these styles are as follows (of the italic styles, only the base “jItalic” is listed; append “Italic” to any of the others, except “jRegular”):

\jRegular
\jItalic
\jCond
\jSmCond
\jSmExp
\jExp
\jLight
\jCondLight
\jSmCondLight
\jSmExpLight
\jExpLight
\jMedium
\jCondMedium
\jSmCondMedium
\jSmExpMedium
\jExpMedium
\jSmbold
\jCondSmbold
\jSmCondSmbold
\jSmExpSmbold
\jExpSmbold
\jBold
\jCondBold
\jSmCondBold
\jSmExpBold
\jExpBold

These commands will be self-explanatory if you bear in mind Junicode's abbreviations for style names: Cond=Condensed, Exp=Expanded, Sm=Semi.[3] Use them to shift temporarily to a style other than that of the main text. For example, to shift to the Condensed Light style for a short phrase, use this code:

```
{\jCondLight a short phrase}.
```

The result: a short phrase.

To add features to any of these styles, use the style name (without the prefixed "j" and with "Features" appended) as a package option. To change the size features for the style, do the same, but with **SizeFeatures** instead of **Features** appended. For example:

```
\usepackage[
    CondLightFeatures={
        Language=English,
        StylisticSet=2
    },
    CondLightSizeFeatures={
        SizeFeatures={
            Size={5-},RawFeature={axis={wght=325,wdth=80}}
        }
    }
]{junicodevf}
```

[3]The purpose of these abbreviations is to keep font names under the character-limit imposed by some systems.

This will shift text in the Condensed Light style from default to insular letter-shapes and slightly increase the weight and width of all glyphs in that style. Here the **SizeFeatures** section is very simple (as in the package file itself), but you can have as many size ranges as you want, just as you can for the main font.

5 The Enlarge Axis

Junicode's Enlarge axis is for a special purpose: to represent the enlarged minuscule letters that often begin sentences and other textual units in medieval manuscripts. Thus it should normally be applied only to single letters, not to runs of text.

This package defines three different styles for the Enlarge axis, in three sizes:

\EnlargedOne b
\EnlargedTwo b
\EnlargedThree b

You can produce an italic version of the enlarged minuscule by appending "Italic" to the style name. You can also customize these styles just as you can the other alternate styles. The only difference is that you need to supply a value for the Enlarge axis (ENLA) as well as the others. Again, a command for this purpose may help:

```
\newcommand{\ENLASizeRecord}[4]{
    {Size={#1},RawFeature={axis={wght=#2,wdth=#3,ENLA=#4}}}
}
\usepackage[ENLAOneFeatures={
    SizeFeatures={
        \ENLASizeRecord{5-}{600}{100}{65}
    }
}]{junicodevf}
```

6 Other Commands

This package's other commands are offered as conveniences—shorter and more mnemonic than the `fontspec` commands they invoke (though of course all `fontspec` commands remain available). Each of these commands also has a corresponding "text" command that works like `\textit{}`—that is, it takes as its sole argument the text to which the command will be applied. Each "text" command consists of the main command with "text" prefixed—for example, `\textInsularLetterForms{}` corresponding to `\InsularLetterForms`. For a fuller account of the OpenType features applied by these commands, see Chapter 4 of the *Junicode Manual*, "Feature Reference."

\AltThornEth	Applies ss01, Alternate thorn and eth.
\InsularLetterForms	Applies ss02, Insular letter-forms.
\IPAAlternates	Applies ss03, IPA alternates.
\HighOverline	Applies ss04, High Overline.
\MediumHighOverline	Applies ss05, Medium-high Overline.
\EnlargedMinuscules	Applies ss06, Enlarged minuscules.
\Underdotted	Applies ss07, Underdotted.
\ContextualLongS	Applies ss08, Contextual long s.
\AlternateFigures	Applies ss09, Alternate Figures.
\EntitiesAndTags	Applies ss10, Entities and Tags.
\EarlyEnglishFuthorc	Applies ss12, Early English Futhorc.
\ElderFuthark	Applies ss13, Elder Futhark.
\YoungerFuthark	Applies ss14, Younger Futhark.
\LongBranchToShortTwig	Applies ss15, Long Branch to Short Twig.
\ContextualRRotunda	Applies ss16, Contextual r rotunda.
\RareDigraphs	Applies ss17, Rare Digraphs.
\OldStylePunctuation	Applies ss18, Old-style Punctuation.
\LatinToGothic	Applies ss19, Latin to Gothic.
\LowDiacritics	Applies ss20, Low Diacritics.
\jcv, \textcv	Applies any Character Variant feature (see below).

The syntax of \jcv is `\jcv[num]{num}`, where the second (required) argument is the number of the Character Variant feature, and the first (optional) argument is an index into the variants provided by that feature (starting with zero, the default). \textcv takes an additional required argument (`\textcv[num]{num}{text}`—the text to which the feature should be applied.

Character Variant features can also be selected with mnemonics, listed below. For example, a feature for lowercase **a** can be expressed as `\textcv[2]{\jcva}{a}`, yielding **ɑ**.

\jcvA
\jcva
\jcvB
\jcvb
\jcvC
\jcvc
\jcvD
\jcvd
\jcvE
\jcve
\jcvF
\jcvf
\jcvG
\jcvg
\jcvH
\jcvh
\jcvI
\jcvi
\jcvJ
\jcvj
\jcvK
\jcvk
\jcvL
\jcvl
\jcvM
\jcvm
\jcvN
\jcvn
\jcvO
\jcvo
\jcvP
\jcvp
\jcvQ
\jcvq
\jcvR
\jcvr
\jcvS
\jcvs
\jcvT
\jcvt
\jcvU
\jcvu
\jcvV
\jcvv
\jcvW
\jcvw
\jcvX
\jcvx
\jcvY
\jcvy
\jcvZ
\jcvz
\jcvaa
\jcvAE
\jcvae
\jcvAO
\jcvao
\jcvAogonek
\jcvaogonek
\jcvASCIItilde
\jcvasterisk
\jcvav
\jcvbrevebelow
\jcvcombiningdieresis
\jcvcombiningdoublemacron
\jcvcombininginsulard
\jcvcombiningopena
\jcvcombiningoverline
\jcvcombiningrrotunda
\jcvcombiningzigzag
\jcvcomma
\jcvcurrency
\jcvdbar
\jcvdcroat
\jcvEng
\jcvEogonek
\jcvetabbrev
\jcvexclam
\jcvflorin
\jcvGermanpenny
\jcvglottal
\jcvlb
\jcvlhighstroke
\jcvmacron
\jcvmiddot
\jcvoPolish
\jcvounce
\jcvperiod
\jcvpunctuselevatus
\jcvquestion
\jcvrum
\jcvsemicolon
\jcvslash
\jcvspacingusabbrev
\jcvspacingzigzag
\jcvsterling
\jcvthorncrossed
\jcvTironianEt
\jcvYogh