Junicode VF

Peter S. Baker

December 6, 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. 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 JunicodeVF

Load the package in the usual way, with `\usepackage{JunicodeVF}`. By default, the main font is not a set of fixed outlines that can change only by being scaled, but rather a set of *variable* outlines where glyphs sized for footnotes (typically about 8 points) are a little heavier and wider than glyphs sized for block quotations (10 points), those are heavier and wider than glyphs for the main text (11 or 12 points), and those are heavier and wider than glyphs for headers (18 points and higher):

This text is for footnotes,
this is for block quotations,
this is for main text,
and this is for headers.

It's not easy to see the difference between sizes in a brief example like this one, but the effect is to promote legibility at small sizes and prevent text blocks from looking

either darker or lighter than the main text on a page.[1] If the package's default settings don't work for you, you can substitute your own, as explained below.

The `\usepackage` command for this package can take many options, which will be discussed in the appropriate sections below. However, the package will work perfectly well with no options at all.

3 Customizing Numbers

Junicode's default numbers are **oldstyle proportional** (1234). You can change this default for the entire document with either or both of the options **lining** and **tabular**:

```
\usepackage[lining,tabular]{JunicodeVF}
```

You can pass other options affecting the numbers of the four standard styles (Regular, Italic, Bold, Bold Italic) of the main text font via the **MainFeatures** option:

```
\usepackage[MainFeatures={StylisticSet=9}]{JunicodeVF}
```

Like most of this package's options, **MainFeatures** is a way of passing features to fontspec's various font-setting commands. In this case, Stylistic Set 9 is Junicode's "Alternate Figures" feature, which substitutes more modern shapes for certain oldstyle numbers (it is used in this document).

4 Customizing the Main Font

The main font is organized into the four traditional styles (Regular, Italic, Bold, Bold Italic), which can be invoked in the usual ways (`\textit{}`, etc.). As explained above, however, each of these styles varies with the font size, so that the actual number of styles is potentially very large.

This package's default styles stay fairly close to what you would get with Junicode's static fonts. But you can customize these styles in a number of ways. For example, the

[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.

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 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 use if we were addressing fontspec directly. 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]{
    {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**. These are **MainRegularFeatures**, **MainItalicFeatures**, and so on—named like the other options, but without **Size**. For example, if you want Discretionary Ligatures to be on only for the Regular style, simply add a **MainRegularFeatures** option:

```
\newcommand{\SizeRecord}[3]{
    {Size={#1},RawFeature={axis={wght=#2,wdth=#3}}}
}
\usepackage[
    MainFeatures={StylisticSet=10},
    MainRegularFeatures={Ligatures=Discretionary},
    MainRegularSizeFeatures={
        SizeFeatures={
```

[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.

```
            \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}
```

5 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 "Regular"):

\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. Use them to shift temporarily to a style other than that of the main text. For example, to shift briefly to the Light Condensed style for a short phrase, use this code:

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

The result: {\jCondLight 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=85}}
        }
    }
]{JunicodeVF}
```

This will shift text in the Light Condensed 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.

6 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}
```

7 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 fonspec 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 sso1, Alternate thorn and eth.
\InsularLetterForms	Applies sso2, Insular letter-forms.
\IPAAlternates	Applies sso3, IPA alternates.
\HighOverline	Applies sso4, High Overline.
\MediumHighOverline	Applies sso5, Medium-high Overline.
\EnlargedMinuscules	Applies sso6, Enlarged minuscules.
\Underdotted	Applies sso7, Underdotted.
\ContextualLongS	Applies sso8, Contextual long s.
\AlternateFigures	Applies sso9, 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