മലയാളം – LinguisTiX-maLayaLam

निरंजन

14 May 2026 (v1.0)

https://ctan.org/pkg/linguistix

https://puszcza.gnu.org.ua/projects/linguistix

https://matrix.to/#/#linguistix:matrix.org

This file is translated by അഭിലാഷ്. The original author of the document is നിരഞ്ജൻ who also maintains the package. We initialise the package with basic information.

```
⟨*malayalam⟩
\ProvidesExplPackage{linguistix-malayalam}
                     {2026-05-15}
                     {v1.0}
                     {മലയാളം (Malayalam)}
```

Average height of Malayalam conjuncts could be more than that of Latin letters with descenders. This requires minor adjustments in the baseline stretch. To do it uniformly for footnotes also, we use the setspace package (if not loaded already).

```

\IfPackageLoadedF { setspace } {
  \RequirePackage { setspace }
}
```

We need to set the spacing between lines for which we use an internal floating point number. It is declared here.

```

\fp_gzero_new:N \g_മലയാളം_വരികൾക്കിടയിലെ_അകലം_fp
```

This macro grabs an argument, (g-)sets it as the value of `\g_മലയാളം_വരികൾക്കിടയിലെ_അകലം_fp` and resets the baseline stretch with `\setstretch` command. If the current class is memoir, then there is a method to do that without any package. I use that.

```
\cs_new_protected:Npn \മലയാളം_വരികൾക്കിടയിലെ_അകലം:n #1 {
  \fp_gset:Nn \g_മലയാളം_വരികൾക്കിടയിലെ_അകലം_fp { #1 }
  \IfClassLoadedTF { memoir } {
    \setSingleSpace {
      \fp_use:N \g_മലയാളം_വരികൾക്കിടയിലെ_അകലം_fp
    }
    \SingleSpacing
  } {
    \setstretch {
      \fp_use:N \g_മലയാളം_വരികൾക്കിടയിലെ_അകലം_fp
    }
  }
}

\cs_gset_eq:NN \വരികൾക്കിടയിലെഅകലം
               \മലയാളം_വരികൾക്കിടയിലെ_അകലം:n
```

Here, we define two keys for adjusting the space between lines.

```
\keys_define:nn { മലയാളം } {
  വരികൾക്കിടയിലെ~ അകലം
  .code:n                 = {
    \മലയാളം_വരികൾക്കിടയിലെ_അകലം:n { #1 }
  },
  വരികൾക്കിടയിലെ~ അകലം
  .initial:n              = { 1.2 }
}
```

The babel package defines `\extrasxxxx` commands for languages where the additional code that should go with a particular language (`xxxx` in this case) is set. The `\addto` command is used for appending to the same. We start with defining additional macros for Malayalam.

```
\addto { \extrasmalayalam } {
```

We have added the command for using the socket for native numbering. It will pickup its value from the current assignment of the plugs and produce the results accordingly.

```
  \socket_use:n { lngx / native-numbering }
```

By default, LaTeX prints roman numeral (in capital) as part numbers. They are not suitable for Malayalam. Thus we change them to arabic instead.

```
  \cs_set:Npn \thepart { \lngx_counter:n { part } }
```

The default LaTeX produces Latin numerals, roman (small and capital both) alphabets and a-z alphabets with enumerate at different levels of nesting. In Malayalam, both of these are irrelevant. Thus we renew all of the concerned commands and change them to

print numbers instead. Since Malayalam doesn't have much variety available for counters, we have chosen a simpler style, i.e. I, I.I, I.I.I and I.I.I.I.

```
\cs_set:Npn \theenumi { \lngx_counter:n { enumi } }
\cs_set:Npn \theenumii {
  \lngx_counter:n { enumi } .
  \lngx_counter:n { enumii }
}
```

A period is added after each 'label' in enumerate. In default LaTeX, the label for second level enumeration is printed inside brackets. We don't need it in Malayalam. So we change the `\labelenumii` command and add a period. It is not added for the first level because that's LaTeX-default too. Similarly, this is extended to all the other levels.

```
\cs_set:Npn \labelenumii { \theenumii . }
\cs_set:Npn \theenumiii {
  \lngx_counter:n { enumi } .
  \lngx_counter:n { enumii } .
  \lngx_counter:n { enumiii }
}
\cs_set:Npn \labelenumiii { \theenumiii . }
\cs_set:Npn \theenumiv {
  \lngx_counter:n { enumi } .
  \lngx_counter:n { enumii } .
  \lngx_counter:n { enumiii } .
  \lngx_counter:n { enumiv }
}
\cs_set:Npn \labelenumiv { \theenumiv . }
```

The expex package has an independent mechanism of defining and using counters. We define a set called മലയാളം here and make it the default when Malayalam is used. Note that this change will disappear with the `\lngx_misc_reset:` command. Refer to `linguistix.pdf` for more information on this.

```
\IfPackageLoadedT { expex } {
  \definelabeltype { മലയാളം } {
    labelgen                = { list },
    labellist               = {
      ക, ഖ, ഗ, ഘ, ങ,
      ച, ഛ, ജ, ഝ, ഞ,
      ട, ഠ, ഡ, ഢ, ണ,
      ത, ഥ, ദ, ധ, ന,
      പ, ഫ, ബ, ഭ, മ,
      യ, ര, റ, ല, ള, ഴ, വ, ശ, ഷ, സ, ഹ
    },
    labelformat             = {A.},
    fullrefformat           = {XA},
    labelalign              = {left},
```

```
      labelwidth                = {1.5em}
    }
    \lingset {
      labeltype                 = { മലയാളം }
    }
  }
```

Malayalam doesn't distinguish between Italic and Upright. So I redefine `\emph` to produce the argument in bold instead. This also goes when `\lngx_misc_reset:` is used.

```
  \cs_gset_eq:NN \emph \textbf
}
```

Now we make some changes that are necessary when the main language of the document is Malayalam.

```

\tl_if_eq:NnTF \g_lngx_main_language_tl { malayalam } {
```

The `\arraystretch` command needs to be reset to a larger value so that it can incorporate Malayalam's vertical conjuncts.

```
  \cs_set:Npn \arraystretch { 1.2 }
```

The Rachana Institute of Typography has created a variety of fonts that have full glyph set of Malayalam. Rachana and MeeraNew have the most robust feature-set and hence they are set in main and sans families respectively. In the context of mono, despite being a little less feature-rich, tnjoy looks like a better fit. So I have used it.

```
  \lngx_set_keys:n {
    text~ main~ font         = { RIT-Rachana-Regular.ttf },
    text~ sans~ font         = { RIT-MeeraNew.ttf },
    text~ mono~ font         = { RIT-tnjoy-regular.ttf }
  }
```

RIT's fonts have Italic shapes, but since Indian scripts do not have the tradition of Italic shapes, their regular versions are used where Italic would be expected. MeeraNew doesn't provide a bold font, so a slight amount of boldness is faked.

```
  \clist_map_inline:nn {
    upright,
    italic,
    slanted,
    swash,
    small~ caps
  } {
    \lngx_set_keys:n {
      text~ #1               = { RIT-Rachana-Regular.ttf },
      text~ bold~ #1         = { RIT-Rachana-Bold.ttf },
      text~ sans~ #1         = { RIT-MeeraNew.ttf },
      text~ sans~ bold~ #1   = { RIT-MeeraNew.ttf },
      text~ sans~ bold~ #1~
```

```
      features                = {
        FakeBold                = { 1.2 }
      },
      text~ mono~ #1          = { RIT-tnjoy-regular.ttf },
      text~ mono~ bold~ #1    = { RIT-tnjoy-bold.ttf }
    }
  }
```

Using the `text extra features` key, I declare NFSS families called `മലയാളം_main`, `മലയാളം_sans`, `മലയാളം_mono` that will be used later. Malayalam looks too small when combined with Latin script as the base characters are set at the x-height in RIT's fonts. In order to present both the scripts with equal importance, I scale Malayalam fonts and bring them to the cap-height.

```
  \lngx_set_keys:n {
    text~ main~ extra~
    features                = {
      NFSSFamily              = { മലയാളം_main },
      Scale                   = { 1.3 }
    },
    text~ sans~ extra~
    features                = {
      NFSSFamily              = { മലയാളം_sans },
      Scale                   = { 1.3 }
    },
    text~ mono~ extra~
    features                = {
      NFSSFamily              = { മലയാളം_mono },
      Scale                   = { 1.3 }
    }
  }
} {
```

In the false branch (i.e., if Malayalam is not the main language), I use the macros that set the 'other' (non-main) fonts for Malayalam. The fonts are the same, but the key-value interface is not available, so the code is slightly verbose. Before that we load the `onchar` option with `ids` and `fonts` options.

```
  \babelprovide [
    onchar                  = { ids~ fonts }
  ] { malayalam }
  \lngx_other_main_font:nnn { malayalam } {
    NFSSFamily              = { മലയാളം_main },
    Scale                   = { 1.3 },
    UprightFont             = { RIT-Rachana-Regular.ttf },
    ItalicFont              = { RIT-Rachana-Regular.ttf },
    BoldFont                = { RIT-Rachana-Bold.ttf    },
    BoldItalicFont          = { RIT-Rachana-Bold.ttf    },
```

```
    SlantedFont              = { RIT-Rachana-Regular.ttf },
    BoldSlantedFont          = { RIT-Rachana-Bold.ttf    },
    SwashFont                = { RIT-Rachana-Regular.ttf },
    BoldSwashFont            = { RIT-Rachana-Bold.ttf    },
    SmallCapsFont            = { RIT-Rachana-Regular.ttf }
  } { RIT-Rachana-Regular.ttf }
  \lngx_other_sans_font:nnn { malayalam } {
    NFSSFamily               = { മലയാളം_sans },
    Scale                    = { 1.3 },
    UprightFont              = { RIT-MeeraNew.ttf },
    ItalicFont               = { RIT-MeeraNew.ttf },
    BoldFont                 = { RIT-MeeraNew.ttf },
    BoldFeatures             = { FakeBold = { 1.2 } },
    BoldItalicFont           = { RIT-MeeraNew.ttf },
    BoldItalicFeatures       = { FakeBold = { 1.2 } },
    SlantedFont              = { RIT-MeeraNew.ttf },
    BoldSlantedFeatures      = { FakeBold = { 1.2 } },
    BoldSlantedFont          = { RIT-MeeraNew.ttf },
    SwashFont                = { RIT-MeeraNew.ttf },
    BoldSwashFont            = { RIT-MeeraNew.ttf },
    BoldSwashFeatures        = { FakeBold = { 1.2 } },
    SmallCapsFont            = { RIT-MeeraNew.ttf }
  } { RIT-Rachana-Regular.ttf }
  \lngx_other_mono_font:nnn { malayalam } {
    NFSSFamily               = { മലയാളം_mono },
    Scale                    = { 1.3 },
    UprightFont              = { RIT-tnjoy-regular.ttf },
    ItalicFont               = { RIT-tnjoy-regular.ttf },
    BoldFont                 = { RIT-tnjoy-bold.ttf },
    BoldItalicFont           = { RIT-tnjoy-bold.ttf },
    SlantedFont              = { RIT-tnjoy-regular.ttf },
    BoldSlantedFont          = { RIT-tnjoy-bold.ttf },
    SwashFont                = { RIT-tnjoy-regular.ttf },
    BoldSwashFont            = { RIT-tnjoy-bold.ttf },
    SmallCapsFont            = { RIT-tnjoy-regular.ttf }
  } { RIT-Rachana-Regular.ttf }
}
```

In order to allow Malayalam in math mode, I use the following code. Since this setting has to be done after the initialisation, I use a hook for lazy loading.

```

\IfPackageLoadedT { lua-unicode-math } {
  \DeclareSymbolFont { മലയാളം }
                     { TU }
                     { മലയാളം_main }
                     { m }
```

```
                    { n }
}

\hook_gput_code:nnn { begindocument / end } { . } {
  \IfPackageLoadedF { lua-unicode-math } {
    \DeclareSymbolFont { മലയാളം }
                       { TU }
                       { മലയാളം_main }
                       { m }
                       { n }
  }
  \int_step_inline:nnn { "0D00 } { "0D7F } {
    \Umathcode #1 = "0 ~ \use:c { symമലയാളം } ~ #1
  }
}
```
⟨/malayalam⟩